Cisco cBR Converged Broadband Routers DOCSIS Software Configuration Guide for Cisco IOS XE Everest 16.5.1

First Published: 2017-04-07

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

© 2017 Cisco Systems, Inc. All rights reserved.

CONTENTS

PART I **Basic Configuration** 69

CHAPTER 1 **Start Up Configuration of the Cisco cBR Router** 1

- Prerequisites for Configuring the Cisco CMTS 2
- Booting and Logging onto the Cisco CMTS 3
- First Time Boot Up with ROMMON 3
- Configuration Register 4
- Setting Environment Variables 5
- Unsetting Environment Variables 5
- Booting from the TFTP on the Cisco cBR 6
- Listing Supported Devices 6
- Booting from the Device on the Cisco cBR 7
- Setting AUTOBOOT image in ROMMON 7
- Verifying the ROMMON Version 8
- Resetting the Cisco cBR 8
- File Systems 9
- Verification of Hardware Bring Up 10
  - Monitoring the Cisco cBR Chassis Using CLI 10
- Gigabit Ethernet Management Interface Overview 17
- Gigabit Ethernet Port Numbering 18
- IP Address Handling in ROMMON and the Management Ethernet Port 18
- Gigabit Ethernet Management Interface VRF 18
- Common Ethernet Management Tasks 19
- Viewing the VRF Configuration 19
- Setting a Default Route in the Management Ethernet Interface VRF 19
- Setting the Management Ethernet IP Address 19

Telnetting over the Management Ethernet Interface 20
Pinging over the Management Ethernet Interface 20
Copy Using TFTP or FTP 20
NTP Server 20
SYSLOG Server 21
SNMP-Related Services 21
Domain Name Assignment 21
DNS service 21
RADIUS or TACACS+ Server 21
VTY lines with ACL 22
Configuring the AUX Port for Network Management 22
Preprovisioning the Supervisor in the Cisco cBR Chassis 22
Configuring the Gigabit Ethernet Interface for Network Management 23
Configuring the DTI Port on the Supervisor PIC 24
Configuring the TenGigabit Ethernet Interface for Network Management 25
Connecting the New Router to the Network 25
Setting Password Protection on the Cisco CMTS 26
Recovering Lost Password on the Cisco CMTS 27
Saving Your Configuration Settings 29
Reviewing Your Settings and Configurations 30

CHAPTER 2 **Cisco Smart Licensing 31**
Hardware Compatibility Matrix for Cisco cBR Series Routers 31
Prerequisites for Cisco Smart Licensing 32
Information About Cisco Smart Licensing 33
Downstream License 34
Out of Compliance Enforcement 34
How to Configure Cisco Smart Licensing 34
Using Cisco Smart Licensing Agent on the Router 34
Setting Up a Cisco Smart Account 35
Creating Virtual Accounts 41
Creating a Product Instance Registration Token 42
Registering the Router with the Cisco Licensing Cloud Using the Registration Token 44
How to Configure Cisco Smart Licensing using Transport Gateway Solution 44

Verifying Cisco Smart Licensing Configuration 46
Troubleshooting Cisco Smart Licensing 52
Manually Renewing the Smart License Registration 52
Unregistering the Router from Cisco Smart Licensing 52
Additional References 53
Feature Information for Cisco Smart Licensing 53

CHAPTER 3 **Consolidated Packages and SubPackages Management** 55
Finding Feature Information 55
Running the Cisco cBR Series Routers Using Individual and Optional SubPackages: An Overview 56
Running the Cisco cBR Series Routers Using a Consolidated Package: An Overview 56
Running the Cisco cBR Series Routers: A Summary 57
Software File Management Using Command Sets 58
Managing and Configuring the Router to Run Using Consolidated Packages and Individual SubPackages 59
Cable Line Card Process Restart 59
Cable Line Card Control Plane Process Restart 59
Cable Line Card Upstream Scheduler Process Restart 65
Quick Start Software Upgrade 72
Managing and Configuring a Consolidated Package Using the copy Command 72
Managing and Configuring a Router to Run Using Individual SubPackages From a Consolidated Package 73
Extracting a Consolidated Package and Booting Using the Provisioning File 74
Copying a Set of Individual SubPackage Files, and Booting Using a Provisioning File 76
Installing an Optional SubPackage 77
Upgrading Individual SubPackages 79
Patch Installation 79
Installing a Patch that Affects Both Line Card and Supervisor Card 79
Installing a Patch that Affects Only Line Cards 79
Installing a Patch that Affects Only Supervisor Cards 79
Upgrading a Line Card SubPackage 80
Additional References 87
Feature Information for Consolidated Packages and SubPackages Management 87

PART II High Availability Configuration 89

CHAPTER 4 Cisco IOS-XE In-Service Software Upgrade Process 91

Hardware Compatibility Matrix for Cisco cBR Series Routers 91

Information about In-Service Software Upgrade 92

How to Configure In-Service Software Upgrade 92

Configuring Subpackage Upgrade 93

Subpackages Upgrade 93

Linecard Only In-Service Software Upgrade 94

Additional References 94

Feature Information for In-Service Software Upgrade 95

CHAPTER 5 Supervisor Redundancy 97

Hardware Compatibility Matrix for Cisco cBR Series Routers 97

Prerequisites for Supervisor Redundancy 98

Information About Supervisor Redundancy 98

Switchover Procedure 99

Using Redundant File Systems 100

Console Port Usage After Supervisor Switchover 101

Benefits 101

How to Configure Supervisor Redundancy 102

Forcing Switchover 102

Changing the System Boot Behavior 103

Saving a Configuration File to the Bootflash or Hard Disk 106

Verifying the Supervisor Redundancy Configuration 107

Verifying Supervisor Redundancy 107

Verifying Supervisor Switchover 109

Configuration Example for Supervisor Redundancy 110

Additional References 110

Feature Information for Supervisor Redundancy 110

CHAPTER 6 Line Card Redundancy 113

Hardware Compatibility Matrix for Cisco cBR Series Routers 113

Prerequisites for Line Card Redundancy 114
Restrictions for Line Card Redundancy 114
Information About Line Card Redundancy 115
How to Configure Line Card Redundancy 116
Configuring Line Card Manual Switchover 116
Configuring N+1 Line Card Redundancy 116
Verifying the Line Card Redundancy Configuration 117
Additional References 121
Feature Information for Line Card Redundancy 122

PART III **Layer 2 and DOCSIS 3.0 Configuration** 123

CHAPTER 7 **Downstream Interface Configuration** 125
Finding Feature Information 125
Hardware Compatibility Matrix for Cisco cBR Series Routers 125
Information About Downstream Interface Configuration 126
How to Configure Downstream Interfaces 128
Configuring the Cisco CMTS Manually Using Configuration Mode 128
Configuring the QAM Profile on the Downstream Channels 128
Configuring the Frequency Profile on the Downstream Channels 129
Configuring the Controller on the Downstream Channels 130
Configuring the RF Channel on a Controller 131
Configuration Examples 132
Additional References 135
Feature Information for Downstream Interface Configuration on the Cisco cBR Router 136

CHAPTER 8 **Upstream Interface Configuration** 137
Finding Feature Information 137
Hardware Compatibility Matrix for Cisco cBR Series Routers 137
Information About Upstream Interface Configuration 138
How to Configure Upstream Interfaces 139
Configuring the Cisco CMTS Manually Using Configuration Mode 139
Configuring the Modulation Profile and Assigning to an Upstream Channel 139
Configuring the Upstream Channel with PHY Layer 140

Associating Upstream Channels with a MAC Domain and Configuring Upstream Bonding 141
Configuration Examples 142
Additional References 143
Feature Information for Upstream Interface Configuration on the Cisco cBR Router 143

CHAPTER 9 **DOCSIS Interface and Fiber Node Configuration 145**
Hardware Compatibility Matrix for Cisco cBR Series Routers 145
Overview of DOCSIS Interfaces and Fiber Node Configurations 146
Downstream Features 146
Upstream Features 147
MAC Domains (Cable Interfaces) 147
Fiber Nodes 147
Configuring DOCSIS Interfaces and Fiber Nodes 147
Configuring Upstream Channels 147
Verifying the Controller Configuration 147
Binding Upstream Channels to MAC Domain 148
Configuring Primary Capable Downstream Channels 149
Verifying Downstream Configuration in Controller 149
Configuring Integrated-cable Interface 149
Binding Primary Capable Downstream Channels to a MAC Domain 151
Configuring MAC Domain Service Groups 153
Configuring the Fiber Nodes 153
Verify MD-DS-SG Channel Membership 155
Verify MD-US-SG Channel Membership 156
Downstream Bonding Group Configuration 156
Configuring Wideband-cable Interface (Downstream Bonding Grouping) 156
Verifying the Bonding Group Interfaces 158
Upstream Bonding Group Configuration 160
Restrictions for Upstream Bonding Groups 160
Configuring Upstream Bonding Groups 161
Verifying Upstream Bonding Groups 162
Additional References 163
Feature Information for DOCSIS Interface and Fiber Node Configuration 163

CHAPTER 10 **Service Group Based Configuration of the Cisco cBR Router** 165

Service Group Profile Based Configuration 165

Service Profile Configuration for 16x8 with One MAC Domain 167

Service Profile Configuration for 16x8 with Two MAC Domains 169

CHAPTER 11 **DOCSIS Load Balancing Groups** 173

Hardware Compatibility Matrix for Cisco cBR Series Routers 173

Prerequisites for DOCSIS Load Balancing Groups 174

Restrictions for DOCSIS Load Balancing Groups 174

Information About DOCSIS Load Balancing Groups 175

Service-Based Load Balancing 176

RLBG/GLBG Assignment 177

Channel Assignment 178

Upstream Load Balancing for DOCSIS 3.0 Cable Modems in Single Upstream Mode 181

Auto-generate DOCSIS 2.0 GLBG 181

Independent Upstream/Downstream Throughput Rules 181

How to Configure DOCSIS Load Balancing Groups 182

Configuring DOCSIS 3.0 and 2.0 RLBG and DOCSIS 2.0 GLBG 182

Configuring DOCSIS 3.0 GLBG 185

Configuring a DOCSIS 3.0 General Load Balancing Group 186

Configuring Default Values of DOCSIS 3.0 Load Balancing Group 187

Configuring Cable Modems to RLBG or a Service Type ID 188

Configuring Rules and Policies 189

Troubleshooting Tips 190

Configuring Load Balancing Parameter for a Cable Modem Movement Failure 190

Creating and Configuring TLV type Tag 191

Configuration Examples for DOCSIS Load Balancing Groups 193

Example: Configuring a Tag 193

Example: Disabling Load Balancing 193

Verifying DOCSIS Load Balancing Groups 194

Additional References 199

Feature Information for DOCSIS Load Balancing Groups 199

CHAPTER 12 **DOCSIS Load Balancing Movements** 201

Hardware Compatibility Matrix for Cisco cBR Series Routers 202

Prerequisites 203

Prerequisites for Load Balancing 203

Prerequisites for Dynamic Channel Change for Load Balancing 203

Prerequisites for Dynamic Bonding Change for DOCSIS 3.0 Static Modem Count-Based Load Balancing 204

Restrictions 204

Restrictions for Load Balancing 204

Restrictions for Dynamic Channel Change for Load Balancing 205

DCC Restrictions with N+1 Redundancy and Inter-Card Load Balancing 206

Restrictions for DOCSIS 3.0 Static Modem Count-Based Load Balancing 207

Restrictions for Dynamic Bonding Change for DOCSIS 3.0 Static Modem Count-Based Load Balancing 207

Restrictions for MRC-Only Cable Modems 208

Information on the Load Balancing on the Cisco CMTS 208

Feature Overview 208

Methods to Determine When Interfaces Are Balanced 209

Modems Method 209

Utilization Method 210

Load Balancing Parameters 211

Configurable Minimum Threshold under Utilization Method 211

Single Channel Load Balancing 211

Error Handling of Channel Assignment 211

Downstream Load Balancing Distribution with Upstream Load Balancing 211

Upstream Load Balancing for DOCSIS 3.0 Cable Modems in Single Upstream Mode 212

Interaction with Spectrum Management 213

Using Dynamic Channel Change 213

Multiple Channel Load Balancing 214

Algorithm for Bonded Channel Cable Modem Load Balancing 214

DOCSIS 3.0 Static Modem Count-Based Load Balancing 214

Dynamic Load Balancing for DOCSIS 3.0 Cable Modems 216

Multiple Channel Load Balancing Operation 217

Using DBC for DOCSIS 3.0 Load Balancing Movement 221
Benefits of Load Balancing 223
Exclude Cable Modems from Load Balancing Groups 224
How to Configure Load Balancing 224
Enabling Single Channel Load Balancing 225
Configuring Dynamic Bonding Change for DOCSIS 3.0 Static Load Balancing 225
Excluding Cable Modems from a Load Balancing Group 225
Distributing Downstream Load Balancing with Upstream Load Balancing 226
How to Configure Dynamic Channel Change for Load Balancing 228
Configuring Dynamic Channel Change for Load Balancing 228
Verifying Load Balancing Operations 230
Example 230
Troubleshooting Tips 231
Configuration Examples for Load Balancing 234
Example: Configuring Dynamic Channel Change for Load Balancing 234
Additional References 237
Feature Information for DOCSIS Load Balancing Movements 237

CHAPTER 13 **DOCSIS 3.0 Downstream Bonding** 239
Hardware Compatibility Matrix for Cisco cBR Series Routers 239
Information About DOCSIS 3.0 Downstream Bonding 240
Receive Channel Profile 240
Receive Channel Configuration 241
RCC Template 241
Channel Assignment 241
Downstream Traffic Forwarding 241
Service Flow Priority in Downstream Extended Header 242
How to Configure RCP and RCC Encoding 242
Configuring the RCP ID 242
Configuring the RCC Templates 245
Assigning an RCC Template to a MAC Domain (Cable Interface) 247
Verifying the RCC Configuration 251
How to Configure Attribute Masks 251
Configuring Provisioned Attributes for an Integrated Cable Interface 252

Configuring Provisioned Attributes for a Wideband Cable Interface 253
Verifying the Attribute-Based Service Flow Assignments 254
How to Enable Service Flow Priority in Downstream Extender Header 255
Enabling Service Flow Priority in Downstream Extender Header 255
Verifying the Enablement of the Service Flow Priority in Downstream Extended Header 255
Enabling Verbose Reporting for Receive Channel Profiles 257
Configuration Example for an RCC Template 258
Additional References 259
Feature Information for DOCSIS 3.0 Downstream Bonding 259

CHAPTER 14 **DOCSIS 2.0 A-TDMA Modulation Profiles 261**
Hardware Compatibility Matrix for Cisco cBR Series Routers 261
Prerequisites for DOCSIS 2.0 A-TDMA Modulation Profiles 262
Restrictions for DOCSIS 2.0 A-TDMA Services 263
Information About DOCSIS 2.0 A-TDMA Services 263
Modes of Operation 264
Modulation Profiles 265
Benefits 265
How to Configure DOCSIS 2.0 A-TDMA Services 266
Creating Modulation Profiles 266
Creating a TDMA Modulation Profile 266
Creating a Mixed Mode Modulation Profile 267
Creating an A-TDMA Modulation Profile 268
Configuring the DOCSIS Mode and Profile on an Upstream 268
Monitoring the DOCSIS 2.0 A-TDMA Services 270
Displaying Modulation Profiles 270
Displaying Cable Modem Capabilities and Provisioning 271
Configuration Examples for DOCSIS 2.0 A-TDMA services 272
Creating Modulation Profiles Examples 272
Example: DOCSIS 1.0/DOCSIS 1.1 TDMA Modulation Profiles 272
Example: Mixed TDMA/A-TDMA Modulation Profiles 272
Example: DOCSIS 2.0 A-TDMA Modulation Profiles 273
Assigning Modulation Profiles to Upstreams Examples 274
Example: Assigning DOCSIS 1.0/DOCSIS 1.1 TDMA Modulation Profiles 274

Example: Assigning Mixed TDMA/A-TDMA Modulation Profiles 274
Example: Assigning DOCSIS 2.0 A-TDMA Modulation Profiles 275
Additional References 276
Feature Information for DOCSIS 2.0 A-TDMA Modulation Profile 277

CHAPTER 15 **Downstream Resiliency Bonding Group** 279
Hardware Compatibility Matrix for Cisco cBR Series Routers 280
Prerequisites for Downstream Resiliency Bonding Group 280
Restrictions for the Downstream Resiliency Bonding Group 281
Information About Downstream Resiliency Bonding Group 282
Finding a Best-Fit RBG for the Cable Modem 282
How to Configure Downstream Resiliency Bonding Group 283
Enabling Downstream Resiliency Bonding Group 283
Reserving a Resiliency Bonding Group for a Line Card 284
Verifying Downstream Resiliency Bonding Group Configuration 285
Verifying the Downstream Resiliency Bonding Group 285
Verifying a Reserved Resiliency Bonding Group 285
Downstream Resiliency Narrowband Mode Versus Resiliency Bonding Group 285
Troubleshooting the Downstream Resiliency Bonding Group Configuration 289
Configuration Examples for the Downstream Resiliency Bonding Group 289
Additional References 293
Feature Information for Downstream Resiliency Bonding Group 293

CHAPTER 16 **Downstream Channel ID Assignment** 295
Hardware Compatibility Matrix for Cisco cBR Series Routers 295
Information About Downstream Channel ID Assignment on the Cisco CMTS Routers 296
Manual Downstream Channel ID Assignment 297
Automatic Downstream Channel ID Assignment on the Cisco CMTS Routers 298
How to Configure Downstream Channel ID Assignment on the Cisco CMTS Routers 299
Configuring Manual Downstream Channel ID Assignment 299
Configuring Automatic Downstream Channel ID Assignment 300
Additional References 302
Feature Information for Downstream Channel ID Assignment 302

CHAPTER 17 **Upstream Channel Bonding** 303

Hardware Compatibility Matrix for Cisco cBR Series Routers 303

Prerequisites for Upstream Channel Bonding 304

Restrictions for Upstream Channel Bonding 304

Information About Upstream Channel Bonding 305

Multiple Transmit Channel Mode 306

Multiple Receive Channel Mode 306

Dynamic Range Window and Transmit Power Levels for Upstream Channel Bonding 306

Extended Transmit Power 307

Reduced Transmit Channel Set 308

T4 Multiplier 308

Fiber Node Configuration for Upstream Channel Bonding 309

New TLVs for Upstream Channel Bonding 309

Upstream Weighted Fair Queuing 310

Class-Based Weighted Fair Queuing 310

Activity-Based Weighted Fair Queuing 310

Custom Weight for Service Flow Priorities 310

Upstream Scheduler and Service Flows 311

Upstream Service Flow Fairness 312

Distribution of Traffic across all Channels in a USBG 312

DOCSIS 3.0 Load Balancing with USBG Smaller than Cable Modem Capabilities 312

Cisco cBR-8 CCAP Line Card Rate Limiting 313

SID Tracking 313

Service ID Clusters 313

How to Configure Upstream Channel Bonding 314

Enabling MTC Mode on a Cisco CMTS Router 314

Default MTC Mode Configuration on a Cisco CMTS Router 314

Enabling MTC Mode for All CMs 314

Configuring UCSB Required Attribute 315

Creating a Bonding Group 316

Adding Upstream Channels to a Bonding Group 317

Adding Upstream Channel Ports to a Fiber Node 318

Configuring the Class-Based Weighted Fair Queuing 319

Configuring the Activity-Based Weighted Fair Queuing 320
Configuring Custom Weights for Service Flow Priorities 320
Configuring the SID Cluster 321
Configuring the Channel Timeout for a Cable Modem 323
Configuring Cable Upstream Resiliency 323
Configuring Rate Limiting on the Cisco cBR-8 CCAP Line Card 325
Enabling Upstream Related Events for CM Status Reports 326
Modifying the Bonding Group Attributes 326
Modifying the Ranging Poll Interval on Upstream Channels 327
Configuring the Reduced Channel Set Assignment 328
Configuring DOCSIS Extended Transmit Power Feature 329
Troubleshooting Tips 330
Configuration Example for Upstream Channel Bonding 330
Example: Enabling MTC Mode for a Single CM Using the CM Configuration File 331
Verifying the Upstream Channel Bonding Configuration 332
Verifying Weighted Fair Queuing for Upstream Service Flows 332
Verifying Rate Limiting for Upstream Bonded Service Flows 332
Verifying Extended Power Transmission 332
Additional References 333
Feature Information for Upstream Channel Bonding 333

CHAPTER 18 **Spectrum Management and Advanced Spectrum Management** 335
Finding Feature Information 335
Hardware Compatibility Matrix for Cisco cBR Series Routers 336
Prerequisites for Spectrum Management 336
Restrictions for Spectrum Management 337
Shared Spectrum Groups 337
Dynamic Upstream Modulation 337
Fixed-Frequency Spectrum Groups with Advanced Spectrum Management 338
Limitations on Upstream Modulation Parameters for PacketCable VoIP Calls 338
N+1 Redundancy Support 338
Intelligent and Advanced Spectrum Management Support 338
Information About Spectrum Management 339
Spectrum Management Measurements 340

Signal and Carrier Noise Ratios 340
Differences Between the MER (SNR) and CNR (CNiR) Values 341
Additional Measurements 342
Upstream Signal Channel Overview 343
Upstream Segments and Combiner Groups 344
Frequency Management Policy 345
Noise Impairments 346
Spectrum Groups and Frequency Hopping 346
Guidelines for Spectrum Management 347
Guided and Scheduled Spectrum Management 347
Frequency Hopping Capabilities 347
Dynamic Upstream Modulation (MER [SNR]-Based) 349
Input Power Levels 351
Intelligent and Advanced Hardware-Based Spectrum Management 352
Intelligent Spectrum Management Enhancements 352
Benefits 352
Guided and Scheduled Spectrum Management Benefits 352
Intelligent and Advanced Spectrum Management Benefits 353
How to Configure Spectrum Management 354
Guided and Scheduled Spectrum Management Configuration Tasks 355
Creating and Configuring Spectrum Groups 355
Assigning a Spectrum Group to One or More Upstream Ports 357
Configuring Shared Spectrum Groups (Fiber Node Groups) for DOCSIS 3.0 358
Configuring Dynamic Upstream Modulation (MER [SNR]-Based) 358
Verifying Frequency Hopping 361
Intelligent and Advanced Spectrum Management Configuration Tasks 365
Configuring and Assigning Spectrum Groups 365
Configuring Dynamic Upstream Modulation (CNR-Based) 365
Configuring Proactive Channel Management 367
Verifying the Spectrum Management Configuration 370
Monitoring Spectrum Management 373
Using CLI Commands 373
Using SNMP 374
ccsSNRRequestTable 374

ccsSpectrumRequestTable 375
ccsSpectrumDataTable 376
ccsUpSpecMgmtTable 376
ccsHoppingNotification 378
Configuration Examples 378
Spectrum Group and Combiner Group Examples 378
Example: Verifying Spectrum Group Creation 378
Example: Time-Scheduled Spectrum Group 379
Example: Verifying Spectrum Group Configuration 379
Example: Determining the Upstream Ports Assigned to a Combiner Group 379
Example: Combiner Group 380
Example: Other Spectrum Management Configurations 381
Dynamic Upstream Modulation Examples 382
Verifying Your Settings 382
Example: Modulation Profiles 383
Example: Input Power Level 384
Advanced Spectrum Management Configuration Examples 384
Example: Advanced Spectrum Management for the Cisco cBR Series Routers 384
Additional References 385
Feature Information for Spectrum Management and Advanced Spectrum Management 386

CHAPTER 19 **Upstream Scheduler Mode** 387
Finding Feature Information 387
Hardware Compatibility Matrix for Cisco cBR Series Routers 388
Restrictions for Upstream Scheduler Mode 388
Information About Upstream Scheduler Mode for the Cisco CMTS Routers 389
How to Configure Upstream Scheduler Modes 389
Additional References 390
Feature Information for Upstream Scheduler Mode 391

CHAPTER 20 **Generic Routing Encapsulation** 393
Finding Feature Information 393
Hardware Compatibility Matrix for Cisco cBR Series Routers 394
Restrictions for Implementing Tunnels 394

Restrictions for GRE IPv6 Tunnels 395
Information About Implementing Tunnels 395
Tunneling Versus Encapsulation 395
Tunnel ToS 396
Path MTU Discovery 396
QoS Options for Tunnels 397
Information About IPv6 over IPv4 GRE Tunnels 397
Overlay Tunnels for IPv6 397
GRE IPv4 Tunnel Support for IPv6 Traffic 399
Information About GRE IPv6 Tunnels 399
Overview of GRE IPv6 Tunnels 399
How to Implement Tunnels 400
Determining the Tunnel Type 400
Configuring an IPv4 GRE Tunnel 401
GRE Tunnel Keepalive 401
What to Do Next 403
Configuring 6to4 Tunnels 404
What to Do Next 405
Verifying Tunnel Configuration and Operation 406
Configuration Examples for Implementing Tunnels 408
Example: Configuring a GRE IPv4 Tunnel 408
Configuring QoS Options on Tunnel Interfaces Examples 409
Policing Example 409
How to Configure IPv6 over IPv4 GRE Tunnels 410
Configuring GRE on IPv6 Tunnels 410
Configuration Examples for IPv6 over IPv4 GRE Tunnels 412
Example: GRE Tunnel Running IS-IS and IPv6 Traffic 412
Example: Tunnel Destination Address for IPv6 Tunnel 412
How to Configure GRE IPv6 Tunnels 413
Configuring GRE IPv6 Tunnels 413
Configuration Examples for GRE IPv6 Tunnels 414
Example: Configuring GRE IPv6 Tunnels 414
Additional References 415
Feature Information for Generic Routing Encapsulation 416

CHAPTER 21 **Transparent LAN Service over Cable** 417

Hardware Compatibility Matrix for Cisco cBR Series Routers 417
Prerequisites for Transparent LAN Service over Cable 418
Restrictions for Transparent LAN Service over Cable 418
Information About Transparent LAN Service over Cable 419
Feature Overview 419
Transparent LAN Service and Layer 2 Virtual Private Networks 420
IEEE 802.1Q Mapping 420
Overview 420
Details of IEEE 802.1Q Mapping 420
Benefits 421
How to Configure the Transparent LAN Service over Cable 421
Configuring IEEE 802.1Q VLAN Mapping 422
Enabling and Configuring Layer 2 Tunneling for IEEE 802.1Q Mapping 422
Creating the IEEE 802.1Q VLAN Bridge Group 423
Configuration Examples for Transparent LAN Service over Cable 424
Example: Configuring IEEE 802.1Q VLAN Mapping 424
Example: Configuring IEEE 802.1Q Bridge Aggregator 424
Verifying the Transparent LAN Service over Cable Configuration 425
Additional References 426
Feature Information for Transparent LAN Service over Cable 427

CHAPTER 22 **Downgrading Channel Bonding in Battery Backup Mode** 429

Hardware Compatibility Matrix for Cisco cBR Series Routers 429
Prerequisites for Downgrading Channel Bonding in Battery Backup Mode 430
Restrictions for Downgrading Channel Bonding in Battery Backup Mode 430
Information About Downgrading Channel Bonding in Battery Backup Mode 431
How to Configure Downgrading Channel Bonding in Battery Backup Mode 431
Configuring Channel Bonding Downgrade in Battery Backup Mode Globally 432
Configuring Channel Bonding Downgrade in Battery Backup Mode for MAC Domain 433
Verifying the Configuration for Channel Bonding Downgrade in Battery Backup Mode 434
Additional References 437
Feature Information for Downgrading Channel Bonding in Battery Backup Mode 438

CHAPTER 23 **Upstream Bonding Support for D-PON** 439

Hardware Compatibility Matrix for Cisco cBR Series Routers 439

Prerequisites for Upstream Bonding Support for D-PON 440

Restrictions for Upstream Bonding Support for D-PON 440

Information About Upstream Bonding Support for D-PON 441

D-PON on Upstream Scheduling 442

How to Configure Upstream Bonding Support for D-PON 442

DOCSIS 3.0 Cable Modems Upstream Bonding Enters Partial Bonding 443

Verifying the Upstream Bonding Support for D-PON 444

Additional References 444

Feature Information for Upstream Bonding Support for D-PON 445

CHAPTER 24 **Energy Management Mode** 447

Information About Energy Management Mode 447

Dynamic Downstream Bonding Group 447

Flow Chart of the CM Power State 448

Interaction with the Battery Mode 449

Handling Energy Management Request Load 451

Supervisor High Availability and Line Card Switchover 451

Prerequisites for Energy Management Mode 451

Restrictions for the Energy Management Mode 451

Restrictions for CMTS High Availability 451

Restrictions for Dynamic Bonding Group 452

Restrictions for Interaction of CMTS with Other Features 452

Voice 452

Dynamic Bonding Change and Dynamic Channel Change and Related Applications 452

Multicast 452

Committed Information Rate 452

Admission Control 453

Battery Mode 453

Attribute Mask 453

Dynamic Service Addition 453

Restrictions for Configuration Change and Interface Shutdown 453

How to Configure the Energy Management Mode 454
Enabling Energy Management Mode 454
Enabling Energy Management Mode per MAC Domain 455
Configuring Initialization Ranging Technique in Dynamic Bonding Channel 455
Configuring the Percentage for the Dynamic Channel Bandwidth 455
Configuring the Queue Size for Energy Management 455
Verifying the Energy Management Mode 455
Viewing the Basic Statistics for Energy Management Receive Request 456
Verifying the Configuration Parameters 456
Viewing Information Regarding a Cable Modem 456
Feature Information for Energy Management Mode 458

CHAPTER 25 **Cable Modem Steering** 459
Cable Modem Steering on the Cisco cBR Series Converged Broadband Routers 459
Prerequisites for Cable Modem Steering 460
Restrictions for Cable Modem Steering 460
Information About Cable Modem Steering 460
Upstream Channel Descriptor TLV for Ranging Hold-off 461
Ranging Class ID 461
Cable Modem Exclusion for DOCSIS Load Balance 461
How to Configure Cable Modem Steering on the CMTS Router 462
Configuring an Upstream Channel Class ID 462
Configuring an Upstream Ranging Hold-off Priority Value 463
Verifying and Troubleshooting Cable Modem Steering 464
Verifying an Upstream Ranging Class ID Configuration 464
Additional References 466
Feature Information for Cable Modem Steering 467

PART IV **Layer 2 and DOCSIS 3.1 Configuration** 469

CHAPTER 26 **DOCSIS 3.1 OFDM Channel Configuration** 471
Hardware Compatibility Matrix for Cisco cBR Series Routers 471
Information about OFDM Channel Configuration 472
OFDM Channels 472

Channel Profile 472
Modulation Profile 473
OFDM Channel Exclusion Band 473
How to Configure OFDM Channel 473
Configuring OFDM Modulation Profile 473
Verifying OFDM Modulation Profile Configuration 473
Configuring OFDM Channel Profile 474
Verifying OFDM Channel Profile Configuration 475
Configuring OFDM Channel as Primary Channel 476
Verifying OFDM Primary Channel Configuration 476
Configuring Port/Controller and Channel 477
Verifying Port/Controller and Channel Configuration 478
Configuration Examples 481
Additional References 482
Feature Information for DOCSIS 3.1 OFDM Channel Configuration 483

CHAPTER 27 **OFDM Channel Power Profile** 485
Hardware Compatibility Matrix for Cisco cBR Series Routers 485
Information About OFDM Channel Power Profile 486
Restrictions for Configuring OFDM Power Profile 487
How to Configure the OFDM Channel Power Profile 487
Configuring OFDM Power Profile Using Band-index 487
Verifying the Power Profile Configuration 488
Configuring OFDM Power Profile with Linear Power-tilt 488
Verifying the Power Profile Using show controller Command 488
Configuration Example for OFDM Power Profile 489
Feature Information for OFDM Channel Power Profile 489

CHAPTER 28 **DOCSIS 3.1 Path Selection** 491
Hardware Compatibility Matrix for Cisco cBR Series Routers 491
Information about Path Selection 492
How to Configure Path Selection 492
Configuring Downstream Bonding Group with OFDM Channel 492
Verifying Downstream Bonding Group with OFDM Channel Configuration 493

Verifying the Path Selection Status 493
Clearing the Path Selection Status 494
Verifying the RCC Configuration 494
Additional References 495
Feature Information for DOCSIS 3.1 Path Selection 496

CHAPTER 29 **DOCSIS 3.1 Downstream Profile Selection 497**
Hardware Compatibility Matrix for Cisco cBR Series Routers 497
Information about Downstream Profiles 498
Default Data Profile 498
Recommended Profile 499
Unfit Profile 499
How to Configure Profiles 499
Configuring Profile Downgrade 499
Configuring RxMER to Bit Loading Mapping 500
Additional References 501
Feature Information for Downstream Profile Selection 501

CHAPTER 30 **DOCSIS 3.1 Commanded Power for Upstream SC-QAMs 503**
Hardware Compatibility Matrix for Cisco cBR Series Routers 503
Information About Commanded Power Feature for Upstream SC-QAMs 504
Feature TLVs 505
TLVs Affected by Commanded Power for US SC-QAMs 505
Commanded Power Sub-TLVs 505
Additional References 506
Feature Information for Commanded Power for US SC-QAMs 506

PART V **Layer 2 and Layer 3 VPN Configuration 509**

CHAPTER 31 **L2VPN Support over Cable 511**
Finding Feature Information 511
Hardware Compatibility Matrix for Cisco cBR Series Routers 512
Prerequisites for L2VPN Support over Cable 512
Restrictions for L2VPN Support over Cable 513

VPN ID Restrictions 513
Information About L2VPN Support over Cable 514
Point-to-Point L2VPN Forwarding Mode 514
L2VPN Encodings in the CM Configuration File 515
Supported L2VPN Encodings 516
Voice-Call Support on L2VPN CM 517
How to Configure L2VPN Support over Cable 517
Configuring the Ethernet Network System Interface 517
Preparing the DOCSIS Configuration File for L2VPN Support 518
Manual Switchover Command Line Interface 518
Verifying L2VPN Support over Cable 519
Enabling Voice-Call on a L2VPN CM 521
Verifying Dynamic Service Flows 521
Configuration Examples for L2VPN over Cable 522
Example: Specifying the Ethernet NSI Interface 522
Example: Enabling Voice Call Support on MPLS L2VPN 522
Example: Enabling Voice Call Support on 802.1q L2VPN 523
Example: Enabling Voice Call Support on CLI-based L2VPN 524
Additional References 524
Feature Information for L2VPN Support over Cable 526

CHAPTER 32 **L2VPN Over Port-Channel** 527
Information About L2VPN Over Port-Channel 527
TLS L2VPN 527
DOCSIS L2VPN 527
Benefits of L2VPN Over Port-Channel 528
Restrictions for L2VPN Over Port-Channel 528
How to Configure the L2VPN Over Port-Channel 528
Configuring the Port-Channel Uplink Port for TLS L2VPN 528
Configuring the Port-Channel Uplink Port for DOCSIS L2VPN 528
Verifying Port-Channel Configuration 528
Feature Information for L2VPN Over Port-Channel 529

CHAPTER 33 **MPLS Pseudowire for Cable L2VPN** 531

Finding Feature Information 531
Hardware Compatibility Matrix for Cisco cBR Series Routers 532
Prerequisites for MPLS Pseudowire for Cable L2VPN 532
Restrictions for MPLS Pseudowire for Cable L2VPN 533
Information About MPLS Pseudowire for Cable L2VPN 533
How MPLS Transports Layer 2 Packets 534
Supported Ethernet Encapsulation on UNI 535
MPLS Pseudowire 536
Bundle254 Interface 536
Ingress Process 536
Egress Process 536
MPLS Pseudowire Control Plane Process 537
L2VPN Pseudowire Redundancy 537
MPLS Pseudowire Provisioning Methods 537
Static Provisioning Method for MPLS Pseudowires 537
Dynamic Provisioning Method for MPLS Pseudowires 538
Cisco-Specific L2VPN TLVs 539
How to Enable MPLS on a Cisco CMTS Router 541
Configuring an LDP Router ID 542
Configuring MPLS on a Gigabit Ethernet Interface 543
Configuring an MPLS Label Distribution Protocol 544
Enabling the Cisco CMTS Support for MPLS Pseudowire for Cable L2VPN 545
How to Provision MPLS Pseudowires 546
Dynamic Provisioning of MPLS Pseudowires 546
Static Provisioning Method for MPLS Pseudowires 546
How to Configure L2VPN Pseudowire Redundancy 547
Configuring the Backup Pseudowire 547
Configuring Backup Delay 548
Performing Manual Switchover 550
Troubleshooting Tips 550
Configuration Examples for MPLS Pseudowire for Cable L2VPN 550
Configuration Example for Static Provisioning of MPLS Pseudowires 551
Configuration Examples for Dynamic Provisioning of MPLS Pseudowires 551
BSOD Specification-Based MPLS Pseudowire Provisioning: Example 551

Type-4 MPLS Pseudowire Provisioning Using the CM Configuration File: Example 553
Type-5 MPLS Pseudowire Provisioning Using the CM Configuration File: Example 555
Configuration Examples for L2VPN Pseudowire Redundancy 555
Example: Configuring Backup Pseudowire Peer and VC ID 555
Example: Configuring Backup Delay 556
Example: L2VPN Backup MPLS Pseudowire Provisioning Using the CM Configuration File 556
Verifying the MPLS Pseudowire Configuration 557
Additional References 560
Feature Information for MPLS Pseudowire for Cable L2VPN 561

CHAPTER 34 **MPLS VPN Cable Enhancements** 563
Finding Feature Information 563
Hardware Compatibility Matrix for Cisco cBR Series Routers 563
Feature Overview 564
Benefits 567
Restrictions 567
Prerequisites 568
Other Important Information 568
Configuration Tasks 569
Creating VRFs for each VPN 569
Defining Subinterfaces on a Virtual Bundle Interface and Assigning VRFs 570
Configuring Cable Interface Bundles 571
Configuring Subinterfaces and MPLS VPNs on a Virtual Bundle Interface 572
Configuring MPLS in the P Routers in the Provider Core 572
Verifying the MPLS VPN Configuration 573
Configuration Examples 573
VRF Definition Configuration 573
Cable Bundle SubInterface Configuration 574
PE WAN Interface Configuration 575
PE BGP Configuration 576
Additional References 577
Feature Information for MPLS VPN Cable Enhancements 578

CHAPTER 35 **Multicast VPN and DOCSIS 3.0 Multicast QoS Support** 581

Finding Feature Information 581
Hardware Compatibility Matrix for Cisco cBR Series Routers 582
Restrictions for the Multicast VPN and DOCSIS 3.0 Multicast QoS Support 582
Information About the Multicast VPN and DOCSIS 3.0 Multicast QoS Support 583
Enhanced Quality of Service 583
Intelligent Multicast Admission Control 583
Multicast Session Limit Support 584
Multicast Virtual Private Network 584
How to Configure the Multicast VPN and DOCSIS 3.0 Multicast QoS Support 584
Configuring a QoS Profile for a Multicast Group 585
Configuring a Multicast QoS Group 585
Configuring a Default Multicast QoS Group for VRF 587
Verifying Configuration of the Multicast VPN and DOCSIS 3.0 Multicast QoS Support 588
Configuration Examples for the Multicast VPN and DOCSIS 3.0 Multicast QoS Support 589
Example: Configuring Group QoS and Group Encryption Profiles 589
Example: Configuring a QoS Group 589
Additional References 589
Feature Information for Multicast VPN and DOCSIS3.0 Multicast QoS Support 590

CHAPTER 36 **EtherChannel for the Cisco CMTS 593**
Hardware Compatibility Matrix for Cisco cBR Series Routers 594
Restrictions for EtherChannel on the Cisco CMTS 594
Information About EtherChannel on the Cisco CMTS 595
Introduction to EtherChannel on the Cisco CMTS 595
Cisco Ten Gigabit EtherChannel on the Cisco cBR Series Routers 595
How to Configure EtherChannel on the Cisco CMTS 595
Configuring Ten Gigabit EtherChannel on the Cisco CMTS 596
Troubleshooting Tips 598
What to Do Next 598
Verifying EtherChannel on the Cisco CMTS 598
Configuration Examples for EtherChannel on the Cisco CMTS 598
Additional References 599
Feature Information for EtherChannel on Cisco CMTS 600

CHAPTER 37 **Flow-Based per Port-Channel Load Balancing** 603

Hardware Compatibility Matrix for Cisco cBR Series Routers 603

Restrictions for Flow-Based per Port-Channel Load Balancing 604

Information About Flow-Based per Port-Channel Load Balancing 604

Flow-Based Load Balancing 604

Buckets for Flow-Based Load Balancing 605

Load Balancing on Port Channels 605

How to Enable Flow-Based per Port-Channel Load Balancing 607

Configuring Load Balancing on a Port Channel 607

Verifying Load Balancing Configuration on a Ten GEC Interface 608

Configuration Examples for Flow-Based per Port-Channel Load Balancing 610

Example: Flow-Based Load Balancing 610

Additional References 610

Feature Information for Flow-Based per Port-Channel Load Balancing 611

CHAPTER 38 **MPLS QoS via TLV for non-L2VPN Service Flow** 613

Hardware Compatibility Matrix for Cisco cBR Series Routers 613

Restrictions for MPLS QoS via TLV for non-L2VPN Service Flow 614

Information About MPLS QoS via TLV for non-L2VPN Service Flow 614

Configuring MPLS QoS via TLV for non-L2VPN Service Flow 615

Traffic Class for MPLS Imposition Packets 615

Traffic Classification for MPLS Disposition Packets 615

Using Vendor-Specific TLVs with AToM L2VPN and MPLS L3VPN 615

Configuration Examples 616

Example: Upstream Service Flow Marking TLV 616

Example: Downstream Packet Classification TLV 616

Example: MPLS QoS Configuration File 617

Additional References 619

Feature Information for MPLS QoS via TLV for non-L2VPN Service Flow 620

PART VI **Layer 3 Configuration** 621

CHAPTER 39 **DHCP, ToD, and TFTP Services for CMTS Routers** 623

Prerequisites for DHCP, ToD, and TFTP Services 623
Restrictions for DHCP, ToD, and TFTP Services 623
Information About DHCP, ToD, and TFTP Services 624
Feature Overview 624
External DHCP Servers 624
Cable Source Verify Feature 624
Smart Relay Feature 625
GIADDR Field 626
DHCP Relay Agent Sub-option 626
Time-of-Day Server 626
TFTP Server 628
Benefits 628
How to Configure ToD, and TFTP Services 629
Configuring Time-of-Day Service 629
Enabling Time-of-Day Service 629
Disabling Time-of-Day Service 630
Configuring TFTP Service 631
Optimizing the Use of an External DHCP Server 634
Configuring Cable Source Verify Option 634
Configuring Prefix-based Source Address Verification 636
Configuring Optional DHCP Parameters 637
How to Configure ToD, and TFTP Services 641
Configuration Examples 641
ToD Server Example 641
TFTP Server Example 641
Additional References 642
Feature Information for the DHCP, ToD, and TFTP Services for the CMTS Routers 642

CHAPTER 40 **Virtual Interface Bundling** 643
Hardware Compatibility Matrix for Cisco cBR Series Routers 643
Information About Virtual Interface Bundling 644
Overview of Virtual Interface Bundling 644
Guidelines for Virtual Interface Bundling 645
Virtual Interface Bundle-aware and Bundle-unaware Support 646

Configuring Virtual Interface Bundling 646
Verfiying the Virtual Interface Bundling Configuration 649
Additional References 651
Feature Information for Virtual Interface Bundling 651

CHAPTER 41 **IPv6 on Cable** 653
Hardware Compatibility Matrix for Cisco cBR Series Routers 654
Restrictions for IPv6 on Cable 655
Multicast Restrictions 655
QoS Restrictions 655
Information About IPv6 on Cable 656
Features Supported 656
Overview of the DOCSIS 3.0 Network Model Supporting IPv6 656
Overview of Cable Modem IPv6 Address Provisioning 658
Overview of IPv6 Dual Stack CPE Support on the CMTS 659
Overview of IPv6 over Subinterfaces 659
Overview of High Availability on IPv6 659
DOCSIS PRE HA 660
DOCSIS Line Card HA 660
Dynamic Channel Change 660
Overview of IPv6 VPN over MPLS 661
Cable Monitor 662
Overview of IPv6 CPE Router Support on the Cisco CMTS 662
Support for IPv6 Prefix Stability on the CMTS 663
Configurable DHCPv6 Relay Address 663
Support for Multiple IAPDs in a Single Advertise 664
IPv6 Neighbor Discovery Gleaning 664
How to Configure IPv6 on Cable 664
Configuring IPv6 Switching Services 664
Implementing IPv6 Addressing and Basic Connectivity for Cable Interfaces and Bundles 666
Configuring the Cable Virtual Bundle Interface 666
Configuring the IP Provisioning Mode and Bundle on the Cable Interface 668
Configuring IPv6 Cable Filter Groups 669
Configuring IPv6 Cable Filter Groups 669

Cable Filter Groups and the DOCSIS Subscriber Management MIB 669
Troubleshooting Tips 674
Configuring IPv6 Domain Name Service 674
Configuring IPv6 Source Verification 677
Configuring IPv6 VPN over MPLS 677
Configuring DHCPv6 Relay Agent 678
Disabling IPv6 ND Gleaning 679
How to Verify IPv6 Dual Stack CPE Support 679
Examples 680
Configuration Examples for IPv6 on Cable 681
Example: IPv6 over Subinterfaces 681
Example: Basic IPv6 Cable Filter Groups 682
Example: Complete Cable Configuration with IPv6 682
Example: BGP Configuration for 6VPE 690
Example: Subinterface Configuration for 6VPE 690
Example: Cable Interface Bundling 691
Example: VRF Configuration for 6VPE 691
Verifying IPv6 on Cable 692
Verifying IPv6 VRF Configuration 692
Verifying IPv6 BGP Status 692
Verifying MPLS Forwarding Table 692
Verifying IPv6 Cable Modem and its Host State 693
Verifying Multiple IAPDs in a Single Advertise 693
Additional References 694
Feature Information for IPv6 on Cable 694

CHAPTER 42 **Cable DHCP Leasequery** 695
Hardware Compatibility Matrix for Cisco cBR Series Routers 695
Prerequisites for Cable DHCP Leasequery 696
Restrictions for Cable DHCP Leasequery 696
Information About Cable DHCP Leasequery 697
DHCP MAC Address Exclusion List 698
Unitary DHCPv6 Leasequery 698
How to Configure Filtering of Cable DHCP Leasequery Requests 698

Enabling DHCP Leasequery Filtering on Downstreams 698
Enabling DHCP Leasequery Filtering on Upstreams 699
Configuring Unitary DHCPv6 Leasequery Filtering 700
Enabling DHCPv6 Leasequery Filtering on Downstreams 702
Configuration Examples for Filtering of DHCP Leasequery 702
Example: DHCP Leasequery Filtering 702
Example: Unitary DHCPv6 Leasequery Filtering 703
Additional References 703
Feature Information for Cable DHCP Leasequery 704

CHAPTER 43 **Layer 3 CPE Mobility** 705
Hardware Compatibility Matrix for Cisco cBR Series Routers 705
Prerequisites for Layer 3 CPE Mobility 706
Restrictions for Layer 3 CPE Mobility 706
Information About Layer 3 CPE Mobility 707
Benefits of Layer 3 CPE Mobility 707
How to Configure Layer 3 Mobility 708
Configuring CPE Mobility 708
Configure Source-Based Rate Limit (SBRL) for L3-mobility 709
Disabling CPE Mobility 710
Verifying Layer 3 Mobility Configuration 711
Configuration Examples for Layer 3 Mobility 711
Example: Configuring CPE Layer 3 Mobility 711
Example: Configuring SBRL for L3-mobility 712
Additional References 712
Feature Information for Layer 3 CPE Mobility 712

CHAPTER 44 **DOCSIS 3.0 Multicast Support** 713
Hardware Compatibility Matrix for Cisco cBR Series Routers 713
Prerequisites for the DOCSIS 3.0 Multicast Support 714
Restrictions for the DOCSIS 3.0 Multicast Support 714
Information About the DOCSIS 3.0 Multicast Support 715
Multicast DSID Forwarding 715
Multicast Forwarding on Bonded CM 716

Static TLV Forwarding 716
Explicit Tracking 716
Multicast Quality of Service Enhancement 717
Multicast Secondary Bonding Group 717
Load Balancing 718
Multicast DSID Forwarding Disabled Mode 718
MDF1 Support for DOCSIS 2.0 Hybrid Cable Modems 718
DSG Disablement for Hybrid STBs 719
Benefits of MDF1 Support 719
How to Configure the DOCSIS 3.0 Multicast Support 719
Configuring Basic Multicast Forwarding 719
Configuring Multicast DSID Forwarding 720
Configuring Explicit Tracking 720
Configuring Multicast QoS 720
Selecting a Forwarding Interface Based on Service Flow Attribute 722
Configuring Multicast DSID Forwarding Disabled Mode 725
How to Monitor the DOCSIS 3.0 Multicast Support 726
Verifying the Basic Multicast Forwarding 726
Verifying the Multicast DSID Forwarding 727
Verifying the Explicit Tracking Feature 728
Verifying the Multicast QoS Feature 728
Verifying the Service Flow Attributes 729
Verifying the Multicast Group Classifiers 729
Troubleshooting Tips 729
Configuration Examples for DOCSIS 3.0 Multicast Support 729
Example: Configuring Basic Multicast Forwarding 730
Example: Configuring Multicast QoS 730
Example: Configuring Forwarding Interface Selection Based on Service Flow Attribute 730
Additional References 731
Feature Information for DOCSIS 3.0 Multicast Support 732

CHAPTER 45 **IPv6 Segment Routing on Cisco cBR 733**
Hardware Compatibility Matrix for Cisco cBR Series Routers 733
Information about IPv6 Segment Routing 734

Restriction for Configuring IPv6 Segment Routing 734
How to Configure IPv6 Segment Routing 735
Configuring IPv6 Segment Routing on cBR 735
Verifying IPv6 Segment Routing Configuration 735
Configure Multiple IPv6 Addresses for Segment Routing 735
Verifying IPv6 Segment Routing Configuration on Multiple IPv6 Addresses 735
Disabling Prefix SID 736
Verifying whether Prefix SID is Disabled 736
Disabling SRv6 for a Prefix-SID 736
Verifying whether SRv6 is Disabled and Prefix SID Removed 737
Configuration Examples 737
Example: Configuring IPv6 Segment Routing on Cisco cBR 737
Example: Configure Multiple IPv6 Addresses for SRv6 737
Example: Disabling Prefix SID 737
Example: Disabling SR with an Active Prefix SID 738
Feature Information for IPv6 Segment Routing 738

PART VII **Layer 3 Configuration—IP Access Control Lists 739**

CHAPTER 46 **IP Access Control Lists 741**
Hardware Compatibility Matrix for Cisco cBR Series Routers 741
Information About IP Access Lists 742
Benefits of IP Access Lists 742
Border Routers and Firewall Routers Should Use Access Lists 743
Definition of an Access List 743
Access List Rules 744
Helpful Hints for Creating IP Access Lists 745
Named or Numbered Access Lists 746
Standard or Extended Access Lists 746
IP Packet Fields You Can Filter to Control Access 747
Wildcard Mask for Addresses in an Access List 747
Access List Sequence Numbers 748
Access List Logging 748
Alternative to Access List Logging 749

Additional IP Access List Features 749
Where to Apply an Access List 749
Additional References 750
Feature Information for IP Access Lists 751

CHAPTER 47 **Creating an IP Access List and Applying It to an Interface 753**
Hardware Compatibility Matrix for Cisco cBR Series Routers 754
Information About Creating an IP Access List and Applying It to an Interface 754
Helpful Hints for Creating IP Access Lists 754
Access List Remarks 755
Additional IP Access List Features 755
How to Create an IP Access List and Apply It to an Interface 756
Creating a Standard Access List to Filter on Source Address 756
Creating a Named Access List to Filter on Source Address 756
Creating a Numbered Access List to Filter on Source Address 758
Creating an Extended Access List 760
Creating a Named Extended Access List 760
Creating a Numbered Extended Access List 762
Applying an Access List to an Interface 764
Configuration Examples for Creating an IP Access List and Applying It to an Interface 764
Example: Filtering on Host Source Address 764
Example: Filtering on Subnet Source Address 765
Example: Filtering on Source and Destination Addresses and IP Protocols 765
Example: Filtering on Source Addresses Using a Numbered Access List 765
Example: Preventing Telnet Access to a Subnet 766
Example: Filtering on TCP and ICMP Using Port Numbers 766
Example: Allowing SMTP E-mail and Established TCP Connections 766
Example: Preventing Access to the Web by Filtering on Port Name 767
Example: Filtering on Source Address and Logging the Packets 767
Example: Limiting Debug Output 767
Additional References Creating an IP Access List and Applying It to an Interface 768
Feature Information Creating an IP Access List and Applying It to an Interface 769

CHAPTER 48 **Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports 771**

Hardware Compatibility Matrix for Cisco cBR Series Routers 771
Prerequisites for Creating an IP Access List to Filter IP Options TCP Flags Noncontiguous Ports 772
Information About Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports 772
IP Options 772
Benefits of Filtering IP Options 773
Benefits of Filtering on TCP Flags 773
TCP Flags 774
Benefits of Using the Named ACL Support for Noncontiguous Ports on an Access Control Entry Feature 774
How Filtering on TTL Value Works 774
Benefits of Filtering on TTL Value 775
How to Create an IP Access List to Filter IP Options TCP Flags Noncontiguous Ports 776
Filtering Packets That Contain IP Options 776
What to Do Next 777
Filtering Packets That Contain TCP Flags 777
Configuring an Access Control Entry with Noncontiguous Ports 779
Consolidating Access List Entries with Noncontiguous Ports into One Access List Entry 781
What To Do Next 783
Filtering Packets Based on TTL Value 783
Enabling Control Plane Policing to Filter on TTL Values 0 and 1 784
Configuration Examples for Filtering IP Options, TCP Flags, Noncontiguous Ports 786
Example: Filtering Packets That Contain IP Options 786
Example: Filtering Packets That Contain TCP Flags 787
Example: Creating an Access List Entry with Noncontiguous Ports 787
Example: Consolidating Some Existing Access List Entries into One Access List Entry with Noncontiguous Ports 787
Example: Filtering on TTL Value 788
Example: Control Plane Policing to Filter on TTL Values 0 and 1 788
Additional References 789
Feature Information for Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values 790

CHAPTER 49 **Refining an IP Access List** 791
Hardware Compatibility Matrix for Cisco cBR Series Routers 791

Information About Refining an IP Access List 792
Access List Sequence Numbers 792
Benefits of Access List Sequence Numbers 792
Sequence Numbering Behavior 793
Benefits of Time Ranges 793
Benefits Filtering Noninitial Fragments of Packets 794
Access List Processing of Fragments 794
How to Refine an IP Access List 795
Revising an Access List Using Sequence Numbers 796
Restricting an Access List Entry to a Time of Day or Week 798
What to Do Next 800
Configuration Examples for Refining an IP Access List 800
Example Resequencing Entries in an Access List 800
Example Adding an Entry with a Sequence Number 801
Example Adding an Entry with No Sequence Number 801
Example Time Ranges Applied to IP Access List Entries 801
Example Filtering IP Packet Fragments 802
Additional References 802
Feature Information for Refining an IP Access List 803

CHAPTER 50 **IP Named Access Control Lists** 805
Hardware Compatibility Matrix for Cisco cBR Series Routers 805
Information About IP Named Access Control Lists 806
Definition of an Access List 806
Named or Numbered Access Lists 807
Benefits of IP Access Lists 807
Access List Rules 808
Helpful Hints for Creating IP Access Lists 809
Where to Apply an Access List 810
How to Configure IP Named Access Control Lists 810
Creating an IP Named Access List 810
Applying an Access List to an Interface 812
Additional References for IP Named Access Control Lists 813
Feature Information for IP Named Access Control Lists 813

CHAPTER 51 **IPv4 ACL Chaining Support** 815

Hardware Compatibility Matrix for Cisco cBR Series Routers 815
Restrictions for IPv4 ACL Chaining Support 816
Information About IPv4 ACL Chaining Support 816
ACL Chaining Overview 816
IPv4 ACL Chaining Support 817
How to Configure IPv4 ACL Chaining Support 817
Configuring an Interface to Accept Common ACL 817
Configuration Examples for IPv4 ACL Chaining Support 818
Example: Configuring an Interface to Accept a Common ACL 818
Additional References for IPv4 ACL Chaining Support 819
Feature Information for IPv4 ACL Chaining Support 820

CHAPTER 52 **IPv6 ACL Chaining with a Common ACL** 821

Hardware Compatibility Matrix for Cisco cBR Series Routers 821
Information About IPv6 ACL Chaining with a Common ACL 822
ACL Chaining Overview 822
IPv6 ACL Chaining with a Common ACL 823
How to Configure IPv6 ACL Chaining with a Common ACL 823
Configuring IPv6 ACL to an Interface 823
Configuration Examples for IPv6 ACL Chaining with a Common ACL 824
Example: Configuring an Interface to Accept a Common ACL 824
Additional References for IPv6 ACL Chaining with a Common ACL 825
Feature Information for IPv6 ACL Chaining with a Common ACL 826

CHAPTER 53 **Commented IP Access List Entries** 827

Hardware Compatibility Matrix for Cisco cBR Series Routers 827
Information About Commented IP Access List Entries 828
Benefits of IP Access Lists 828
Access List Remarks 829
How to Configure Commented IP Access List Entries 829
Writing Remarks in a Named or Numbered Access List 829
Additional References for Commented IP Access List Entries 830

Feature Information for Commented IP Access List Entries 831

CHAPTER 54 **Standard IP Access List Logging** 833

Hardware Compatibility Matrix for Cisco cBR Series Routers 833

Restrictions for Standard IP Access List Logging 834

Information About Standard IP Access List Logging 834

Standard IP Access List Logging 834

How to Configure Standard IP Access List Logging 835

Creating a Standard IP Access List Using Numbers 835

Creating a Standard IP Access List Using Names 836

Configuration Examples for Standard IP Access List Logging 837

Example: Limiting Debug Output 837

Additional References for Standard IP Access List Logging 837

Feature Information for Standard IP Access List Logging 838

CHAPTER 55 **IP Access List Entry Sequence Numbering** 839

Hardware Compatibility Matrix for Cisco cBR Series Routers 839

Restrictions for IP Access List Entry Sequence Numbering 840

Information About IP Access List Entry Sequence Numbering 840

Purpose of IP Access Lists 840

How an IP Access List Works 841

IP Access List Process and Rules 841

Helpful Hints for Creating IP Access Lists 842

Source and Destination Addresses 843

Wildcard Mask and Implicit Wildcard Mask 843

Transport Layer Information 843

Benefits IP Access List Entry Sequence Numbering 843

Sequence Numbering Behavior 844

How to Use Sequence Numbers in an IP Access List 844

Sequencing Access-List Entries and Revising the Access List 844

Configuration Examples for IP Access List Entry Sequence Numbering 847

Example: Resequencing Entries in an Access List 847

Example: Adding Entries with Sequence Numbers 848

Example: Entry Without Sequence Number 848

Additional References 849
Feature Information for IP Access List Entry Sequence Numbering 849

CHAPTER 56 ACL IP Options Selective Drop 851
Hardware Compatibility Matrix for Cisco cBR Series Routers 851
Restrictions for ACL IP Options Selective Drop 852
Information About ACL IP Options Selective Drop 852
Using ACL IP Options Selective Drop 852
Benefits of Using ACL IP Options Selective Drop 853
How to Configure ACL IP Options Selective Drop 853
Configuring ACL IP Options Selective Drop 853
Configuration Examples for ACL IP Options Selective Drop 854
Example Configuring ACL IP Options Selective Drop 854
Example Verifying ACL IP Options Selective Drop 854
Additional References for IP Access List Entry Sequence Numbering 854
Feature Information for ACL IP Options Selective Drop 855

CHAPTER 57 ACL Syslog Correlation 857
Hardware Compatibility Matrix for Cisco cBR Series Routers 857
Prerequisites for ACL Syslog Correlation 858
Information About ACL Syslog Correlation 858
ACL Syslog Correlation Tags 858
ACE Syslog Messages 859
How to Configure ACL Syslog Correlation 859
Enabling Hash Value Generation on a Device 859
Disabling Hash Value Generation on a Device 860
Configuring ACL Syslog Correlation Using a User-Defined Cookie 861
Configuring ACL Syslog Correlation Using a Hash Value 862
Changing the ACL Syslog Correlation Tag Value 864
Troubleshooting Tips 865
Configuration Examples for ACL Syslog Correlation 865
Example: Configuring ACL Syslog Correlation Using a User-Defined Cookie 865
Example: Configuring ACL Syslog Correlation using a Hash Value 865
Example: Changing the ACL Syslog Correlation Tag Value 866

Additional References for IPv6 IOS Firewall 866
Feature Information for ACL Syslog Correlation 867

CHAPTER 58 IPv6 Access Control Lists 869
Hardware Compatibility Matrix for Cisco cBR Series Routers 869
Information About IPv6 Access Control Lists 870
Access Control Lists for IPv6 Traffic Filtering 870
IPv6 Packet Inspection 870
Access Class Filtering in IPv6 871
How to Configure IPv6 Access Control Lists 871
Configuring IPv6 Traffic Filtering 871
Creating and Configuring an IPv6 ACL for Traffic Filtering 871
Applying the IPv6 ACL to an Interface 872
Controlling Access to a vty 873
Creating an IPv6 ACL to Provide Access Class Filtering 873
Applying an IPv6 ACL to the Virtual Terminal Line 874
Configuration Examples for IPv6 Access Control Lists 875
Example: Verifying IPv6 ACL Configuration 875
Example: Creating and Applying an IPv6 ACL 875
Example: Controlling Access to a vty 876
Additional References 876
Feature Information for IPv6 Access Control Lists 876

CHAPTER 59 IPv6 Template ACL 879
Hardware Compatibility Matrix for Cisco cBR Series Routers 880
Information About IPv6 ACL—Template ACL 880
IPv6 Template ACL 880
How to Enable IPv6 ACL—Template ACL 881
Enabling IPv6 Template Processing 881
Configuration Examples for IPv6 ACL—Template ACL 882
Example: IPv6 Template ACL Processing 882
Additional References 882
Feature Information for IPv6 Template ACL 883

CHAPTER 60 **IPv6 ACL Extensions for Hop by Hop Filtering** 885

Hardware Compatibility Matrix for Cisco cBR Series Routers 885

Information About IPv6 ACL Extensions for Hop by Hop Filtering 886

ACLs and Traffic Forwarding 886

How to Configure IPv6 ACL Extensions for Hop by Hop Filtering 887

Configuring IPv6 ACL Extensions for Hop by Hop Filtering 887

Configuration Example for IPv6 ACL Extensions for Hop by Hop Filtering 888

Example: IPv6 ACL Extensions for Hop by Hop Filtering 888

Additional References 889

Feature Information for IPv6 ACL Extensions for Hop by Hop Filtering 889

PART VIII **Application—Voice and Video Configuration** 891

CHAPTER 61 **Unique Device Identifier Retrieval** 893

Hardware Compatibility Matrix for Cisco cBR Series Routers 893

Unique Device Identifier Overview 894

Benefits of the Unique Device Identifier Retrieval Feature 895

Retrieving the Unique Device Identifier 895

Troubleshooting Tips 898

Additional References 898

Feature Information for Unique Device Identifier Retrieval 899

CHAPTER 62 **Advanced-Mode DOCSIS Set-Top Gateway 1.2 for the Cisco CMTS Routers** 901

Hardware Compatibility Matrix for Cisco cBR Series Routers 902

Prerequisites for Advanced-Mode DSG Issue 1.2 902

Restrictions for Advanced-Mode DSG Issue 1.2 902

DSG Configuration File Transfer Operations 903

Multicast Configuration Restrictions 903

NAT for DSG Unicast-only Mapping 903

PIM and SSM for Multicast 903

Subinterfaces 903

Information About Advanced-Mode DSG Issue 1.2 903

DSG 1.2 Clients and Agents 904

FQDN Support 904
DSG Name Process and DNS Query 904
A-DSG Forwarding on the Primary Channel 905
DOCSIS 3.0 DSG MDF Support 905
Source Specific Multicast Mapping 905
How to Configure Advanced-Mode DSG Issue 1.2 906
Configuring the Default Multicast Quality of Service 906
Configuring Global Tunnel Group Settings for Advanced-Mode DSG 1.2 907
Global A-DSG 1.2 Tunnel Settings 907
Adding DSG Tunnel Group to a Subinterface 909
Configuring the DSG Client Settings for Advanced-Mode DSG 1.2 909
Configuring Downstream DSG 1.2 Settings for Advanced-Mode DSG 1.2 911
Configuring IP Multicast Operations 912
Enabling DNS Query and DSG Name Process 914
Configuring NAT to Support Unicast Messaging 915
Configuring WAN Interfaces for Multicast Operations 916
Configuring a Standard IP Access List for Packet Filtering 917
Configuring a Standard IP Access List for Multicast Group Filtering 918
Disabling A-DSG Forwarding on the Primary Channel 919
How to Monitor and Debug the Advanced-mode DOCSIS Set-Top Gateway Feature 920
Displaying Global Configurations for Advanced-Mode DSG 1.2 920
show cable dsg cfr 920
show cable dsg host 921
show cable dsg tunnel 921
show cable dsg tg 921
show running-config interface 921
show cable dsg static-group bundle 922
Displaying Interface-level Configurations for Advanced-Mode DSG 1.2 922
show cable dsg tunnel interfaces 922
show interfaces cable dsg downstream 922
show interfaces cable dsg downstream dcd 922
show interfaces cable dsg downstream tg 922
show interfaces cable dsg downstream tunnel 922
Debugging Advanced-Mode DSG 922

Configuration Examples for Advanced-Mode DSG 923
Example: Enabling DNS Query 925
Example: Disabling A-DSG Forwarding on the Primary Channel 925
Additional References 926
Feature Information for Advanced-Mode DSG 1.2 for the Cisco CMTS Routers 926

CHAPTER 63 **Cisco Network Registrar for the Cisco CMTS Routers** 927
Hardware Compatibility Matrix for Cisco cBR Series Routers 928
Servers Required on the HFC Network 928
Cisco Network Registrar Description 929
Overview of DHCP Using CNR 930
How Cisco Converged Broadband Routers and Cable Modems Work 931
DHCP Fields and Options for Cable Modems 931
Cisco Network Registrar Sample Configuration 932
Cable Modem DHCP Response Fields 935
DOCSIS DHCP Fields 935
DHCP Relay Option (DOCSIS Option 82) 935
Overview of Scripts 936
Two-way Cable Modem Scripts 936
Telco Return Cable Modem Scripts 936
Placement of Scripts 936
Windows NT 936
Solaris 937
Activating Scripts in Cisco Network Registrar 937
Configuring the Cisco CMTS Routers to Use Scripts 937
Configuring the System Default Policy 937
Cable Modems 938
PCs 938
Creating Selection Tag Scopes 938
General 938
Telco Return for the Cisco cBR-8 Router 939
Creating Network Scopes 939
Creating Policies for Class of Service or for Upgrading Cable Modem Cisco IOS Images 940
CNR Steps to Support Subinterfaces 940

Additional References 941

PART IX PacketCable and PacketCable Multimedia Configuration 943

CHAPTER 64 PacketCable and PacketCable Multimedia 945

Finding Feature Information 945

Hardware Compatibility Matrix for Cisco cBR Series Routers 946

Restrictions for PacketCable Operations 946

Information About PacketCable Operations 947

Feature Overview 947

Emergency 911 Features 947

PacketCable Emergency 911 Cable Interface Line Card Prioritization 947

PacketCable Emergency 911 Services Listing and History 948

PacketCable Network Components 948

Dynamic Quality of Service 949

Two-Stage Resource Reservation Process 949

Making a Call Using DQoS 950

DQoSLite Based IPv6 Voice Support 950

Dynamic Service Transaction ID Support 951

PacketCable Subscriber ID Support 951

Benefits 952

How to Configure PacketCable Operations 953

Enabling PacketCable Operation 953

Disabling PacketCable Operation 953

Configuring PacketCable Operation 954

Enabling Both PacketCable and Non-PacketCable UGS Service Flows 955

Enabling PacketCable Subscriber ID Support 956

Configuring RADIUS Accounting for RKS Servers 957

PacketCable Client Accept Timeout 959

Configuration Examples for PacketCable 960

Example: Typical PacketCable Configuration 960

Verifying PacketCable Operations 962

Verifying Emergency 911 Calls 964

Information About PacketCable Multimedia Operations 966

PCMM Overview 966
PCMM Enhancements over PacketCable 1.x 967
PCMM and High Availability Features on the Cisco CMTS Router 968
PCMM Gates 968
PCMM Gate Overview and PCMM Dynamic Quality of Service 968
PCMM Persistent Gate 968
PCMM Interfaces 969
PCMM to COPS Interface 969
PCMM and Distributed Cable Interface Line Cards 969
PCMM Unicast and Multicast 969
PCMM Multicast Session Range 969
How to Configure PCMM Operations 969
Enabling PCMM Operations on the Cisco CMTS Router 969
Configuring a PCMM Multicast Session Range 971
Configuration Examples for PacketCable Multimedia 972
Example: Enabling PCMM Operations on the Cisco CMTS Router 972
Example: Enabling a Multicast Session Range on the Cisco CMTS Router 972
Verifying PCMM Operations 972
High Availability Stateful Switchover (SSO) for PacketCable and PacketCable MultiMedia 974
PacketCable and PCMM with Admission Control 974
Voice MGPI Support 975
Voice Support Over DOCSIS 3.0 E-MTAs 975
PacketCable and PCMM Call Trace 975
Verifying PacketCable and PCMM Statistics 975
Additional References 977
Feature Information for PacketCable and PacketCable Multimedia 978

CHAPTER 65 **COPS Engine Operation** 981
Finding Feature Information 981
Hardware Compatibility Matrix for Cisco cBR Series Routers 981
Prerequisites for the COPS Engine on the Cisco CMTS Routers 982
Restrictions for the COPS Engine on the Cisco CMTS 982
Information About the COPS Engine on the Cisco CMTS 982
How to Configure the COPS Engine on the Cisco CMTS 983

Configuring COPS TCP and DSCP Marking 983
Configuring COPS TCP Window Size 985
Configuring Access Control List Support for COPS Engine 986
Restricting RSVP Policy to Specific Access Control Lists 986
Displaying and Verifying COPS Engine Configuration on the Cisco CMTS 987
Show Commands for COPS Engine Information 988
Displaying COPS Servers on the Network 988
Displaying COPS Policy Information on the Network 988
Displaying Access Lists for COPS 988
COPS Engine Configuration Examples for Cable 988
Example: COPS Server Specified 988
Example: COPS Server Display 989
Additional References 989
Feature Information for COPS Engine Operation 990

PART X **Quality of Services Configuration** 993

CHAPTER 66 **Dynamic Bandwidth Sharing** 995
Hardware Compatibility Matrix for Cisco cBR Series Routers 995
Information About Dynamic Bandwidth Sharing 996
How to Configure Dynamic Bandwidth Sharing 996
Configuring DBS for a Wideband Cable Interface 997
Configuring DBS for an Integrated Cable Interface 997
Verifying the Dynamic Bandwidth Sharing Configuration 998
Additional References 1001
Feature Information for Dynamic Bandwidth Sharing 1002

CHAPTER 67 **Modular Quality of Service Command-Line Interface QoS** 1003
Finding Feature Information 1003
Hardware Compatibility Matrix for Cisco cBR Series Routers 1003
Restrictions for Applying QoS Features Using the MQC 1004
About 1005
The MQC Structure 1005
Elements of a Traffic Class 1005

Elements of a Traffic Policy 1007
Nested Traffic Classes 1008
match-all and match-any Keywords of the class-map Command 1009
input and output Keywords of the service-policy Command 1009
Benefits of Applying QoS Features Using the MQC 1010
How to Apply QoS Features Using the MQC 1010
Creating a Traffic Class 1010
Creating a Traffic Policy 1011
Attaching a Traffic Policy to an Interface Using the MQC 1012
Verifying the Traffic Class and Traffic Policy Information 1013
Configuration Examples for Applying QoS Features Using the MQC 1014
Creating a Traffic Class 1014
Creating a Policy Map 1014
Example: Attaching a Traffic Policy to an Interface 1015
Using the match not Command 1015
Configuring a Default Traffic Class 1015
How Commands "class-map match-any" and "class-map match-all" Differ 1016
Establishing Traffic Class as a Match Criterion (Nested Traffic Classes) 1016
Example: Nested Traffic Class for Maintenance 1017
Example: Nested Traffic Class to Combine match-any and match-all Characteristics in One Traffic Class 1017
Example: Traffic Policy as a QoS Policy (Hierarchical Traffic Policies) 1018
How to Configure Input MQC on the Port-Channel Interfaces 1018
Creating a Traffic Class 1018
Creating a Policy Map 1019
Defining QoS Actions in a Policy Map 1019
Set Actions 1019
Configuring Aggregate Port-Channel Interface 1019
Attaching a Traffic Policy to an Interface 1020
Example: Configuring Input MQC on the Port-Channel Interfaces 1020
Additional References 1020
Feature Information for Modular Quality of Service Command-Line Interface QoS 1021

CHAPTER 68 **DOCSIS 1.1 for the Cisco CMTS Routers** 1023

Hardware Compatibility Matrix for Cisco cBR Series Routers 1023
Prerequisites for DOCSIS 1.1 Operations 1024
Restrictions for DOCSIS 1.1 Operations 1025
Information about DOCSIS 1.1 1027
Baseline Privacy Interface Plus 1027
Concatenation 1028
Dynamic MAC Messages 1028
Enhanced Quality of Service 1028
Fragmentation 1029
Interoperability 1029
Payload Header Suppression 1029
Downstream ToS Overwrite 1029
DOCSIS 1.1 Quality of Service 1029
Service Flow 1030
Service Class 1030
Packet Classifiers 1031
Packet Header Suppression Rules 1032
Quality of Service Comparison 1032
Enhanced Rate Bandwidth Allocation (ERBA) Support for DOCSIS 1.0 Cable Modems 1034
DOCSIS 3.0 Downstream Peak Traffic Rate TLV Support for ERBA 1035
Suppressing Upstream and Downstream Peak Rate TLVs for pre DOCSIS 3.0 Cable Modems 1036
Downstream Classification Enhancement with MAC Addresses 1037
Benefits 1037
How to Configure the Cisco CMTS for DOCSIS 1.1 Operations 1039
Configuring Baseline Privacy Interface 1039
Downloading the DOCSIS Root Certificate to the CMTS 1042
Adding a Manufacturer's Certificate as a Trusted Certificate 1045
Adding a Certificate as a Trusted Certificate Using SNMP Commands 1045
Adding a Manufacturer's or CM Certificate to the Hotlist 1046
Adding a Certificate to the Hotlist Using SNMP Commands 1046
Enabling Concatenation 1047
Enabling DOCSIS Fragmentation 1048
Enabling DOCSIS 1.1 Downstream Maximum Transmit Burst on the Cisco cBR-8 Router 1050
Monitoring DOCSIS Operations 1051

Monitoring the DOCSIS Network 1052
Displaying the Status of Cable Modems 1052
Displaying a Summary Report for the Cable Modems 1054
Displaying the Capabilities of the Cable Modems 1054
Displaying Detailed Information About a Particular Cable Modem 1054
Monitoring the RF Network and Cable Interfaces 1054
Displaying Information About Cloned Cable Modems 1054
Denying RF Access For Cable Modems 1054
Displaying Information About the Mac Scheduler 1055
Displaying Information About QoS Parameter Sets 1055
Displaying Information About Service Flows 1055
Displaying Information About Service IDs 1055
Monitoring BPI+ Operations 1055
Displaying the Current BPI+ State of Cable Modems 1055
Displaying the BPI+ Timer Values on the CMTS 1056
Displaying the Certificate List on the CMTS 1056
Configuration Examples for DOCSIS 1.1 Operations 1057
Example: DOCSIS 1.1 Configuration for Cisco cBR-8 Router (with BPI+) 1057
Additional References 1060
Feature Information for DOCSIS 1.1 for Cisco CMTS Routers 1061

CHAPTER 69 **Default DOCSIS 1.0 ToS Overwrite 1063**
Hardware Compatibility Matrix for Cisco cBR Series Routers 1063
Restrictions for Default DOCSIS 1.0 ToS Overwrite 1064
Information About Default DOCSIS 1.0 ToS Overwrite 1064
Default DOCSIS 1.0 ToS Overwrite Overview 1064
DOCSIS 1065
Type-of-Service (ToS) 1065
How to Configure Default DOCSIS 1.0 ToS Overwrite 1065
Enabling Default DOCSIS 1.0 ToS Overwrite 1065
Editing QoS Profiles 1066
Additional References 1067
Feature Information for Default DOCSIS 1.0 ToS Overwrite 1067

CHAPTER 70 **DOCSIS WFQ Scheduler on the Cisco CMTS Routers** 1069

Hardware Compatibility Matrix for Cisco cBR Series Routers 1069
Prerequisites for DOCSIS WFQ Scheduler 1070
Restrictions for DOCSIS WFQ Scheduler 1070
Information About DOCSIS WFQ Scheduler 1070
Queue Types 1071
Priority Queues 1072
CIR Queues 1072
Best Effort Queues 1072
DOCSIS QoS Support 1072
Traffic Priority 1073
Maximum Sustained Traffic Rate 1074
Minimum Reserved Traffic Rate 1074
High Priority Traffic 1074
Enhanced Rate Bandwidth Allocation 1074
Peak Traffic Rate 1075
DOCSIS 3.0 Downstream Bonding Support with Bonding Group Dynamic Bandwidth Sharing 1075
How to Configure DOCSIS WFQ Scheduler 1076
Mapping DOCSIS Priority to Excess Ratio 1076
Verifying the Downstream Queues Information 1077
Additional References 1077
Feature Information for DOCSIS WFQ Scheduler 1077

CHAPTER 71 **Fairness Across DOCSIS Interfaces** 1079

Hardware Compatibility Matrix for Cisco cBR Series Routers 1079
Prerequisites for Fairness Across DOCSIS Interfaces 1080
Restrictions for Fairness Across DOCSIS Interfaces 1080
Information About Fairness Across DOCSIS Interfaces 1081
On-demand CIR Acquisition 1081
Fairness Across Bonding Groups 1081
OFDM Channels 1081
Interface Bandwidth 1082
How to Configure Fairness Across DOCSIS Interfaces 1082

Configuring Fairness Across DOCSIS Interfaces 1082
Configuring Maximum Excess Information Rate Ratio 1083
Configuring Constant Excess Information Rate Demand 1084
Configuring Maximum Bonus Bandwidth 1084
Verifying the Fairness Across DOCSIS Interfaces 1086
Verifying Reservable Bandwidth 1086
Verifying Global Fairness Across DOCSIS Interfaces Status and Statistics 1087
Verifying Per-Controller Fairness Across DOCSIS Interfaces Status and Statistics 1087
Verifying Per-Interface Fairness Across DOCSIS Interfaces Status and Statistics 1088
Configuration Examples for Fairness Across DOCSIS Interfaces 1088
Example: Fairness Across DOCSIS Interfaces 1088
Example: Maximum EIR Demand Ratio 1089
Example: Constant EIR Demand 1089
Example: Maximum Bonus Bandwidth 1090
Additional References 1091
Feature Information for Fairness Across DOCSIS Interfaces 1091

CHAPTER 72 **Service Group Admission Control** 1093
Finding Feature Information 1093
Hardware Compatibility Matrix for Cisco cBR Series Routers 1093
Restrictions for Service Group Admission Control 1094
Information About Service Group Admission Control 1094
Overview 1094
SGAC and Downstream Bandwidth Utilization 1095
Categorization of Service Flows 1095
Thresholds for Downstream Bandwidth 1095
Overview of Bonding Group Admission Control 1096
How to Configure, Monitor, and Troubleshoot Service Group Admission Control 1096
Defining Rules for Service Flow Categorization 1096
Naming Application Buckets 1099
Preempting High-Priority Emergency 911 Calls 1099
Calculating Bandwidth Utilization 1100
Enabling SGAC Check 1101
Configuration Examples for SGAC 1102

Example: SGAC Configuration Commands 1102
Example: SGAC for Downstream Traffic 1104
Additional References 1104
Feature Information for Service Group Admission Control 1105

CHAPTER 73 **Subscriber Traffic Management** 1107
Hardware Compatibility Matrix for Cisco cBR Series Routers 1108
Restrictions for Subscriber Traffic Management on the Cisco CMTS Routers 1108
Information About Subscriber Traffic Management on the Cisco CMTS Routers 1109
Feature Overview 1109
Feature List 1110
Sliding Window for Monitoring Service Flows 1111
Weekend Monitoring 1112
SNMP Trap Notifications 1112
Cable Modem Interaction with the Subscriber Traffic Management Feature 1113
How to Configure the Subscriber Traffic Management Feature on the Cisco CMTS Routers 1114
Creating and Configuring an Enforce-Rule 1114
Examples 1117
Configuring Weekend Monitoring 1119
Prerequisites 1119
Restrictions 1119
Configuring Different Legacy Monitoring Conditions for Weekends 1119
Configuring Different Peak-Offpeak Monitoring Conditions for Weekends 1120
Disabling Weekend Monitoring 1121
Removing Weekend Monitoring Conditions and Use the Same Monitoring Criteria Every Day 1122
Disabling an Enforce-Rule 1123
Removing an Enforce-Rule 1123
Changing a Cable Modem Service Class 1124
Monitoring the Subscriber Traffic Management Feature on the Cisco CMTS Routers 1125
Displaying the Currently Defined Enforce-Rules 1125
Displaying the Current Subscriber Usage 1127
Configuration Examples for Subscriber Traffic Management on the Cisco CMTS Routers 1128
Example: DOCSIS Configuration File and STM Service Classes 1128
Example: Downstream Configuration 1130

Example: Upstream Configuration 1130
Example: Downstream and Upstream Configuration 1130
Example: Weekend Monitoring Configuration 1131
Additional References 1131
Feature Information for Subscriber Traffic Management 1133

PART XI **Security and Cable Monitoring Configuration** 1135

CHAPTER 74 **Dynamic Shared Secret** 1137
Hardware Compatibility Matrix for Cisco cBR Series Routers 1138
Prerequisites for Dynamic Shared Secret 1138
Restrictions for Dynamic Shared Secret 1139
General Restrictions for Dynamic Shared Secret 1139
Cable Modem Restrictions for Dynamic Shared Secret 1140
DHCP Restriction for Incognito Server and Thomson Cable Modems 1140
DOCSIS Compliance 1141
TFTP Restrictions 1142
Information About Dynamic Shared Secret 1142
Modes of Operation 1143
Operation of the Dynamic Shared Secret 1144
Interaction with Different Commands 1144
Performance Information 1145
SNMP Support 1145
System Error Messages 1146
Benefits 1147
Related Features 1148
How to Configure the Dynamic Shared Secret Feature 1148
Enabling and Configuring the Dynamic Shared Secret Feature 1148
Disabling the Dynamic Shared Secret on a Cable Interface 1150
Excluding Cable Modems from the Dynamic Shared Secret Feature 1151
Clearing the Lock on One or More Cable Modems 1152
Upgrading Firmware on the Cable Modems 1153
How to Monitor the Dynamic Shared Secret Feature 1154
Displaying Marked Cable Modems 1155

Displaying the Current Dynamic Secrets 1156
Troubleshooting Cable Modems with Dynamic Shared Secret 1158
Configuration Examples for Dynamic Shared Secret 1158
Mark Configuration: Example 1159
Lock Configuration: Example 1159
Reject Configuration: Example 1160
Disabled Configuration: Example 1160
Additional References 1160
Feature Information for Dynamic Shared Secret 1161

CHAPTER 75 **Lawful Intercept Architecture** 1163
Hardware Compatibility Matrix for Cisco cBR Series Routers 1163
Prerequisites for Lawful Intercept 1164
Restrictions for Lawful Intercept 1165
Information About Lawful Intercept 1165
Introduction to Lawful Intercept 1165
Cisco Service Independent Intercept Architecture 1165
PacketCable Lawful Intercept Architecture 1166
Cisco cBR Series Routers 1166
VRF Aware LI 1167
Lawful Intercept- Redundant Mediation Devices 1168
Lawful Intercept MIBs 1168
Restricting Access to the Lawful Intercept MIBs 1168
Service Independent Intercept 1168
Restricting Access to Trusted Hosts (without Encryption) 1169
How to Configure Lawful Intercept 1169
Creating a Restricted SNMP View of Lawful Intercept MIBs 1169
Where to Go Next 1171
Enabling SNMP Notifications for Lawful Intercept 1171
Disabling SNMP Notifications 1172
Provisioning a MAC Intercept for Cable Modems Using SNMPv3 1173
Provisioning a MAC Intercept for a CPE Device Using SNMPv3 1173
Configuration Examples for Lawful Intercept 1173
Example: Enabling Mediation Device Access Lawful Intercept MIBs 1173

Example: Configuring Lawful Intercept- Redundant Mediation Devices 1174
Additional References 1174
Feature Information for Lawful Intercept 1175

CHAPTER 76 Cable Monitoring Feature for Cisco cBR Series Routers 1177
Overview of Cable Monitor Command for cBR 1178
Configuring Cable Monitoring for cBR Routers 1178
Capturing Sniffed Packets 1180
Capturing Sniffed Packets on an External Host 1180
Capturing Sniffed Packets on a Local Hard Drive 1181
Feature Information for Cable Monitoring 1183

CHAPTER 77 Source-Based Rate Limit 1185
Hardware Compatibility Matrix for Cisco cBR Series Routers 1185
Prerequisites for Source-Based Rate Limit 1186
Restrictions for Source-Based Rate Limit 1186
Information About Source-Based Rate Limit 1186
How to Configure Source-Based Rate Limit 1187
Configuring WAN-Side Source-Based Rate Limit 1187
Configuring Control Plane Policing 1188
Enabling WAN-Side Source-Based Rate Limit 1190
Configuring WAN-Side Quarantine 1190
Configuring Subscriber-Side Source-Based Rate Limit 1191
Configuring Subscriber-Cable Modem Source-Based Rate Limit 1192
Configuring Subscriber-MAC Address Source-Based Rate Limit 1192
Configuring Punt Policing 1193
Verifying the Source-Based Rate Limit Configuration 1194
Configuration Example for Source-Based Rate Limit 1197
Conversion of Divert Rate Limit Configuration on the Cisco uBR10012 Router to SBRL Configuration on the Cisco cBR Series Routers 1199
Additional References 1202
Feature Information for Source-Based Rate Limit 1202

CHAPTER 78 Cable Duplicate MAC Address Reject 1203

Hardware Compatibility Matrix for Cisco cBR Series Routers 1203
Prerequisites for Cable Duplicate MAC Address Reject 1204
Restrictions for Cable Duplicate MAC Address Reject 1205
Information About Cable Duplicate MAC Address Reject 1205
Early Authentication and Encryption 1205
EAE Enforcement Policies 1206
EAE Exclusion 1206
BPI+ Security and Cloned Cable Modems 1206
Logging of Cloned Cable Modems 1206
DOCSIS 3.0 BPI+ Policy Enforcement 1207
BPI+ Policy Enforcement Exclusion 1208
How to Configure EAE and BPI+ Enforcement Features 1208
Configuring EAE Enforcement Policies 1208
Configuring BPI+ Enforcement Policies 1209
Configuring AES-128 for non-MTC DOCSIS3.0 Cable Modem 1210
Verifying AES-128 for non-MTC DOCSIS3.0 Cable Modem 1210
Troubleshooting Tips 1210
Configuration Example for EAE and BPI+ Enforcement Policies 1210
Verifying EAE and BPI+ Enforcement Policies 1211
What to Do Next 1211
System Messages Supporting Cable Duplicate MAC Address Reject 1211
Additional References 1212
Feature Information for Cable Duplicate MAC Address Reject 1212

CHAPTER 79 **Cable ARP Filtering** 1215
Hardware Compatibility Matrix for Cisco cBR Series Routers 1215
Prerequisites for Cable ARP Filtering 1216
Restrictions for Cable ARP Filtering 1216
Information About Cable ARP Filtering 1217
Overview 1217
Filtering ARP Traffic 1217
Monitoring Filtered ARP Traffic 1218
Linksys Wireless-Broadband Router (BEFW11S4) 1218
ARP Filtering in FP 1218

Filtering ARP Traffic in FP 1219
How to Configure Cable ARP Filtering 1219
Monitoring ARP Processing 1219
Enabling ARP Filtering 1221
Identifying the Sources of Major ARP Traffic 1222
Examples 1224
Clearing the Packet Counters 1225
Identifying ARP Offenders in FP 1225
cBR-8 Outputs in FP 1226
Configuration Examples for Cable ARP Filtering 1226
ARP Filtering Configuration on an Individual Cable Interface: Example 1226
ARP Filtering Configuration on Bundled Cable Interfaces: Example 1227
ARP Filtering in FP Default Configuration: Example 1229
Additional References 1229
Feature Information for Cable ARP Filtering 1229

CHAPTER 80 **Subscriber Management Packet Filtering Extension for DOCSIS 2.0** 1231
Hardware Compatibility Matrix for Cisco cBR Series Routers 1231
Prerequisites for Configuring Subscriber Management Packet Filtering 1232
Restriction for Configuring Subscriber Management Packet Filtering 1232
Information About Configuring Subscriber Management Packet Filtering 1233
How to Configure Subscriber Management Packet Filtering 1233
Configuring the Filter Group 1233
Defining the Upstream and Downstream MTA Filter Group 1234
Defining the Upstream and Downstream STB Filter Group 1235
Defining the Upstream and Downstream PS Filter Group 1235
Configuration Examples for Subscriber Management Packet Filtering 1236
Configuring the Filter Group: Example 1236
Defining the Upstream and Downstream MTA Filter Group: Example 1237
Defining the Upstream and Downstream STB Filter Group: Example 1237
Defining the Upstream and Downstream PS Filter Group: Example 1237
Additional References 1237
Feature Information for Subscriber Management Packet Filtering 1238

CHAPTER 81 MAC Filtering 1239
Hardware Compatibility Matrix for Cisco cBR Series Routers 1239
Information About MAC Filtering 1240
How to Configure MAC Filtering 1240
Configuring MAC Filtering 1241
Verifying MAC Filtering 1241
Configuration Examples for MAC Filtering 1244
Feature Information for MAC Filtering 1244

CHAPTER 82 IPv6 Segment Routing on Cisco cBR 1245
Hardware Compatibility Matrix for Cisco cBR Series Routers 1245
Information about IPv6 Segment Routing 1246
Restriction for Configuring IPv6 Segment Routing 1246
How to Configure IPv6 Segment Routing 1247
Configuring IPv6 Segment Routing on cBR 1247
Verifying IPv6 Segment Routing Configuration 1247
Configure Multiple IPv6 Addresses for Segment Routing 1247
Verifying IPv6 Segment Routing Configuration on Multiple IPv6 Addresses 1247
Disabling Prefix SID 1248
Verifying whether Prefix SID is Disabled 1248
Disabling SRv6 for a Prefix-SID 1248
Verifying whether SRv6 is Disabled and Prefix SID Removed 1249
Configuration Examples 1249
Example: Configuring IPv6 Segment Routing on Cisco cBR 1249
Example: Configure Multiple IPv6 Addresses for SRv6 1249
Example: Disabling Prefix SID 1249
Example: Disabling SR with an Active Prefix SID 1250
Feature Information for IPv6 Segment Routing 1250

PART XII Troubleshooting and Network Management Configuration 1251

CHAPTER 83 Call Home 1253
Hardware Compatibility Matrix for Cisco cBR Series Routers 1254

Prerequisites for Call Home 1254
Restrictions for Call Home 1255
Information About Call Home 1255
Benefits of Call Home 1256
Obtaining Smart Call Home Services 1256
Anonymous Reporting 1257
Smart Licensing 1257
How to Configure Call Home 1257
Configuring Smart Call Home (Single Command) 1257
Configuring Call Home 1259
Enabling and Disabling Call Home 1259
Configuring Contact Information 1259
Configuring Destination Profiles 1261
Subscribing to Alert Groups 1266
Configuring General email Options 1271
Sending Call Home Messages Manually 1277
Configuring Diagnostic Signatures 1281
Prerequisites for Diagnostic Signatures 1281
Information About Diagnostic Signatures 1282
Diagnostic Signature Overview 1282
Diagnostic Signature Downloading 1282
Diagnostic Signature Signing 1283
Diagnostic Signature Workflow 1283
Diagnostic Signature Events and Actions 1284
How to Configure Diagnostic Signatures 1286
Configuring the Service Call Home for Diagnostic Signatures 1286
Configuring Diagnostic Signatures 1288
Verifying the Call Home Configuration 1289
Configuration Example for Call Home 1294
Example: Call Home Configuration 1294
Example: Configuring HTTP Transport for Call Home on the Cisco cBR Series Router 1295
Example: Configuring Email Transport for Call Home on the Cisco cBR Series Router 1297
Default Settings 1299
Alert Groups Trigger Events and Commands 1300

Message Contents 1304
Sample syslog Alert Notification in XML Format 1308
Additional References 1317
Feature Information for Call Home 1318

CHAPTER 84 SNMP Support over VPNs—Context-Based Access Control 1319
Finding Feature Information 1319
Hardware Compatibility Matrix for Cisco cBR Series Routers 1319
Restrictions for SNMP Support over VPNs—Context-Based Access Control 1320
Information About SNMP Support over VPNs—Context-Based Access Control 1320
SNMP Versions and Security 1320
SNMPv1 or SNMPv2 Security 1321
SNMPv3 Security 1321
SNMP Notification Support over VPNs 1321
VPN-Aware SNMP 1322
VPN Route Distinguishers 1322
SNMP Contexts 1323
How to Configure SNMP Support over VPNs—Context-Based Access Control 1323
Configuring an SNMP Context and Associating the SNMP Context with a VPN 1323
Configuring SNMP Support and Associating an SNMP Context 1325
Configuration Examples for SNMP Support over VPNs—Context-Based Access Control 1327
Example: Configuring Context-Based Access Control 1327
Additional References 1328
Feature Information for SNMP Support over VPNs—Context-Based Access Control 1329

CHAPTER 85 SNMP Engine Enhancement 1331
Hardware Compatibility Matrix for Cisco cBR Series Routers 1331
Restrictions for SNMP Cache Engine Enhancement 1332
Information About SNMP Cache Engine Enhancement 1332
How to Configure SNMP Cache Engine Enhancement 1333
Verifying the SNMP Cache Engine Status 1334
Additional References 1335
Feature Information for SNMP Cache Engine Enhancement 1335

CHAPTER 86 **Onboard Failure Logging** 1337

Finding Feature Information 1337

Hardware Compatibility Matrix for Cisco cBR Series Routers 1337

Understanding OBFL 1338

Configuring OBFL 1339

Displaying OBFL Logging Information 1339

Clearing OBFL Logging 1339

Configuration and Verification Examples 1340

Feature Information for Onboard Failure Logging 1342

CHAPTER 87 **Control Point Discovery** 1343

Hardware Compatibility Matrix for Cisco cBR Series Routers 1343

Prerequisites for Control Point Discovery 1344

Restrictions for Control Point Discovery 1344

Information About Control Point Discovery 1345

Control Points 1345

Network Layer Signaling (NLS) 1345

NLS for CPD 1345

Control Point Discovery 1346

CPD Protocol Hierarchy 1346

Control Relationship 1346

How to Configure CPD 1347

Enabling CPD Functionality 1347

Examples for CPD Enable 1348

Debugging CPD Functionality 1348

Configuring Control Relationship Identifier 1348

Examples 1349

Enabling NLS Functionality 1349

Examples 1349

Debugging NLS Functionality 1350

Configuring Authorization Group Identifier and Authentication Key 1350

Examples 1350

Configuring NLS Response Timeout 1351

Examples 1351
Additional References 1351
Feature Information for Control Point Discovery 1352

CHAPTER 88 IPDR Streaming Protocol 1353
Restrictions for Configuring IPDR Streaming Protocol 1353
Information About IPDR Streaming Protocol 1354
Data Collection Methodologies 1354
How to Configure IPDR Streaming Protocol 1355
Configuring the IPDR Session 1355
Configuring the IPDR Type 1356
Configuring the IPDR Collector 1356
Configuring the IPDR Associate 1357
Configuring the IPDR Template 1358
Configuring the IPDR Exporter 1358
Configuration Examples for IPDR Streaming Protocol 1360
Example: Configuring the IPDR Session 1360
Example: Configuring the IPDR Type 1360
Example: Configuring the IPDR Collector 1360
Example: Configuring the IPDR Associate 1360
Example: Configuring the IPDR Template 1360
Example: Configuring the IPDR Exporter 1361
Verifying IPDR Streaming Protocol 1361
Verifying the IPDR Collector 1361
Verifying IPDR exporter 1361
Verifying IPDR session 1362
Verifying IPDR Session Collector 1362
Verifying IPDR Session Template 1362
Additional References 1363
Feature Information for IPDR Streaming Protocol 1363

CHAPTER 89 Usage-Based Billing (SAMIS) 1365
Hardware Compatibility Matrix for Cisco cBR Series Routers 1365
Prerequisites for Usage-Based Billing (SAMIS) 1366

Restrictions for Usage-based Billing 1367
Information About Usage-based Billing 1368
Feature Overview 1368
Usage-Based Billing and DOCSIS Support on the Cisco CMTS Routers 1368
Standards 1369
IPDR Service Definition Schemas 1369
IPDR CM-STATUS-2008 1370
DOCSIS SAMIS Service Definitions 1370
DOCSIS Diagnostic Log Service Definitions 1371
DOCSIS Spectrum Measurement Service Definition 1371
DOCSIS CMTS CM Registration Status Service Definition 1372
DOCSIS CMTS CM Upstream Status Service Definition 1372
DOCSIS CMTS Topology Service Definition 1372
DOCSIS CPE Service Definition 1372
DOCSIS CMTS Utilization Statistics Service Definition 1373
Modes of Operation 1373
Billing Record Format 1374
SNMP Support 1377
Benefits 1378
How to Configure the Usage-based Billing Feature 1378
Enabling Usage-based Billing Feature File Mode Using CLI Commands 1378
Enabling Usage-based Billing Feature File Mode Using SNMP Commands 1380
Examples for Enabling Usage Billing using SNMP Mode 1382
Enabling Usage-based Billing Feature Streaming Mode Using CLI Commands 1383
Enabling Usage-based Billing Feature Streaming Mode Using SNMP Commands 1385
Examples for SNMP Commands 1388
Enabling and Configuring the Secure Copy Protocol (optional) 1389
Configuring the Cisco CMTS for SSL Operation 1391
Prerequisites for CA 1392
Retrieving Records from a Cisco CMTS in File Mode 1392
Using SCP 1393
Using TFTP 1393
Using SNMP 1394
Using SNMP 1398

Examples To Transfer Using SNMP 1399
Disabling the Usage-based Billing Feature 1401
Configuring Certified SSL Servers for Usage-Based Billing 1402
Generating SSL Server Certification 1402
Configuring and Testing the Cisco CMTS for Certified SSL Server Support 1403
Monitoring the Usage-based Billing Feature 1404
Configuration Examples for Usage-based Billing 1406
File Mode Configuration (with Secure Copy) 1406
Non-Secure Streaming Mode Configuration 1406
Secure Streaming Mode Configuration 1407

CHAPTER 90 **Frequency Allocation Information for the Cisco CMTS Routers** 1409
Frequency Allocation for the Cisco CMTS Routers 1409

CHAPTER 91 **Flap List Troubleshooting** 1421
Finding Feature Information 1421
Hardware Compatibility Matrix for Cisco cBR Series Routers 1422
Prerequisites for Flap List Troubleshooting 1422
Restrictions for Flap List Troubleshooting 1422
Information About Flap List Troubleshooting 1423
Feature Overview 1423
Information in the Flap List 1423
Cisco Cable Manager and Cisco Broadband Troubleshooter 1424
Benefits 1425
How to Configure Flap List Troubleshooting 1425
Configuring Flap List Operation Using the CLI (optional) 1425
Clearing the Flap List and Counters Using the CLI (optional) 1427
Enabling or Disabling Power Adjustment Using the CLI (optional) 1428
Configuring Flap List Operation Using SNMP (optional) 1429
Clearing the Flap List and Counters Using SNMP (optional) 1430
How to Monitor and Troubleshoot Using Flap Lists 1431
Displaying the Flap List Using the show cable flap-list Command 1431
Displaying the Flap List Using the show cable modem flap Command 1431
Displaying the Flap List Using SNMP 1432

Displaying Flap-List Information for Specific Cable Modems 1433
Example 1434
Troubleshooting Suggestions 1434
Troubleshooting Tips 1435
Performing Amplitude Averaging 1435
Using Other Related Commands 1436
Configuration Examples for Flap List Troubleshooting 1437
Additional References 1437
Feature Information for Flap List Troubleshooting 1439

CHAPTER 92 **Maximum CPE and Host Parameters** 1441
Finding Feature Information 1441
Hardware Compatibility Matrix for Cisco cBR Series Routers 1441
Information About the MAX CPE and Host Parameters 1442
MAX CPE 1443
MAX Host 1444
Specifying an Unlimited Value for Max Host 1444
MAX CPE IP 1444
MAX CPE IPv6 1445
Interoperation of the Maximum CPE Parameters 1445
Benefits 1446
How to Configure the MAX CPE and Host Parameters 1446
Configuring the Maximum Number of CPE Devices on the Cisco CMTS 1446
Configuration Examples 1448
Additional References 1449
Feature Information for Maximum CPE and Host Parameters 1450

CHAPTER 93 **SNMP Background Synchronization** 1451
Information About SNMP Background Synchronization 1451
How to Configure SNMP Background Synchronization 1452
Enabling SNMP Background Synchronization 1452
Setting Data Interval 1452
Verifying SNMP Background Synchronization 1453
Configuring Example for SNMP Background Synchronization 1459

Reference 1459

CHAPTER 94 **Online Offline Diagnostics 1461**

Overview of Online Offline Diagnostics 1461

Benefits of Online Offline Diagnostics 1461

Prerequisites for Online Offline Diagnostics 1462

Restrictions for Online Offline Diagnostics 1462

How to Configure Online Offline Diagnostics 1462

Configuring Field Diagnostic Test 1462

Verifying the Testing Process 1462

Removing the Field Diagnostic Image from a Line Card 1463

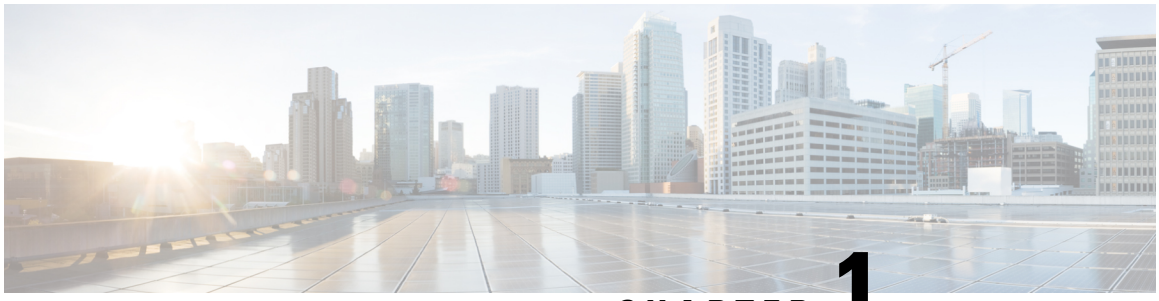
Configuration Example for Online Offline Diagnostics 1463

Feature Information for Online Offline Diagnostics 1463

PART I

Basic Configuration

- Start Up Configuration of the Cisco cBR Router, on page 1
- Cisco Smart Licensing, on page 31
- Consolidated Packages and SubPackages Management, on page 55



CHAPTER 1

Start Up Configuration of the Cisco cBR Router

This document describes the basic start up configuration tasks that must be completed on a Cisco cBR Series Converged Broadband Router .

- Prerequisites for Configuring the Cisco CMTS, on page 2
- Booting and Logging onto the Cisco CMTS , on page 3
- First Time Boot Up with ROMMON, on page 3
- Configuration Register, on page 4
- Setting Environment Variables, on page 5
- Unsetting Environment Variables, on page 5
- Booting from the TFTP on the Cisco cBR, on page 6
- Listing Supported Devices, on page 6
- Booting from the Device on the Cisco cBR, on page 7
- Setting AUTOBOOT image in ROMMON, on page 7
- Verifying the ROMMON Version, on page 8
- Resetting the Cisco cBR, on page 8
- File Systems, on page 9
- Verification of Hardware Bring Up, on page 10
- Gigabit Ethernet Management Interface Overview, on page 17
- Gigabit Ethernet Port Numbering, on page 18
- IP Address Handling in ROMMON and the Management Ethernet Port, on page 18
- Gigabit Ethernet Management Interface VRF, on page 18
- Common Ethernet Management Tasks, on page 19
- Viewing the VRF Configuration, on page 19
- Setting a Default Route in the Management Ethernet Interface VRF, on page 19
- Setting the Management Ethernet IP Address, on page 19
- Telnetting over the Management Ethernet Interface, on page 20
- Pinging over the Management Ethernet Interface, on page 20
- Copy Using TFTP or FTP, on page 20
- NTP Server, on page 20
- SYSLOG Server, on page 21
- SNMP-Related Services, on page 21
- Domain Name Assignment, on page 21
- DNS service, on page 21
- RADIUS or TACACS+ Server, on page 21

• VTY lines with ACL, on page 22
• Configuring the AUX Port for Network Management , on page 22
• Preprovisioning the Supervisor in the Cisco cBR Chassis, on page 22
• Configuring the Gigabit Ethernet Interface for Network Management, on page 23
• Configuring the DTI Port on the Supervisor PIC, on page 24
• Configuring the TenGigabit Ethernet Interface for Network Management , on page 25
• Connecting the New Router to the Network , on page 25
• Setting Password Protection on the Cisco CMTS, on page 26
• Recovering Lost Password on the Cisco CMTS, on page 27
• Saving Your Configuration Settings, on page 29
• Reviewing Your Settings and Configurations, on page 30

Prerequisites for Configuring the Cisco CMTS

Complete these prerequisite steps before you power on and configure the Cisco CMTS:

- Ensure that your network supports reliable broadband data transmission. Your plant must be swept, balanced, and certified based on National Television Standards Committee (NTSC) or appropriate international cable plant recommendations. Ensure your plant meets all Data-over-Cable Service Interface Specifications (DOCSIS) downstream and upstream radio frequency (RF) requirements.
- Ensure that your Cisco CMTS is installed according to the instructions in the hardware installation guide available on Cisco.com.
- Ensure that all other required headend or distribution hub routing and network interface equipment is installed, configured, and operational (based on the supported services). This includes:
  - All routers
  - Servers (Dynamic Host Configuration Protocol (DHCP) servers, Trivial File Transfer Protocol (TFTP) servers, and time-of-day (ToD) servers)
  - Network management systems
  - Other configuration or billing systems
- Ensure that DHCP and DOCSIS configuration files have been created and pushed to appropriate servers so that each CM, when initialized, can:
  - Transmit a DHCP request
  - Receive an IP address
  - Obtain TFTP and ToD server addresses
  - Download a DOCSIS configuration file (or updated software image if using Cisco uBR924 cable access routers or Cisco uBR910 cable data service units (DSUs) in your network)
- Ensure that customer premises equipment (CPE)—CMs or set-top boxes (STBs), PCs, telephones, or facsimile machines—meet requirements for your network and service offerings.
- Be familiar with your channel plan to assign appropriate frequencies. Outline your strategies for setting up bundling, if applicable to your headend or distribution hub. As appropriate, obtain:
  - Passwords
  - IP addresses
  - Subnet masks
  - Device names

After these prerequisites are met, you are ready to configure the Cisco CMTS. This includes, at a minimum:

- Configuring a host name and password for the Cisco CMTS
- Configuring the CMTS to support IP over the cable plant and network backbone

Note If you plan to use service-class-based provisioning, the service classes must be configured at the CMTS before CMs attempt to make a connection.

Note Do not configure the **logging event link-status** command during system initialization. It may take long time or even stop the standby SUP from booting up.

Booting and Logging onto the Cisco CMTS

The Cisco CMTS is administered using the Cisco command interpreter, called the EXEC. You must boot and log in to the router before you can enter an EXEC command.

Procedure

Step 1 Connect to the console port on the Supervisor PIC and the Supervisor card.

Step 2 Establish a terminal session. You can open terminal application (Hyper Terminal) on a PC as follows:

a) Connect using: Direct to Com 1
b) Set bits per second:9600
c) Set data bits: 8
d) Set parity: none
e) Set stop bit: 1
f) Set flow control: none

Type **no** when the following message is displayed:

```
Would you like to enter the initial dialog?[yes]: no
Router>
```

First Time Boot Up with ROMMON

The Cisco cBR-8 boots up with ROMMON on the console with 9600 baud default configuration. It boots image either from TFTP or from local device. Local devices supported include the bootflash and USB.

Example of the boot up display:

```
Initializing Hardware ...˘
```

```
System Bootstrap, Version 15.5(2r)S, RELEASE SOFTWARE
Copyright (c) 1994-2015  by cisco Systems, Inc.

Current image running: Boot ROM0

Last reset cause: PowerOn

CPUID: 0x000206d7
UCODE: 0x00000710_00000000
Viper version register: 0x14121111
Set Chassis Type to 13RU
Cisco cBR-8  platform with 50331648 Kbytes of main memory

rommon 1 >
```

Configuration Register

The **confreg** ROMMON command displays the configuration and allows modification of the settings.

```
rommon > confreg

           Configuration Summary
   (Virtual Configuration Register: 0x0)
enabled are:
 [ 0 ] break/abort has effect
 [ 1 ] console baud: 9600
 boot: ...... the ROM Monitor

do you wish to change the configuration? y/n  [n]:  y
 enable  "diagnostic mode"? y/n  [n]:
 enable  "use net in IP bcast address"? y/n  [n]:
 enable  "load rom after netboot fails"? y/n  [n]:
 enable  "use all zero broadcast"? y/n  [n]:
 disable "break/abort has effect"? y/n  [n]:
 enable  "ignore system config info"? y/n  [n]:
 change console baud rate? y/n  [n]:
change the boot characteristics? y/n  [n]:

           Configuration Summary
   (Virtual Configuration Register: 0x0)
enabled are:
 [ 0 ] break/abort has effect
 [ 1 ] console baud: 9600
 boot: ...... the ROM Monitor
do you wish to change the configuration? y/n  [n]:
Console baud rate options:
change console baud rate? y/n  [n]:  y
0=9600, 1=4800, 2=1200, 3=2400, 4=19200, 5=38400, 6=57600, 7=115200
enter rate  [0]:
Boot characteristics options:
change the boot characteristics? y/n  [n]:  y

 enter to boot:
 0 = ROM Monitor
 1 = the boot helper image
 2-15 = boot system
 [0]:
```

Setting Environment Variables

No Environment variables are required to boot the Cisco IOS-XE image.

There are variables set by default. The ROMMON command **set** displays the default variables.

```
rommon  > set
PS1=rommon ! >
?=0
rommon  >
```

To set a variable, the format is VARIABLE="value".

The**set** command displays the new variable and the **sync** command saves the variable to NVRAM.

Note If the variable value has a space in between, specify the value within quotes.

```
rommon  > set
PS1=rommon ! >
?=0
rommon  > IP_ADDRESS=1.2.3.4
rommon  > IP_SUBNET_MASK=255.255.255.128
rommon  > DEFAULT_GATEWAY=1.2.9.10
rommon  > TFTP_SERVER=1.2.3.6
rommon  > sync
```

Unsetting Environment Variables

The **unset** ROMMON command removes the Environment variables and the **sync** command saves the variable to NVRAM.

```
rommon 1 > set
PS1=rommon ! >
?=0
BSI=0
BOOT=bootflash:cbrsup-adventerprisek9.SSA.bin,12;
RANDOM_NUM=1357042312
RET_2_RTS=17:45:06 PDT Sat Dec 31 2011
RET_2_RCALTS=1325378706
rommon 2 > unset BOOT
rommon 3 > sync
rommon 4 > set
PS1=rommon ! >
?=0
BSI=0
RANDOM_NUM=1357042312
RET_2_RTS=17:45:06 PDT Sat Dec 31 2011
RET_2_RCALTS=1325378706
rommon 5 >
```

Booting from the TFTP on the Cisco cBR

ROMMON boots up with default environment variables. The BinOS image is booted up from TFTP over the management port. This requires a minimum set of environment variables: IP_ADDRESS, IP_SUBNET_MASK, DEFAULT_GATEWAY, and TFTP_SERVER.

Procedure

Step 1 Type the **set** command and define the required environment variables.

```
rommon  > set
PS1=rommon ! >
?=0
rommon  > IP_ADDRESS=1.2.3.4
rommon  > IP_SUBNET_MASK=255.255.255.128
rommon  > DEFAULT_GATEWAY=1.2.9.10
rommon  > TFTP_SERVER=1.2.3.6
rommon  > sync
```

Step 2 Type the **sync** command to save the variables to NVRAM.

```
rommon 6 > sync
```

Step 3 Type the **boot** command to load the image.

```
rommon 7 > boot tftp:/tftpboot/username/cbrsup-universalk9.SSA.bin

          IP_ADDRESS: 1.2.3.4
      IP_SUBNET_MASK: 255.255.255.128
     DEFAULT_GATEWAY: 1.2.9.10
         TFTP_SERVER: 1.2.3.6
           TFTP_FILE: /tftpboot/username/cbrsup-universalk9.SSA.bin
        TFTP_MACADDR: c4:14:3c:17:e8:00
        TFTP_VERBOSE: Progress
    TFTP_RETRY_COUNT: 18
        TFTP_TIMEOUT: 7200
       TFTP_CHECKSUM: Yes
          ETHER_PORT: 2

    ETHER_SPEED_MODE: Auto Detect
link up........
Receiving /tftpboot/username/cbrsup-universalk9.SSA.bin from 172.19.211.47
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Listing Supported Devices

The **dev** command lists the devices supported on the router.

```
rommon 1 > dev
Devices in device table:
        id  name
```

```
 harddisk:  Internal hard disk
bootflash:  Internal flash drive
     usb0:  External USB drive 0
     usb1:  External USB drive 1
rommon 2 >
```

Booting from the Device on the Cisco cBR

Procedure

Step 1 Type the **dir bootflash:** command.

```
rommon  > dir bootflash:
File System: EXT2/EXT3

12         691955580 -rw-r--r--     cbrsup-xe315.SSA.bin
45         83475     -rw-r--r--     reload.log.20120103004502
```

Step 2 Type the **boot bootflash:***imagename* command.

```
rommon  > boot bootflash:cbrsup-xe315.bin
File size is 0x293e67bc
Located cbrsup-xe315.bin
Image size 691955644 inode num 145153, bks cnt 168935 blk size 8*512
###################################################################
```

Setting AUTOBOOT image in ROMMON

To set AUTOBOOT of an image from bootflash:, add the Environment Variable BOOT and then change the configuration register boot characteristics to boot and reset the system.

Procedure

Step 1 Type the **boot=bootflash:***imagename* command to load the image.

```
rommon  > BOOT=bootflash:cbrsup-xe315-20150131.bin
```

Step 2 Type the **sync** command to copy the variables to NVRAM.

```
rommon  > sync
```

Step 3 Type the **confreg** command to configure and modify the settings.

```
rommon  > confreg

          Configuration Summary
   (Virtual Configuration Register: 0x0)
enabled are:
 [ 0 ] break/abort has effect
```

```
 [ 1 ] console baud: 9600
 boot: ...... the ROM Monitor

do you wish to change the configuration? y/n  [n]:  y
 enable  "diagnostic mode"? y/n  [n]:
 enable  "use net in IP bcast address"? y/n  [n]:
 enable  "load rom after netboot fails"? y/n  [n]:
 enable  "use all zero broadcast"? y/n  [n]:
 disable "break/abort has effect"? y/n  [n]:
 enable  "ignore system config info"? y/n  [n]:
 change console baud rate? y/n  [n]:  n
change the boot characteristics? y/n  [n]:  y

 enter to boot:
 0 = ROM Monitor
 1 = the boot helper image
 2-15 = boot system
 [0]:  2

           Configuration Summary
   (Virtual Configuration Register: 0x2)
enabled are:
 [ 0 ] break/abort has effect
 [ 1 ] console baud: 9600
 boot: ...... image specified by the boot system commands or default to: cisco2-Cisco cBR-8


do you wish to change the configuration? y/n  [n]:

You must reset or power cycle for new config to take effect
```

Step 4 Type the **reset** command for the new configuration to take effect.

```
rommon  > reset
```

What to do next

Verifying the ROMMON Version

Use the **showmon** command to display the version of ROMMON.

```
rommon  > showmon
Current image running (0/1): Boot ROM0
System Bootstrap, Version 15.5(2r)S, RELEASE SOFTWARE
Copyright (c) 1994-2015  by cisco Systems, Inc.

Viper version register: 0x14121111
rommon  >
```

Resetting the Cisco cBR

Use the **reset** command to soft reset the Supervisor.

```
rommon  > reset
```

```
Resetting .......


Initializing Hardware ...˘

System Bootstrap, Version 15.5(2r)S, RELEASE SOFTWARE
Copyright (c) 1994-2015  by cisco Systems, Inc.

Current image running: Boot ROM0

Last reset cause: LocalSoftware

CPUID: 0x000206d7
UCODE: 0x00000710_00000000
Viper version register: 0x14121111
Set Chassis Type to 13RU
Cisco cBR-8  platform with 50331648 Kbytes of main memory

rommon  >
```

File Systems

The Cisco cBR-8 router runs on the Cisco IOS-XE image. Supported file systems include:

1. IOS File System (IFS) in IOS
2. ext2, vfs, jffs2, tmpfs, autofs, and such common file systems in Linux

Features of the File Systems:

1. Both the Harddisk and USB are hot pluggable.
2. Harddisk is not accessible under Rommon.
3. Bootflash and USB disk are accessible under Rommon.
4. The **dir**, **show**, **copy**, **delete**, **mkdir**, **rmdir**, and **fsck** commands are supported for bootflash, harddisk and USB.

File System Table in the Supervisor

Name	Device	Size	Type	Visible	Usage	Physical Description
bootflash	/dev/bootflash1	7800705024	ext2	IOS/Binos	image,IOScrasinfo,etc	Partition1 of bootflash (eUSB flash).
flash	/dev/bootflash1	7800705024	ext2	IOS	image	A copy of bootflash.
nvram	/dev/bootflash2	32M	N/A	IOS	configuretion, etc	Partition2 of bootflash (eUSB flash).
harddisk	/dev/harddisk1	98394218496	ext2	IOS/Binos	tracelog,corefile,etc	Partition1 of the 100G harddisk.
usb0	/dev/usb11	8G	vfat	IOS/Binos	image	Two USBs can be inserted into one SUP.

Verification of Hardware Bring Up

Monitoring the Cisco cBR Chassis Using CLI

- **show platform**—Verify if the installed cards are in **Ok** or **Inserted** state.

```
Router# show platform

Chassis type: CBR-8-CCAP-CHASS

Slot      Type                State                 Insert time (ago)
--------- ------------------- --------------------- -----------------
1         CBR-CCAP-LC-40G     ok                    03:22:58
1/1       CBR-RF-PIC          ok                    03:19:40
SUP0      CBR-CCAP-SUP-160G   inserted              03:22:58
 R0                           ok, active
 F0                           ok, active
 4                            ok, active
 4/1      CBR-SUP-8X10G-PIC   ok                    03:20:30
P0        PWR-2KW-DC-V2       ok                    03:21:20
P1        PWR-2KW-DC-V2       ok                    03:21:20
P2        PWR-2KW-DC-V2       ok                    03:21:20
P3        PWR-2KW-DC-V2       ok                    03:21:20
P4        PWR-2KW-DC-V2       ok                    03:21:20
P5        PWR-2KW-DC-V2       ok                    03:21:20
P10       CBR-FAN-ASSEMBLY    ok                    03:21:10
P11       CBR-FAN-ASSEMBLY    ok                    03:21:10
P12       CBR-FAN-ASSEMBLY    ok                    03:21:10
P13       CBR-FAN-ASSEMBLY    ok                    03:21:10
P14       CBR-FAN-ASSEMBLY    ok                    03:21:10
```

- **show platform hardware slot** *slot* **serdes status**—Verify if all the links are in **locked** state.

```
Router# show platform hardware slot F1 serdes status

Slot R1-Link A
  RX link locked
  58-bit scrambler, 20 Gbps
  0 Overruns, 0 Underruns
  0 Reframe, 0 Disparity
  0 Out of band, 0 Illegal control codes

Slot 3-Link A
  RX link locked
  58-bit scrambler, 20 Gbps
  0 Overruns, 0 Underruns
  0 Reframe, 0 Disparity
  0 Out of band, 0 Illegal control codes

Slot 5-Link A
  RX link locked
  58-bit scrambler, 20 Gbps
  0 Overruns, 0 Underruns
  0 Reframe, 0 Disparity
  0 Out of band, 0 Illegal control codes

Slot 5-Link B
  RX link locked
  58-bit scrambler, 20 Gbps
  0 Overruns, 0 Underruns
```

```
  0 Reframe, 0 Disparity
  0 Out of band, 0 Illegal control codes

Slot 5-Link C
  RX link locked
  58-bit scrambler, 20 Gbps
  0 Overruns, 0 Underruns
  0 Reframe, 0 Disparity
  0 Out of band, 0 Illegal control codes

Slot 5-Link D
  RX link locked
  58-bit scrambler, 20 Gbps
  0 Overruns, 0 Underruns
  0 Reframe, 0 Disparity
  0 Out of band, 0 Illegal control codes

Slot 5-Link E
  RX link Init
  58-bit scrambler, 20 Gbps
  0 Overruns, 0 Underruns
  0 Reframe, 0 Disparity
  0 Out of band, 0 Illegal control codes

Slot 5-Link F
  RX link Init
  58-bit scrambler, 20 Gbps
  0 Overruns, 0 Underruns
  0 Reframe, 0 Disparity
  0 Out of band, 0 Illegal control codes

Slot 5-Link G
  RX link Init
  58-bit scrambler, 20 Gbps
  0 Overruns, 0 Underruns
  0 Reframe, 0 Disparity
  0 Out of band, 0 Illegal control codes

Slot 5-Link H
  RX link Init
  58-bit scrambler, 20 Gbps
  0 Overruns, 0 Underruns
  0 Reframe, 0 Disparity
  0 Out of band, 0 Illegal control codes
```

- **show environment all**—Verify the environmental status of each FRU after installation.

  This command displays the system temperature, voltage, fan, and power supply conditions.

```
Router# show environment all

Sensor List:  Environmental Monitoring
 Sensor           Location          State             Reading
 AVCC&1P2: Sens   4/1               Normal            81 mV
 AVCC&1P2: Vin    4/1               Normal            12600 mV
 AVCC&1P2: ADin   4/1               Normal            0 mV
 VP1P35: Sens     4/1               Normal            8 mV
 VP1P35: Vin      4/1               Normal            12650 mV
 VP1P35: ADin     4/1               Normal            112 mV
 VP1P0: Sens      4/1               Normal            15 mV
 VP1P0: Vin       4/1               Normal            12625 mV
 VP1P0: ADin      4/1               Normal            0 mV
 MGTAVTT: Sens    4/1               Normal            21 mV
 MGTAVTT: Vin     4/1               Normal            12625 mV
```

```
MGTAVTT: ADin    4/1              Normal           0 mV
VP1P8: Sens      4/1              Normal           41 mV
VP1P8: Vin       4/1              Normal           12600 mV
VP1P8: ADin      4/1              Normal           0 mV
VP3P3: Sens      4/1              Normal           39 mV
VP3P3: Vin       4/1              Normal           12625 mV
VP3P3: ADin      4/1              Normal           0 mV
Temp: RTMAC      4/1              Normal           34 Celsius
Temp: INLET      4/1              Normal           29 Celsius
Temp: OUTLET     4/1              Normal           27 Celsius
Temp: MAX6697    4/1              Normal           50 Celsius
Temp: TCXO       4/1              Normal           37 Celsius
Temp: SUP_OUT    4/1              Normal           49 Celsius
Temp: 3882_1 P   4/1              Normal           44 Celsius
Temp: 3882_2 P   4/1              Normal           39 Celsius
Temp: 3882_3 P   4/1              Normal           39 Celsius
VP5P0: Sens      4/1              Normal           6 mV
VP5P0: Vin       4/1              Normal           12650 mV
VP5P0: ADin      4/1              Normal           0 mV
VP1P8: Sens      4/1              Normal           33 mV
VP1P8: Vin       4/1              Normal           12625 mV
VP1P8: ADin      4/1              Normal           0 mV
3P3&1P0: Sens    4/1              Normal           24 mV
3P3&1P0: Vin     4/1              Normal           12625 mV
3P3&1P0: ADin    4/1              Normal           0 mV
Temp: INLET PD   4/1              Normal           27 Celsius
Temp: OUTLETPD   4/1              Normal           36 Celsius
Temp: 6697-DC    4/1              Normal           38 Celsius
Temp: PHYOUT     4/1              Normal           49 Celsius
Temp: PHYIN      4/1              Normal           38 Celsius
Temp: SSD        4/1              Normal           40 Celsius
Temp: SFP+       4/1              Normal           36 Celsius
Temp: 3882_1PD   4/1              Normal           42 Celsius
3882_PC1_0: VO   4/1              Normal           1198 mV
3882_PC1_1: VO   4/1              Normal           999 mV
3882_PC2_0: VO   4/1              Normal           998 mV
3882_PC3_0: VO   4/1              Normal           1349 mV
PSOC-PC1_0: VO   4/1              Normal           3300 mV
PSOC-PC1_1: VO   4/1              Normal           12590 mV
PSOC-PC1_2: VO   4/1              Normal           6997 mV
PSOC-PC1_3: VO   4/1              Normal           5000 mV
PSOC-PC1_4: VO   4/1              Normal           3299 mV
PSOC-PC1_5: VO   4/1              Normal           1000 mV
PSOC-PC1_6: VO   4/1              Normal           1010 mV
PSOC-PC1_7: VO   4/1              Normal           1801 mV
PSOC-PC1_8: VO   4/1              Normal           2000 mV
PSOC-PC1_9: VO   4/1              Normal           1198 mV
PSOC-PC1_10: V   4/1              Normal           1798 mV
PSOC-PC1_11: V   4/1              Normal           2500 mV
PSOC-PC1_12: V   4/1              Normal           1353 mV
PSOC-PC1_13: V   4/1              Normal           1223 mV
PSOC-PC1_14: V   4/1              Normal           592 mV
PSOC-PC1_15: V   4/1              Normal           596 mV
3882_PDC_0: VO   4/1              Normal           1000 mV
3882_PDC_1: VO   4/1              Normal           3300 mV
PSOC-DC1_0: VO   4/1              Normal           4998 mV
PSOC-DC1_1: VO   4/1              Normal           3280 mV
PSOC-DC1_2: VO   4/1              Normal           1005 mV
PSOC-DC1_3: VO   4/1              Normal           1801 mV
PSOC-DC1_4: VO   4/1              Normal           2500 mV
12_CUR: Sens     9                Normal           14 mV
12_CUR: Vin      9                Normal           12650 mV
12_CUR: ADin     9                Normal           267 mV
G0_CUR: Sens     9                Normal           69 mV
```

```
G0_CUR: Vin      9                Normal           12550 mV
G0_CUR: ADin     9                Normal           0 mV
G1_CUR: Sens     9                Normal           69 mV
G1_CUR: Vin      9                Normal           12575 mV
G1_CUR: ADin     9                Normal           0 mV
LB_CUR: Sens     9                Normal           11 mV
LB_CUR: Vin      9                Normal           12525 mV
LB_CUR: ADin     9                Normal           0 mV
Temp: CAPRICA    9                Normal           40 Celsius
Temp: BASESTAR   9                Normal           47 Celsius
Temp: RAIDER     9                Normal           45 Celsius
Temp: CPU        9                Normal           31 Celsius
Temp: INLET      9                Normal           25 Celsius
Temp: OUTLET     9                Normal           35 Celsius
Temp: DIGITAL    9                Normal           31 Celsius
Temp: UPX        9                Normal           29 Celsius
Temp: LEOBEN1    9                Normal           31 Celsius
Temp: LEOBEN2    9                Normal           35 Celsius
Temp: 3.3-18     9                Normal           43 Celsius
Temp: BS_1V      9                Normal           45 Celsius
Freq: 5338-49    9                Normal           0 MHz
Freq: 5338-52    9                Normal           0 MHz
Freq: 5338-89    9                Normal           0 MHz
3882_1_0: VOUT   9                Normal           3299 mV
3882_1_1: VOUT   9                Normal           1800 mV
3882_2_0: VOUT   9                Normal           2500 mV
3882_2_1: VOUT   9                Normal           1199 mV
3882_3_0: VOUT   9                Normal           1419 mV
3882_4_0: VOUT   9                Normal           1350 mV
3882_5_0: VOUT   9                Normal           1000 mV
3882_6_0: VOUT   9                Normal           1021 mV
3882_7_0: VOUT   9                Normal           1199 mV
3882_7_1: VOUT   9                Normal           1000 mV
3882_8_0: VOUT   9                Normal           1000 mV
3882_9_0: VOUT   9                Normal           999 mV
V2978: VSENSE0   9                Normal           0 mV
V2978: VSENSE1   9                Normal           0 mV
V2978: VSENSE2   9                Normal           0 mV
V2978: VSENSE3   9                Normal           6000 mV
V2978: VSENSE4   9                Normal           2400 mV
V2978: VSENSE5   9                Normal           0 mV
V2978: VSENSE6   9                Normal           6598 mV
V2978: VSENSE7   9                Normal           4998 mV
V2978: VIN       9                Normal           25218 mV
PSOC_2_0: VOUT   9                Normal           12582 mV
PSOC_2_1: VOUT   9                Normal           4985 mV
PSOC_2_2: VOUT   9                Normal           3256 mV
PSOC_2_3: VOUT   9                Normal           1982 mV
PSOC_2_4: VOUT   9                Normal           1990 mV
PSOC_2_5: VOUT   9                Normal           1782 mV
PSOC_2_6: VOUT   9                Normal           1793 mV
PSOC_2_7: VOUT   9                Normal           1786 mV
PSOC_2_8: VOUT   9                Normal           1483 mV
PSOC_2_9: VOUT   9                Normal           1193 mV
PSOC_2_10: VOU   9                Normal           995 mV
PSOC_2_11: VOU   9                Normal           987 mV
PSOC_2_12: VOU   9                Normal           994 mV
PSOC_2_13: VOU   9                Normal           707 mV
PSOC_2_14: VOU   9                Normal           592 mV
PSOC_2_15: VOU   9                Normal           593 mV
LTC4261: Power   9                Normal           340 Watts
PEM Iout         P0               Normal           5 A
PEM Vout         P0               Normal           55 V DC
PEM Vin          P0               Normal           202 V AC
```

```
Temp: INLET      P0                Normal            26 Celsius
Temp: OUTLET     P0                Normal            48 Celsius
PEM Iout         P1                Normal            6 A
PEM Vout         P1                Normal            55 V DC
PEM Vin          P1                Normal            204 V AC
Temp: INLET      P1                Normal            30 Celsius
Temp: OUTLET     P1                Normal            53 Celsius
PEM Iout         P2                Normal            3 A
PEM Vout         P2                Normal            55 V DC
PEM Vin          P2                Normal            204 V AC
Temp: INLET      P2                Normal            25 Celsius
Temp: OUTLET     P2                Normal            51 Celsius
PSOC-MB2_0: VO   R0                Normal            12758 mV
PSOC-MB2_1: VO   R0                Normal            4998 mV
PSOC-MB2_2: VO   R0                Normal            7082 mV
PSOC-MB2_3: VO   R0                Normal            3287 mV
PSOC-MB2_4: VO   R0                Normal            989 mV
PSOC-MB2_5: VO   R0                Normal            1047 mV
PSOC-MB2_6: VO   R0                Normal            1500 mV
PSOC-MB2_7: VO   R0                Normal            1800 mV
PSOC-MB2_8: VO   R0                Normal            914 mV
PSOC-MB2_9: VO   R0                Normal            885 mV
PSOC-MB2_10: V   R0                Normal            994 mV
PSOC-MB2_11: V   R0                Normal            989 mV
PSOC-MB2_12: V   R0                Normal            1479 mV
PSOC-MB2_13: V   R0                Normal            989 mV
PSOC-MB2_14: V   R0                Normal            984 mV
PSOC-MB2_15: V   R0                Normal            890 mV
PSOC-MB2_16: V   R0                Normal            2485 mV
PSOC-MB2_17: V   R0                Normal            1346 mV
PSOC-MB2_18: V   R0                Normal            1458 mV
PSOC-MB2_19: V   R0                Normal            1208 mV
PSOC-MB2_20: V   R0                Normal            1791 mV
PSOC-MB2_21: V   R0                Normal            3293 mV
PSOC-MB2_22: V   R0                Normal            3250 mV
PSOC-MB2_23: V   R0                Normal            3284 mV
PSOC-MB2_24: V   R0                Normal            4970 mV
PSOC-MB2_25: V   R0                Normal            4451 mV
PSOC-MB3_0: VO   R0                Normal            4983 mV
PSOC-MB3_1: VO   R0                Normal            4979 mV
PSOC-MB3_2: VO   R0                Normal            1500 mV
PSOC-MB3_3: VO   R0                Normal            1192 mV
PSOC-MB3_4: VO   R0                Normal            705 mV
PSOC-MB3_5: VO   R0                Normal            752 mV
PSOC-MB3_6: VO   R0                Normal            579 mV
PSOC-MB3_7: VO   R0                Normal            1500 mV
PSOC-MB3_8: VO   R0                Normal            1501 mV
PSOC-MB3_9: VO   R0                Normal            1250 mV
PSOC-MB3_10: V   R0                Normal            1247 mV
PSOC-MB3_11: V   R0                Normal            1260 mV
PSOC-MB3_12: V   R0                Normal            1038 mV
PSOC-MB3_13: V   R0                Normal            1343 mV
PSOC-MB3_14: V   R0                Normal            670 mV
PSOC-MB3_15: V   R0                Normal            1800 mV
PSOC-MB3_16: V   R0                Normal            908 mV
PSOC-MB3_17: V   R0                Normal            823 mV
PSOC-MB3_18: V   R0                Normal            992 mV
PSOC-MB3_19: V   R0                Normal            984 mV
PSOC-MB3_20: V   R0                Normal            1046 mV
PSOC-MB3_21: V   R0                Normal            1192 mV
PSOC-MB3_22: V   R0                Normal            1169 mV
PSOC-MB3_23: V   R0                Normal            1187 mV
PSOC-MB3_24: V   R0                Normal            1796 mV
PSOC-MB3_25: V   R0                Normal            1792 mV
```

```
PSOC-MB3_26: V   R0              Normal           1787 mV
PSOC-MB3_27: V   R0              Normal           1034 mV
3882_MB1_0: VO   R0              Normal           1001 mV
3882_MB1_1: VO   R0              Normal           1022 mV
3882_MB2_0: VO   R0              Normal           1197 mV
3882_MB3_0: VO   R0              Normal           1045 mV
3882_MB3_1: VO   R0              Normal           996 mV
3882_MB4_0: VO   R0              Normal           898 mV
3882_MB5_0: VO   R0              Normal           1348 mV
3882_MB6_0: VO   R0              Normal           1350 mV
3882_MB6_1: VO   R0              Normal           3297 mV
3882_MB7_0: VO   R0              Normal           998 mV
3882_MB8_0: VO   R0              Normal           1501 mV
3882_MB8_1: VO   R0              Normal           1551 mV
3882_MB9_0: VO   R0              Normal           999 mV
3882_MB9_1: VO   R0              Normal           3296 mV
15301_1: VOUT    R0              Normal           2500 mV
15301_2: VOUT    R0              Normal           1200 mV
15301_3: VOUT    R0              Normal           1200 mV
AS_VRM: Sens     R0              Normal           40 mV
AS_VRM: Vin      R0              Normal           12725 mV
AS_VRM: ADin     R0              Normal           0 mV
Y0_VRM: Sens     R0              Normal           23 mV
Y0_VRM: Vin      R0              Normal           12675 mV
Y0_VRM: ADin     R0              Normal           380 mV
CPU_VCC: Sens    R0              Normal           6 mV
CPU_VCC: Vin     R0              Normal           12725 mV
CPU_VCC: ADin    R0              Normal           0 mV
5P0_BIAS: Sens   R0              Normal           19 mV
5P0_BIAS: Vin    R0              Normal           12700 mV
5P0_BIAS: ADin   R0              Normal           0 mV
7P0_BIAS: Sens   R0              Normal           45 mV
7P0_BIAS: Vin    R0              Normal           12725 mV
7P0_BIAS: ADin   R0              Normal           0 mV
1P0_AA: Sens     R0              Normal           37 mV
1P0_AA: Vin      R0              Normal           12700 mV
1P0_AA: ADin     R0              Normal           0 mV
1P0_RT: Sens     R0              Normal           16 mV
1P0_RT: Vin      R0              Normal           12725 mV
1P0_RT: ADin     R0              Normal           0 mV
1P2: Sens        R0              Normal           37 mV
1P2: Vin         R0              Normal           12675 mV
1P2: ADin        R0              Normal           0 mV
0P9_T0: Sens     R0              Normal           7 mV
0P9_T0: Vin      R0              Normal           12750 mV
0P9_T0: ADin     R0              Normal           0 mV
1P05_CPU: Sens   R0              Normal           11 mV
1P05_CPU: Vin    R0              Normal           12700 mV
1P05_CPU: ADin   R0              Normal           0 mV
1P0_CC: Sens     R0              Normal           16 mV
1P0_CC: Vin      R0              Normal           12700 mV
1P0_CC: ADin     R0              Normal           0 mV
1P35_DDR: Sens   R0              Normal           6 mV
1P35_DDR: Vin    R0              Normal           12725 mV
1P35_DDR: ADin   R0              Normal           0 mV
1P35_RLD: Sens   R0              Normal           0 mV
1P35_RLD: Vin    R0              Normal           12675 mV
1P35_RLD: ADin   R0              Normal           2047 mV
3P3_CCC: Sens    R0              Normal           16 mV
3P3_CCC: Vin     R0              Normal           12700 mV
3P3_CCC: ADin    R0              Normal           1375 mV
1P0_R: Sens      R0              Normal           29 mV
1P0_R: Vin       R0              Normal           12700 mV
1P0_R: ADin      R0              Normal           0 mV
```

```
1P5_A0: Sens     R0              Normal          41 mV
1P5_A0: Vin      R0              Normal          12700 mV
1P5_A0: ADin     R0              Normal          0 mV
1P5: Sens        R0              Normal          34 mV
1P5: Vin         R0              Normal          12675 mV
1P5: ADin        R0              Normal          0 mV
2P5: Sens        R0              Normal          5 mV
2P5: Vin         R0              Normal          12700 mV
2P5: ADin        R0              Normal          0 mV
1P8_A: Sens      R0              Normal          10 mV
1P8_A: Vin       R0              Normal          12675 mV
1P8_A: ADin      R0              Normal          947 mV
1P0_BV: Sens     R0              Normal          24 mV
1P0_BV: Vin      R0              Normal          12700 mV
1P0_BV: ADin     R0              Normal          0 mV
3P3: Sens        R0              Normal          16 mV
3P3: Vin         R0              Normal          12725 mV
3P3: ADin        R0              Normal          0 mV
1P2_B: Sens      R0              Normal          41 mV
1P2_B: Vin       R0              Normal          12725 mV
1P2_B: ADin      R0              Normal          0 mV
ADM1075: Power   R0              Normal          329 Watts
Temp: Y0_DIE     R0              Normal          33 Celsius
Temp: BB_DIE     R0              Normal          29 Celsius
Temp: VP_DIE     R0              Normal          26 Celsius
Temp: RT-E_DIE   R0              Normal          31 Celsius
Temp: INLET_1    R0              Normal          23 Celsius
Temp: INLET_2    R0              Normal          22 Celsius
Temp: OUTLET_1   R0              Normal          25 Celsius
Temp: 3882_1     R0              Normal          46 Celsius
Temp: 3882_1A    R0              Normal          43 Celsius
Temp: 3882_1B    R0              Normal          43 Celsius
Temp: 3882_2     R0              Normal          41 Celsius
Temp: 3882_2A    R0              Normal          40 Celsius
Temp: 3882_2B    R0              Normal          41 Celsius
Temp: 3882_3     R0              Normal          37 Celsius
Temp: 3882_3A    R0              Normal          34 Celsius
Temp: 3882_3B    R0              Normal          33 Celsius
Temp: 3882_4     R0              Normal          46 Celsius
Temp: 3882_4A    R0              Normal          38 Celsius
Temp: 3882_4B    R0              Normal          35 Celsius
Temp: 3882_5     R0              Normal          32 Celsius
Temp: 3882_5A    R0              Normal          23 Celsius
Temp: 3882_5B    R0              Normal          23 Celsius
Temp: 3882_6     R0              Normal          37 Celsius
Temp: 3882_6A    R0              Normal          30 Celsius
Temp: 3882_6B    R0              Normal          32 Celsius
Temp: 3882_7     R0              Normal          38 Celsius
Temp: 3882_7A    R0              Normal          35 Celsius
Temp: 3882_7B    R0              Normal          35 Celsius
Temp: 3882_8     R0              Normal          47 Celsius
Temp: 3882_8A    R0              Normal          45 Celsius
Temp: 3882_8B    R0              Normal          41 Celsius
Temp: 3882_9     R0              Normal          37 Celsius
Temp: 3882_9A    R0              Normal          33 Celsius
Temp: 3882_9B    R0              Normal          32 Celsius
Temp: 8314_1     R0              Normal          40 Celsius
Temp: 8314_2     R0              Normal          36 Celsius
Temp: 3536_1A    R0              Normal          26 Celsius
Temp: 3536_1B    R0              Normal          26 Celsius
Temp: 15301_1A   R0              Normal          31 Celsius
Temp: 15301_1B   R0              Normal          32 Celsius
Temp: 15301_2A   R0              Normal          28 Celsius
Temp: 15301_2B   R0              Normal          34 Celsius
```

```
 Temp: 15301_3A   R0                Normal           38 Celsius
 Temp: 15301_3B   R0                Normal           45 Celsius
 Temp: AS_DIE     R0                Normal           70 Celsius
 Temp: XPT1_DTL   R0                Normal           42 Celsius
 Temp: XPT1_DTR   R0                Normal           42 Celsius
 Temp: XPT1_DBL   R0                Normal           42 Celsius
 Temp: XPT1_DBR   R0                Normal           42 Celsius
 Temp: XPT2_DTL   R0                Normal           42 Celsius
 Temp: XPT2_DTR   R0                Normal           42 Celsius
 Temp: XPT2_DBL   R0                Normal           42 Celsius
 Temp: XPT2_DBR   R0                Normal           42 Celsius
 Temp: XPT3_DTL   R0                Normal           42 Celsius
 Temp: XPT3_DTR   R0                Normal           42 Celsius
 Temp: XPT3_DBL   R0                Normal           42 Celsius
 Temp: XPT3_DBR   R0                Normal           42 Celsius
 Freq: MAX3674    R0                Normal           500 MHz
 Freq: SQ420D     R0                Normal           24 MHz
```

- **show facility-alarm status** —Verify the chassis status.

```
Router# show facility-alarm status

System Totals  Critical: 4  Major: 1  Minor: 8

Source                    Time                   Severity      Description [Index]
------                    ------                 --------      -------------------
slot 3/0                  Apr 13 2015 16:25:58   CRITICAL      Active Card Removed
OIR Alarm [0]
Power Supply Bay 3        Apr 13 2015 13:41:56   CRITICAL      Power Supply/FAN
Module Missing [0]
Power Supply Bay 4        Apr 13 2015 13:41:56   CRITICAL      Power Supply/FAN
Module Missing [0]
Power Supply Bay 5        Apr 13 2015 13:41:56   CRITICAL      Power Supply/FAN
Module Missing [0]
Cable3/0/15-US0           Apr 13 2015 17:32:53   MINOR         Physical Port Link
Down [0]
Cable3/0/15-US1           Apr 13 2015 17:32:53   MINOR         Physical Port Link
Down [0]
Cable3/0/15-US2           Apr 13 2015 17:32:53   MINOR         Physical Port Link
Down [0]
Cable3/0/15-US3           Apr 13 2015 17:32:53   MINOR         Physical Port Link
Down [0]
Cable3/0/15-US4           Apr 13 2015 17:32:53   MINOR         Physical Port Link
Down [0]
```

Gigabit Ethernet Management Interface Overview

The purpose of this interface is to allow users to perform management tasks on the router; it is basically an interface that should not and often cannot forward network traffic but can otherwise access the router, often via Telnet and SSH, and perform most management tasks on the router.

The following aspects of the Management Ethernet interface should be noted:

- Each SUP has a Management Ethernet interface, but only the active SUP has an accessible Management Ethernet interface (the standby SUP can be accessed using the console port, however).
- IPv4, IPv6, and ARP are the only routed protocols supported for the interface.
- The interface provides a method of access to the router even when some software processes are down.

- The Ethernet Management Interface cannot be used as a Lawful Intercept MD source interface.
- The Management Ethernet interface is part of its own VRF.

Gigabit Ethernet Port Numbering

The Gigabit Ethernet Management port is always GigabitEthernet0.

In a dual SUP configuration, the Management Ethernet interface on the active SUP will always be Gigabit Ethernet 0, while the Management Ethernet interface on the standby SUP will not be accessible using the Cisco IOS-XE CLI in the same telnet session. The standby SUP can be telnetted to through the console port, however.

The port can be accessed in configuration mode like any other port on the Cisco cBR Series Routers:

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface gigabitethernet0
Router(config-if)#
```

IP Address Handling in ROMMON and the Management Ethernet Port

Assuming the IOS-XE process has not begun running on the Cisco cBR Series Router, the IP address that was set in ROMMON acts as the IP address of the Management Ethernet interface. In cases where the IOS-XE process is running and has taken control of the Management Ethernet interface, the IP address specified when configuring the Gigabit Ethernet 0 interface in the IOS-XE CLI becomes the IP address of the Management Ethernet interface. The ROMMON-defined IP address is only used as the interface address when the IOS-XE process is inactive.

For this reason, the IP addresses specified in ROMMON and in the IOS-XE CLI can be identical and the Management Ethernet interface will function properly in single SUP configurations.

In dual SUP configurations, however, users should never configure the IP address in the ROMMON on either SUP0 or SUP1 to match each other or the IP address as defined by the IOS-XE CLI. Configuring matching IP addresses introduces the possibility for an active and standby Management Ethernet interface having the same IP address with different MAC addresses, which will lead to unpredictable traffic treatment.

Gigabit Ethernet Management Interface VRF

Placing the management Ethernet interface in its own VRF has the following effects on the Management Ethernet interface:

- Many features must be configured or used inside the VRF, so the CLI may be different for certain Management Ethernet functions on the Cisco cBR Series Routers than on Management Ethernet interfaces on other routers.
- The VRF prevents route leakage and avoids unnecessary traffic through the management port.

The Management Ethernet interface VRF supports both IPv4 and IPv6 address families.

Common Ethernet Management Tasks

Because users can perform most tasks on a router through the Management Ethernet interface, many tasks can be done by accessing the router through the Management Ethernet interface.

This section documents tasks that might be common or slightly tricky on the Cisco cBR Series Routers. It is not intended as a comprehensive list of all tasks that can be done using the Management Ethernet interface.

Viewing the VRF Configuration

The VRF configuration for the Management Ethernet interface is viewable using the **show running-config vrf** command.

This example shows the default VRF configuration:

```
Router# show running-config vrf
Building configuration...

Current configuration : 351 bytes
vrf definition Mgmt-intf
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
(some output removed for brevity)
```

Setting a Default Route in the Management Ethernet Interface VRF

To set a default route in the Management Ethernet Interface VRF, use the **ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0** *next-hop-IP-address* command.

Setting the Management Ethernet IP Address

The IP address of the Management Ethernet port is set like the IP address on any other interface.

Below are two simple examples of configuring an IPv4 address and an IPv6 address on the Management Ethernet interface.

IPv4 Example

```
Router(config)# interface GigabitEthernet 0
Router(config-if)# ip address A.B.C.D A.B.C.D
```

IPv6 Example

```
Router(config)# interface GigabitEthernet 0
Router(config-if)# ipv6 address X:X:X:X::X /prefix-length
```

Telnetting over the Management Ethernet Interface

Telnetting can be done through the VRF using the Management Ethernet interface.

In the following example, the router telnets to 172.17.1.1 through the Management Ethernet interface VRF:

```
Router# telnet 172.17.1.1 /vrf Mgmt-intf
```

Pinging over the Management Ethernet Interface

Pinging other interfaces using the Management Ethernet interface is done through the VRF.

In the following example, the router pings the interface with the IP address of 172.17.1.1 through the Management Ethernet interface:

```
Router# ping vrf Mgmt-intf 172.17.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.17.1.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

Copy Using TFTP or FTP

To copy a file using TFTP through the Management Ethernet interface, the **ip tftp source-interface GigabitEthernet 0** command must be entered before entering the **copy tftp** command because the **copy tftp** command has no option of specifying a VRF name.

Similarly, to copy a file using FTP through the Management Ethernet interface, the **ip ftp source-interface GigabitEthernet 0** command must be entered before entering the **copy ftp** command because the **copy ftp** command has no option of specifying a VRF name.

TFTP Example

```
Router(config)# ip tftp source-interface gigabitethernet 0
```

FTP Example

```
Router(config)# ip ftp source-interface gigabitethernet 0
```

NTP Server

To allow the software clock to be synchronized by a Network Time Protocol (NTP) time server over the Management Ethernet interface, enter the **ntp server vrf Mgmt-intf** command and specify the IP address of the device providing the update.

The following CLI provides an example of this procedure.

```
Router(config)# ntp server vrf Mgmt-intf 172.17.1.1
```

SYSLOG Server

To specify the Management Ethernet interface as the source IP or IPv6 address for logging purposes, enter the **logging host** *ip-address* **vrf Mgmt-intf** command.

The following CLI provides an example of this procedure.

```
Router(config)# logging host ip-address vrf Mgmt-intf
```

SNMP-Related Services

To specify the Management Ethernet interface as the source of all SNMP trap messages, enter the **snmp-server source-interface traps gigabitEthernet 0** command.

The following CLI provides an example of this procedure:

```
Router(config)# snmp-server source-interface traps gigabitEthernet 0
```

Domain Name Assignment

The IP domain name assignment for the Management Ethernet interface is done through the VRF.

To define the default domain name as the Management Ethernet VRF interface, enter the i**p domain-name vrf Mgmt-intf** *domain* command.

```
Router(config)# ip domain-name vrf Mgmt-intf cisco.com
```

DNS service

To specify the Management Ethernet interface VRF as a name server, enter the **ip name-server vrf Mgmt-intf** *IPv4/IPv6 address* command.

RADIUS or TACACS+ Server

To group the Management VRF as part of a AAA server group, enter the **ip vrf forward Mgmt-intf** command when configuring the AAA server group.

The same concept is true for configuring a TACACS+ server group. To group the Management VRF as part of a TACACS+ server group, enter the **ip vrf forwarding Mgmt-intf** command when configuring the TACACS+ server group.

RADIUS Server Group Configuration

```
Router(config)# aaa group server radius hello
Router(config-sg-radius)# ip vrf forwarding Mgmt-intf
```

TACACS+ Server Group Configuration

```
Router(config)# aaa group server tacacs+ hello
Router(config-sg-tacacs+)# ip vrf forwarding Mgmt-intf
```

VTY lines with ACL

To ensure an access control list (ACL) is attached to vty lines that are and are not using VRF, use the **vrf-also** option when attaching the ACL to the vty lines.

```
Router(config)# line vty 0 4
Router(config-line)#access-class 90 in vrf-also
```

Configuring the AUX Port for Network Management

Procedure

Step 1 AUX port is used for IOSd command prompt. Type the **set** command at the rommon prompt.

Step 2 Verify if BOOT_PARAM is defined. It must not be defined.

Step 3 If the BOOT_PARAM is defined, do the following:

a) Type **unset BOOT_PARAM**.

b) Type **sync**.

c) Type **reset**.

Step 4 Boot with the latest image. The AUX port will show IOS command prompt.

Preprovisioning the Supervisor in the Cisco cBR Chassis

Preprovisioning on the Cisco cBR allows you to configure the Supervisor s without their physical presence in the chassis.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**card** *slot*/**1** **sup-pic-8x10g** **Example:** `Router(config)# card 4/1 sup-pic-8x10g`	Preprovisions the Supervisor in the Cisco cBR chassis. • *slot*—Identifies the chassis slot number for the Supervisor PIC . The valid values are 4 and 5.

Configuring the Gigabit Ethernet Interface for Network Management

You must configure the GigabitEthernet0 interface and enable it to use the NME port.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface GigabitEthernet0** **Example:** `Router(config)# interface GigabitEthernet0`	Enters the Gigabit Ethernet interface configuration mode.
Step 4	**vrf forwarding** *vrf-name* **Example:** `Router(config-if)# vrf forwarding Mgmt-intf`	Associates a Virtual Routing and Forwarding (VRF) instance with the interface. • *vrf-name*—The interface name to be associated with the specified VRF.
Step 5	**ip address** *ip-address subnet-mask* **Example:** `Router(config-if)# ip address 192.71.0.1 255.255.255.0`	Sets the IP address of the Gigabit Ethernet interface. • *ip-address*—IP address of the Gigabit Ethernet interface. • *subnet -mask*—Subnet mask for the network.
Step 6	**no shutdown** **Example:**	Enables the Gigabit Ethernet interface.

	Command or Action	Purpose
	`Router(config-if)# no shutdown`	
Step 7	**speed 1000** [**negotiate**] **Example:** `Router(config-if)# speed 1000`	Configures the speed for the Gigabit Ethernet interface.
Step 8	**duplex full** **Example:** `Router(config-if)# duplex full`	Configures full duplex operation on the Gigabit Ethernet interface.
Step 9	**negotiation auto** **Example:** `Router(config-if)# negotiation auto`	Selects the auto-negotiation mode.
Step 10	**end** **Example:** `Router(config-if)# end`	Exits Gigabit Ethernet interface configuration mode. Returns to privileged EXEC mode.

Configuring the DTI Port on the Supervisor PIC

The Cisco cBR router can run in standalone mode, which uses internal clock and does not require any external reference clock source. The Cisco cBR router also supports DTI server as an external clocking source. To use a DTI server as a reference clock source, you must enable the DTI port on the Supervisor PIC .

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable clock dti** **Example:** `Router(config)# cable clock dti`	Configures the DTI clock reference mode for the Supervisor PIC .

Configuring the TenGigabit Ethernet Interface for Network Management

You must configure the TenGigabitEthernet interface and enable it to use the NME port.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface TenGigabitEthernet** **Example:** `Router(config)# interface TenGigabitEthernet4/1/0`	Enters the TenGigabit Ethernet interface configuration mode.
Step 4	**ip address** *ip-address subnet-mask* **Example:** `Router(config-if)# ip address 1.2.3.4 255.255.255.0`	Sets the IP address of the TenGigabit Ethernet interface.
Step 5	**load-interval** *seconds* **Example:** `Router(config-if)# load-interval 30`	Changes the length of time for which data is used to compute load statistics.
Step 6	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the TenGigabit Ethernet interface.
Step 7	**end** **Example:** `Router(config-if)# end`	Exits TenGigabit Ethernet interface configuration mode. Returns to privileged EXEC mode.

Connecting the New Router to the Network

Connect the new router to the network using a n Ethernet interface. After the router successfully resolves its host name, new router sends a TFTP broadcast requesting the file name-confg or name.cfg. The router name must be in all lowercase, even if the true host name is not. The file is downloaded to the new router, where

the configuration commands take effect immediately. If the configuration file is complete, the new router should be fully operational.

To save the complete configuration to NVRAM, use the following commands in privileged EXEC mode:

Procedure

	Command or Action	Purpose
Step 1	**enable password**	Enters privileged mode on the new router.
Step 2	**copy running-config startup-config**	Saves the information from the name-config file into your startup configuration. On most platforms, this step saves the configuration to NVRAM. **Note** Verify that the existing and new routers (or access servers) are connected before entering the **copy running-config startup-config** EXEC command to save configuration changes. Use the **ping** EXEC command to verify connectivity. If an incorrect configuration file is downloaded, the new router will load NVRAM configuration information before it can enter AutoInstall mode. If the configuration file is a minimal configuration file, the new router comes up, but with only one interface operational. Use the following commands to connect to the new router and configure it.
Step 3	**telnet existing**	Establishes a Telnet connection to the existing router.
Step 4	**telnet newrouter**	From the existing router, establishes a Telnet connection to the new router.
Step 5	**enable password**	Enters privileged EXEC mode.
Step 6	**setup**	Enters setup mode to configure the new router.

Setting Password Protection on the Cisco CMTS

Note For security purposes, the EXEC has two levels of access to commands: user EXEC mode and privileged EXEC mode. The commands available at the user level are a subset of those available at the privileged level.

Tip Because many privileged-level EXEC commands are used to set operating parameters, password-protect these commands to prevent unauthorized use.

Note An enable secret password can contain from 1 to 25 uppercase and lowercase alphanumeric characters. An enable password can contain any number of uppercase and lowercase alphanumeric characters. A number cannot be the first character. Spaces are valid password characters; for example, “two words” is a valid password. Leading spaces are ignored. Trailing spaces are recognized. Alphanumeric characters are recognized as uppercase or lowercase.

Passwords should be different for maximum security. If you enter the same password for both during the setup script, the system accepts it, but you receive a warning message indicating that you should enter a different password.

At the EXEC prompt, enter one of the following two commands to set password protection:

- **enable secret password**—a very secure encrypted password.
- **enable**—is a less secure and nonencrypted password.

To gain access to privileged-level commands, enter the desired password.

Recovering Lost Password on the Cisco CMTS

Complete the following steps to recover or replace a lost enable, enable secret, or console login password:

Procedure

Step 1 Attach an ASCII terminal to the console port on your Cisco CMTS.

Step 2 Configure the terminal to operate at 9600 baud, 8 data bits, no parity, and 1 stop bits.

Step 3 If you can log in to the router as a nonprivileged user, enter the **show version** command to display the existing configuration register value. Note the value for later use. If you cannot log in to the router at all, continue with the next step.

Step 4 Press the **Break** key or send a **Break** from the console terminal.

- If Break is enabled, the router enters the ROM monitor, indicated by the ROM monitor prompt (rommon n>), where n is the number of the command line. Proceed to configuring the register.
- If Break is disabled, power cycle the router (turn the router off or unplug the power cord, and then restore power). Within 60 seconds of restoring the power to the router, press the **Break** key or send a **Break**. This action causes the router to enter the ROM monitor and display the ROM monitor prompt (rommon 1>).

Step 5 To set the configuration register on a Cisco CMTS, use the configuration register utility by entering the **confreg** command at the ROM monitor prompt as follows:

```
rommon 1> confreg
```

Answer **yes** to the *enable ignore system config info?* prompt and note the current configuration register settings.

Step 6 Initialize the router by entering the **reset** command as follows:

```
rommon 2> reset
```

The router initializes, the configuration register is set to 0x142, the router boots the system image from Flash memory and enters the System Configuration dialog (setup), as follows:

```
--- System Configuration Dialog --
```

Step 7 Enter **no** in response to the System Configuration dialog prompts until the following message appears:

```
Press RETURN to get started!
```

Step 8 Press **Return**. The user EXEC prompt appears as follows:

```
Router>
```

Step 9 Enter the **enable** command to enter privileged EXEC mode.

Step 10 Enter the **show startup-config** command to display the passwords in the configuration file as follows:

```
Router# show startup-config
```

Step 11 Scan the configuration file display looking for the passwords; the enable passwords are usually near the beginning of the file, and the console login or user EXEC password is near the end. The passwords displayed will look something like this:

```
enable secret 5 $1$ORPP$s9syZt4uKn3SnpuLDrhuei
enable password 23skiddoo
.
.
line con 0
 password onramp
```

Note The enable secret password is encrypted and cannot be recovered; it must be replaced. The enable and console passwords can be encrypted text or clear text.

Proceed to the next step to replace an enable secret, console login, or enable password. If there is no enable secret password, note the enable and console login passwords if they are not encrypted and proceed to set the configuration register to the original value.

Caution Do not perform the next step unless you have determined that you must change or replace the enable, enable secret, or console login passwords. Failure to follow the steps as presented here could cause your router configuration to be erased.

Step 12 (Optional) Enter the configure memory command to load the startup configuration file into running memory. This action allows you to modify or replace passwords in the configuration.

```
Router# configure memory
```

Step 13 Enter the **configure terminal** command for configuration mode:

```
Router# configure terminal
```

Step 14 To change all three passwords, enter the following commands:

```
Router(config)# enable secret newpassword1

Router(config)# enable password newpassword2
Router(config)# line con 0

Router(config)# password newpassword3
```

Change only the passwords necessary for your configuration. You can remove individual passwords by using the **no** form of the previous commands. For example, entering the **no enable secret** command removes the enable secret password.

Step 15 You must configure all interfaces to not be administratively shut down as follows:

```
Router(config)# interface gigabitethernet 0

Router(config)# no shutdown
```

Enter the equivalent commands for all interfaces that were originally configured. If you omit this step, all interfaces are administratively shut down and unavailable when the router is restarted.

Step 16 Use the **config-register** command to set the configuration register to the original value noted earlier.

Step 17 Press **Ctrl-Z** or type **end** to exit configuration mode:

```
Router(config)# end
```

Caution Do not perform the next step unless you have changed or replaced a password. If you skipped changing or replacing the enable, enable secret, or console login passwords previously, then proceed now to reload. Failure to observe this sequence causes the system to erase your router configuration file.

Step 18 Enter the **copy running-config startup-config** command to save the new configuration to nonvolatile memory:

```
Router# copy running-config startup-config
```

Step 19 Enter the **reload** command to reboot the router:

```
Router# reload
```

Step 20 Log in to the router with the new or recovered passwords.

Saving Your Configuration Settings

To store the configuration or changes to your startup configuration in NVRAM, enter the **copy running-config startup-config** command at the *Router#* prompt.

This command saves the configuration settings you set using configuration mode, the Setup facility, or AutoInstall.

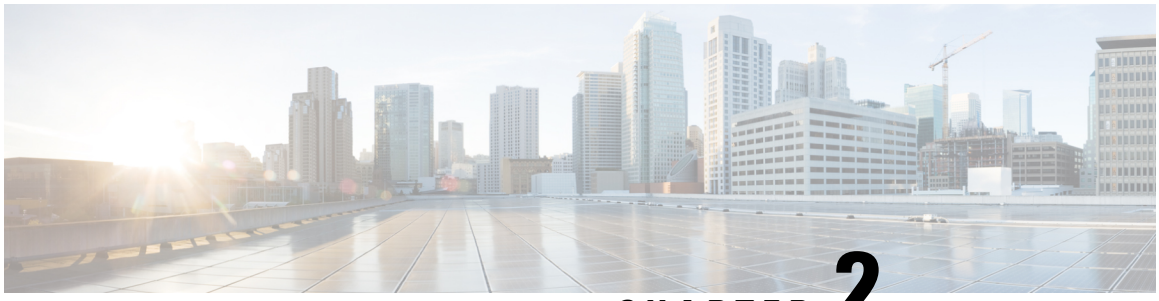
Note If you do not save your settings, your configuration will be lost the next time you reload the router.

Example

```
Router# copy running-config startup-config
```

Reviewing Your Settings and Configurations

- To view the current configuration of a Cisco CMTS, run the **show running-config** command at the command-line interface (CLI) prompt in EXEC mode or privileged EXEC mode.
- To review changes you make to the configuration, use the EXEC **show startup-config** command to display the information stored in NVRAM.



CHAPTER 2

Cisco Smart Licensing

A new licensing model, based on a single technology, has been designed for Cisco called Smart Licensing that is intended to provide Enterprise Level Agreement-like capabilities for all Cisco products. The Cisco Smart Licensing is based on the Trust but Verify model.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 31
- Prerequisites for Cisco Smart Licensing, on page 32
- Information About Cisco Smart Licensing, on page 33
- How to Configure Cisco Smart Licensing, on page 34
- How to Configure Cisco Smart Licensing using Transport Gateway Solution, on page 44
- Verifying Cisco Smart Licensing Configuration, on page 46
- Troubleshooting Cisco Smart Licensing, on page 52
- Additional References, on page 53
- Feature Information for Cisco Smart Licensing, on page 53

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 1: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Cisco Smart Licensing

- You must configure the DNS server using the **ip name-server** global configuration command.
- You must configure the IP DNS-based hostname-to-address translation using the **ip domain-lookup** global configuration command.
- Cisco Smart Licensing is enabled by default on the Cisco cBR router. However, you must ensure that the CiscoTAC-1 call-home profile points to the Cisco Smart Software Manager at the following URL using the **show call-home profile CiscoTAC-1** command:

  https://tools.cisco.com/its/service/oddce/services/DDCEService

  The following is a sample output of the **show call-home profile CiscoTAC-1** command:

```
Router# show call-home profile CiscoTAC-1

Load for five secs: 10%/1%; one minute: 9%; five minutes: 8%
Time source is NTP, 16:49:35.525 PDT Thu Oct 29 2015

Profile Name: CiscoTAC-1
    Profile status: ACTIVE
    Profile mode: Anonymous Reporting Only
    Reporting Data: Smart Call Home, Smart Licensing
    Preferred Message Format: xml
    Message Size Limit: 3145728 Bytes
    Transport Method: http
```

```
Email address(es): callhome@cisco.com
HTTP  address(es): https://tools.cisco.com/its/service/oddce/services/DDCEService

Periodic configuration info message is scheduled every 19 day of the month at 11:41


Periodic inventory info message is scheduled every 19 day of the month at 11:26

Alert-group               Severity
------------------------  ------------
crash                     debug
diagnostic                minor
environment               minor
inventory                 normal

Syslog-Pattern            Severity
------------------------  ------------
.*                        major
```

- Ensure that you can ping the DNS server. If you are unable to ping the server, verify the connectivity to the NME port on the Cisco cBR router.

Note If you are using a Virtual Routing and Forwarding (VRF) instance, ensure that you can ping the VRF instance.

Information About Cisco Smart Licensing

Cisco Smart Licensing is software-based licensing that consists of tools and processes to authorize the customers for the usage and reporting of the Cisco products. The feature has the capability to capture the customer order and communicate with the Cisco Cloud License Service through Smart Call Home transport media to complete the product registration and authorization. If the Cisco products stop communicating with the Cisco Cloud License Service for 90 days, the cable interfaces in the Cisco products will be locked, which means the customer can no longer enable/disable the cable interfaces.

The Cisco Smart Licensing feature is aimed at giving users an experience of a single, standardized licensing solution for all Cisco products.

In the Cisco Smart Licensing Model, you can activate licensed features (also known as entitlements) without the use of a special software key or upgrade license file. You can activate the new functionality using the appropriate product commands and configurations and the functionality is activated. A software reboot is not required for the Cisco cBR router.

The Cisco cBR router supports software activation using Cisco Smart Licensing. The Cisco Smart Licensing is enabled by default on the Cisco cBR router.

Note A LCHA license is needed for each working linecard that is protected by the protect linecard.

Downstream License

The DOCSIS 3.1 license scheme provides support to identify the DOCSIS 3.1 channels and their width. The DOCSIS 3.1 entitlement is DOCSIS 3.1 Downstream Channel License.

Out of Compliance Enforcement

The following two events are responsible for triggering a DOCSIS configuration lock enforcement.

Eval-Expired (Evaluation Period Expired)

When a router is not registered with Smart License Manager for more than ninety days.

Auth-Expired (authorization Period Expired)

When a registered router fails to communicate with the Smart License Manager for more than ninety days.

When either of the above mentioned events occur, the Smart Agent sends a notification to the platform. The platform, upon receiving such notification, **locks** the following CLI commands:

- **[no] cable upstream shutdown** *upstream-port-number*
- **contoller upstream-cable** *slot/subslot/controller-port-number*
- **rf-chan***channel-number*

Any attempt to configure the above mentioned CLIs in this condition would fail and a warning message will be displayed. Under this condition all other CLIs are configurable, some of which may be required to configure Cisco License Call Home, connect to cisco and register the device to come out of either of the above two events and enter into authorized or Out of Compliance (OOC) state.

You can copy the modified configuration file to the startup configuration file and reload the device to make that configuration effective. However, when the device is in enforced state, you can only copy the running configuration file to the startup configuration file.

Note Any attempt to copy any other file fails and a warning message is displayed.

How to Configure Cisco Smart Licensing

This section contains the following:

Using Cisco Smart Licensing Agent on the Router

Procedure

Step 1 Set up a Cisco Smart Account. See Setting Up a Cisco Smart Account, on page 35.

Step 2 Log in to the Cisco Smart Software Manager.

Step 3 (Optional) Create a virtual account. See Creating Virtual Accounts, on page 41.

Note A single default virtual account is always available.

Step 4 Create a product instance registration token. See Creating a Product Instance Registration Token, on page 42.

Step 5 Register the router with the Cisco Licensing Cloud using the product instance registration token. See Registering the Router with the Cisco Licensing Cloud Using the Registration Token, on page 44.

Step 6 Log in to the Cisco Smart Software Manager for managing licenses.

For more information, see the *Cisco Smart Software Manager User Guide*, which is accessible from the Cisco Smart Software Manager tool.

Setting Up a Cisco Smart Account

Cisco Smart Account enables you to fully utilize the license management features of the smart-enabled products.

Before you begin

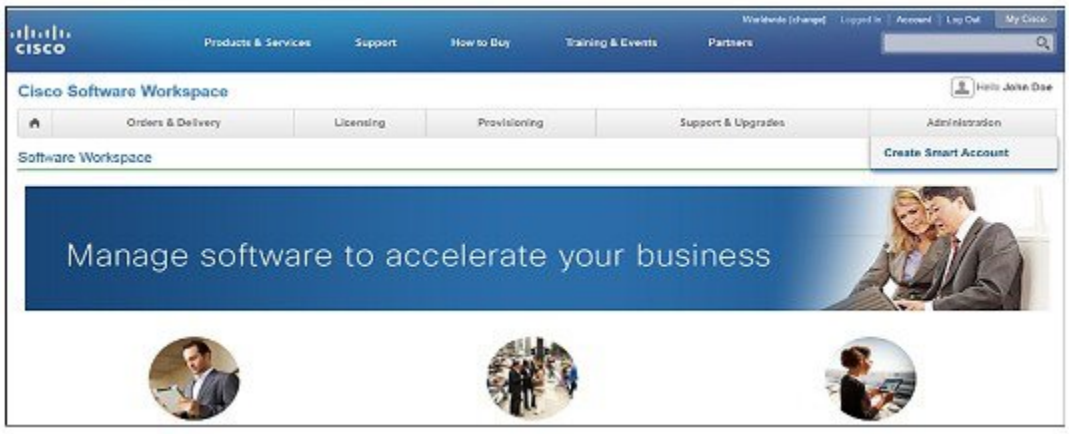
- Ensure that you have a CCO ID.

Procedure

Step 1 Log in to Cisco Software Workspace (CSW) with your CCO ID.

Step 2 Hover the cursor over the *Administration* tab and click **Create Smart Accounts**.

Figure 1: Creating Smart Account

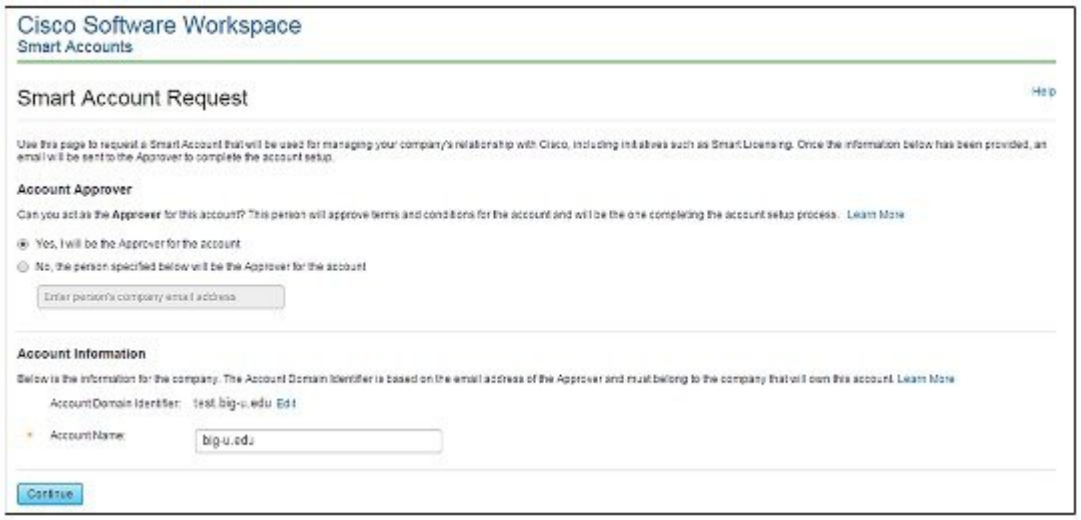


Step 3 Perform one of the following to select the Account Approver:

- To select yourself as the Approver, click the **Yes, I will be the Approver for the account** radio button.
- To select other person as the Approver, click the **No, the person specified below will be the Approver for the account** radio button and specify the person's e-mail ID.

Note The specified Approver must have the authority to enter legal agreements. The Approver serves as the primary owner and nominates account administrators.

Figure 2: Selecting the Approver



Step 4 If you are the Approver, perform the following:

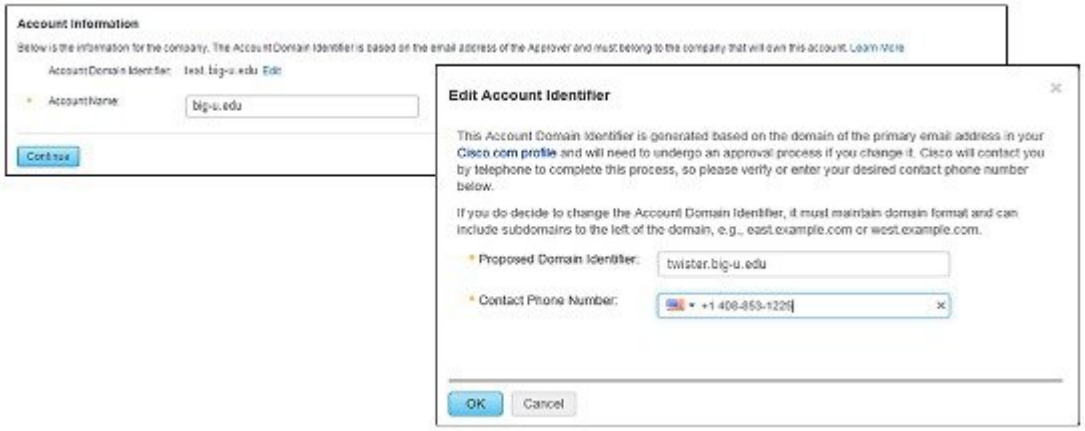
a) Enter the Account Name, Company/Organization Name, Country, and State/Province/Region information.

b) (Optional) Click **Edit**. In the *Edit Account Identifier* window, enter a valid Proposed Domain Identifier and Contact Phone Number. Click **OK**.

Note The default domain identifier is the Approver e-mail domain. If you edit the domain identifier, the change goes through a manual approval process.

c) Click **Continue** to select the legal address to be linked to your Cisco Smart Account.

Figure 3: Setting Up Account Information When You Are The Approver



Step 5 If you are not the Approver, perform the following:

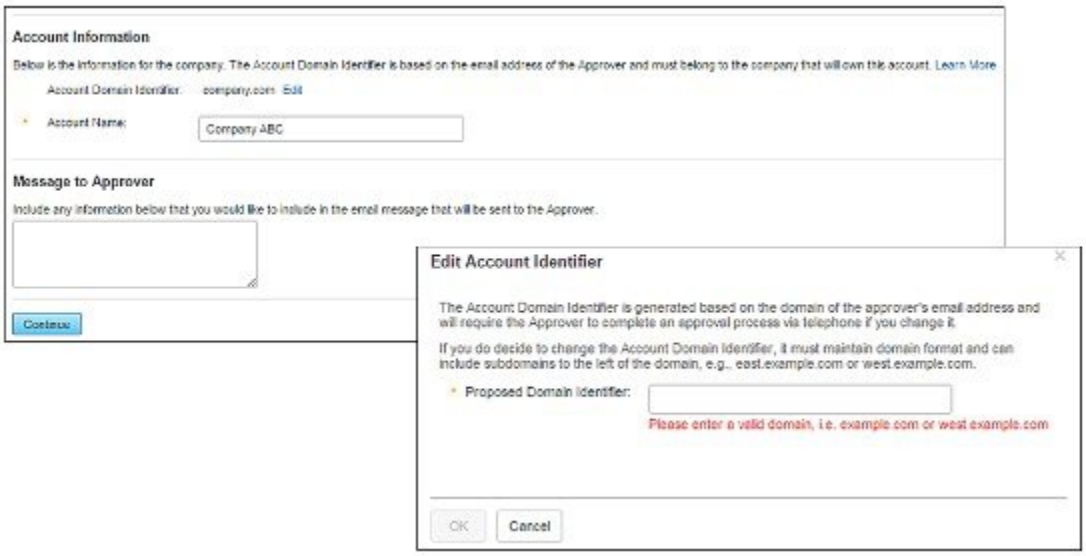
a) Enter the Account Name and an optional Message to Approver.

b) (Optional) Click **Edit**. In the *Edit Account Identifier* window, enter a valid Proposed Domain Identifier. Click **OK**.

Note The default domain identifier is the Approver e-mail domain. If you edit the domain identifier, the change goes through a manual approval process.

c) Click **Continue**.

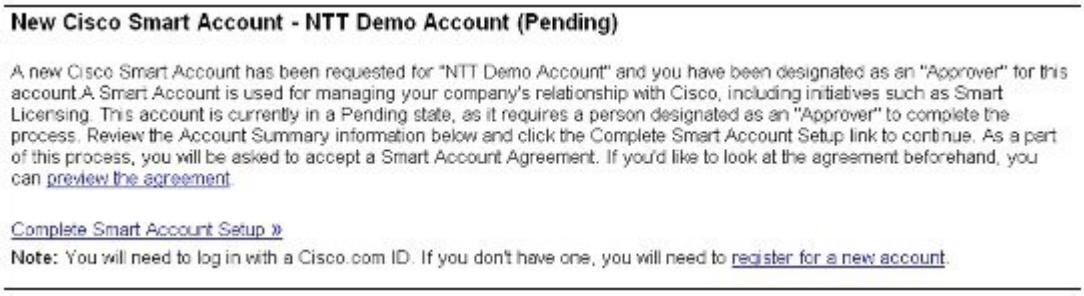
Figure 4: Setting Up Account Information When You Are Not The Approver



Step 6 If you are not the Approver, the Approver will receive an e-mail and must perform the following:

a) Click **Complete Smart Account Setup** in the received e-mail.

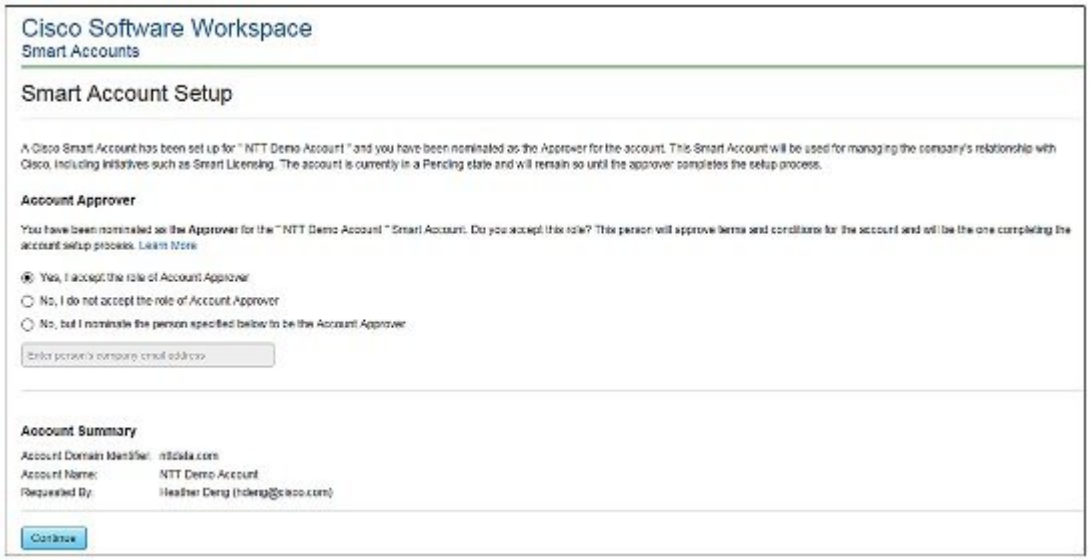
Figure 5: Complete Smart Account Setup Link in E-mail



b) Click the appropriate radio button to accept, decline, or nominate another Approver. To nominate another Approver, enter the person's e-mail address. Click **Continue**.

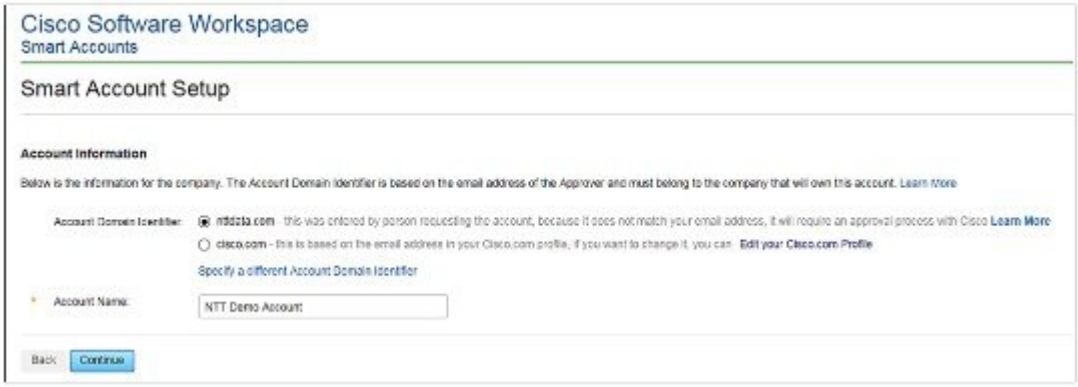
Note If the Approver declines, the Cisco Smart Account is deleted. If the Approver nominates another approver, the new Approver must accept the role.

Figure 6: Accepting the Account Approver Role



c) After accepting the Approver role, click the appropriate radio button to select the Account Domain Identifier or specify a different Account Domain Identifier.

Figure 7: Completing the Account Information



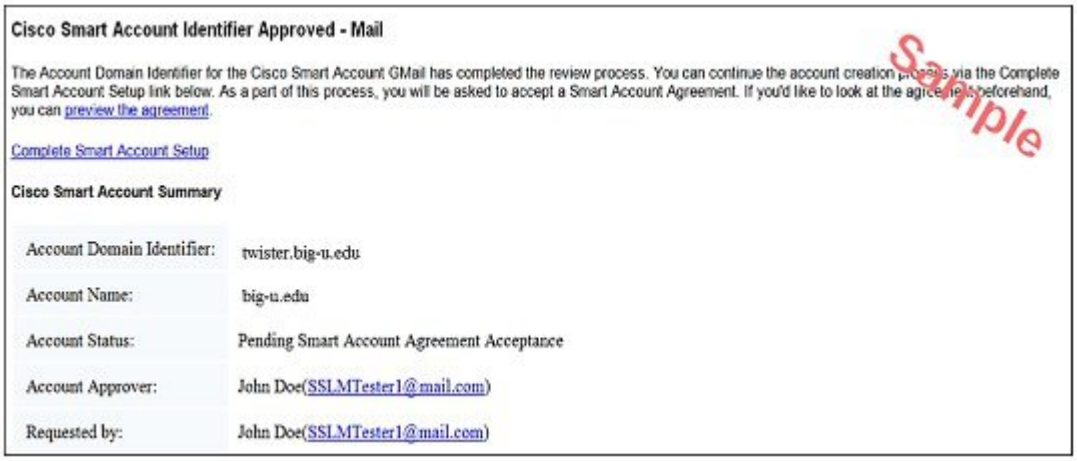
d) Enter the Account Name and click **Continue**.

The Approver role is accepted and Cisco Smart Account is pending Account Domain approval.

Step 7 After the Account Domain is approved, the Approver will receive an e-mail and must perform the following:

a) Click **Complete Smart Account Setup** in the received e-mail.

Figure 8: Cisco Smart Account Identifier Approved E-mail


Cisco Smart Account Identifier Approved - Mail

The Account Domain Identifier for the Cisco Smart Account GMail has completed the review process. You can continue the account creation process via the Complete Smart Account Setup link below. As a part of this process, you will be asked to accept a Smart Account Agreement. If you'd like to look at the agreement beforehand, you can preview the agreement.

Complete Smart Account Setup

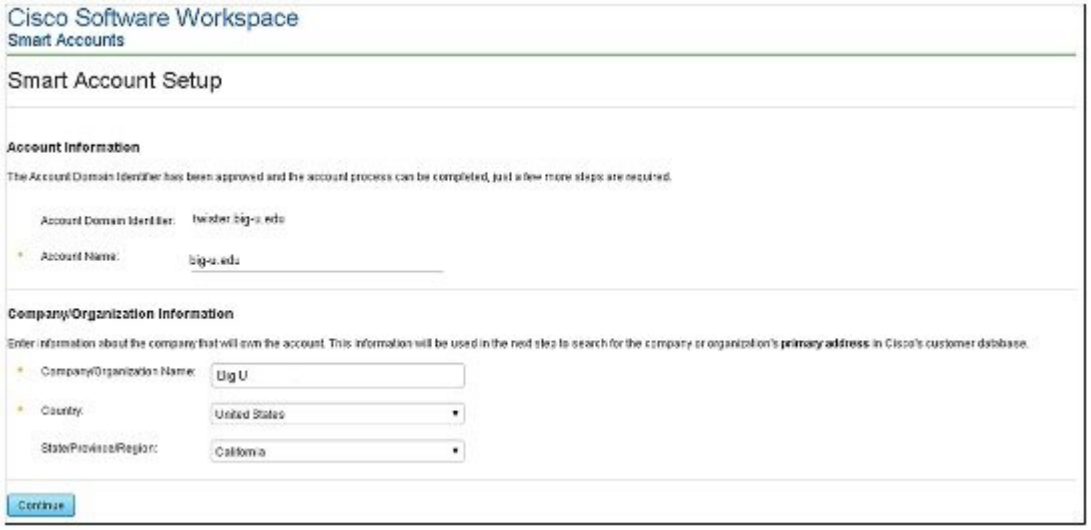
Cisco Smart Account Summary

Account Domain Identifier:	twister.big-u.edu
Account Name:	big-u.edu
Account Status:	Pending Smart Account Agreement Acceptance
Account Approver:	John Doe(SSLMTester1@mail.com)
Requested by:	John Doe(SSLMTester1@mail.com)

Sample

b) Enter the Account Name, Company/Organization Name, Country, and State/Province/Region information.

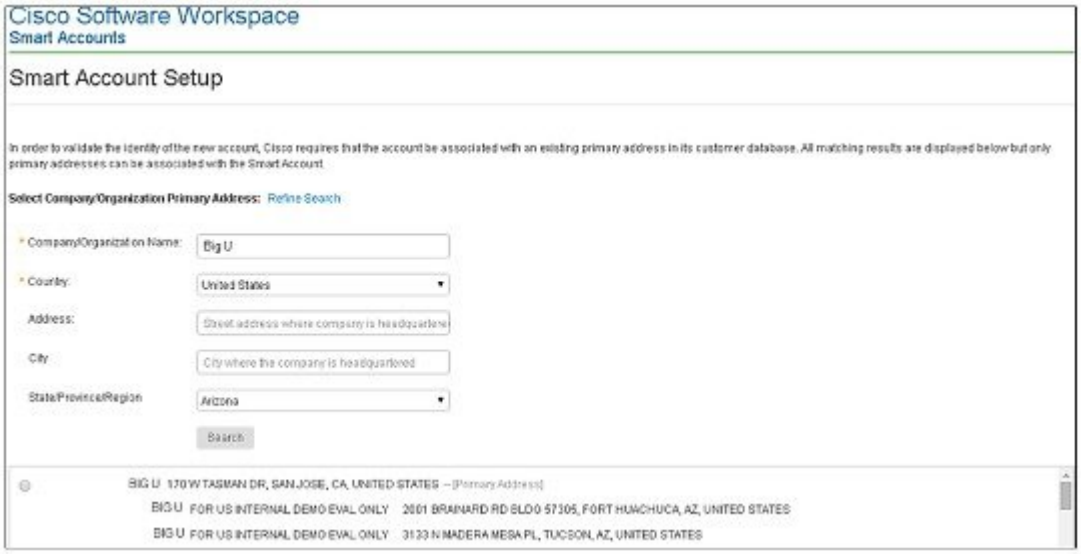
Figure 9: Completing the Account Information and Company/Organization Information



c) Click **Continue** to select the legal address to be linked to the Cisco Smart Account.

d) Select the Company/Organization Primary Address using the Refine Search option and click **Continue**.

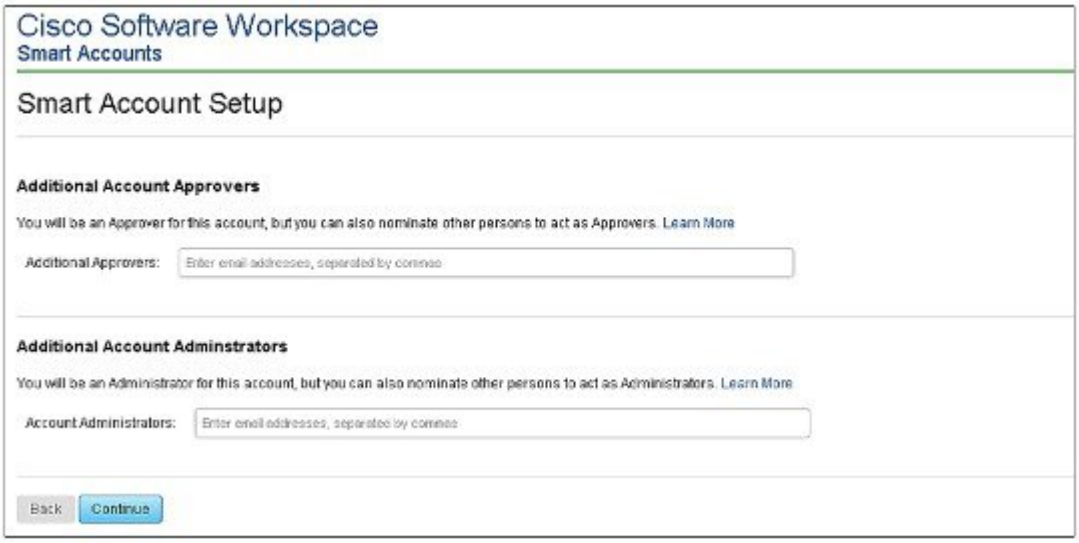
Figure 10: Selecting the Company/Organization Primary Address



e) (Optional) Enter the e-mail addresses of the Additional Account Approvers and Additional Account Administrators.

The initial Approver automatically becomes an Administrator. Additional Administrators can be created or assigned separately from the Approver creation process.

Figure 11: Nominating Additional Account Approvers and Administrators



f) Click **Continue**.
g) Review the agreement and check the **I agree to the terms above** check box to accept.
h) Click **Accept and Create Account** to create the Cisco Smart Account.

Figure 12: Accepting the Agreement and Creating the Cisco Smart Account

You will receive an e-mail confirming the creation of the Cisco Smart Account.

Creating Virtual Accounts

This procedure is optional. Virtual accounts are collections of licenses and product instances. You can create virtual accounts in Cisco Smart Software Manager to organize the licenses for your company into logical entities. A single virtual account is available by default.

Before you begin

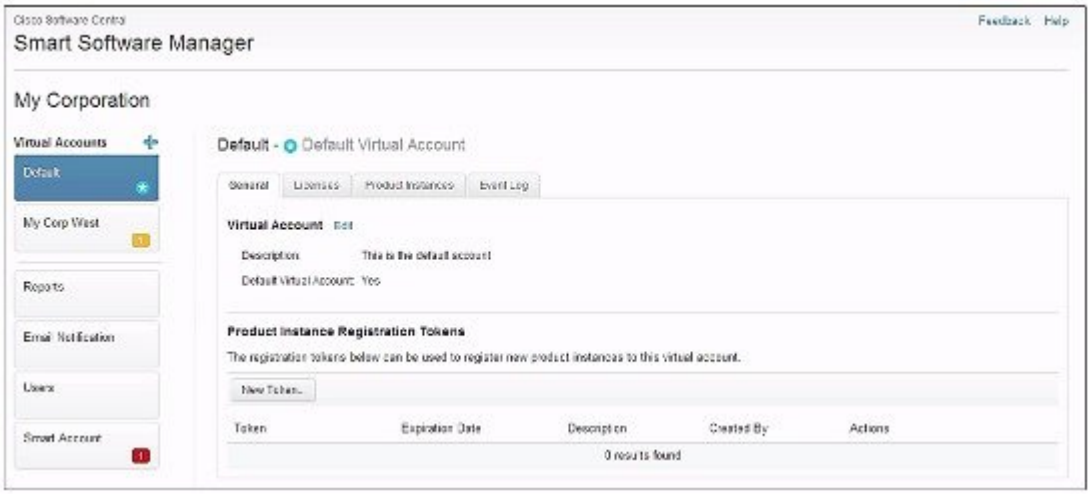
Set up a Cisco Smart Account. See Setting Up a Cisco Smart Account, on page 35.

Procedure

Step 1 Log in to the Cisco Smart Software Manager.

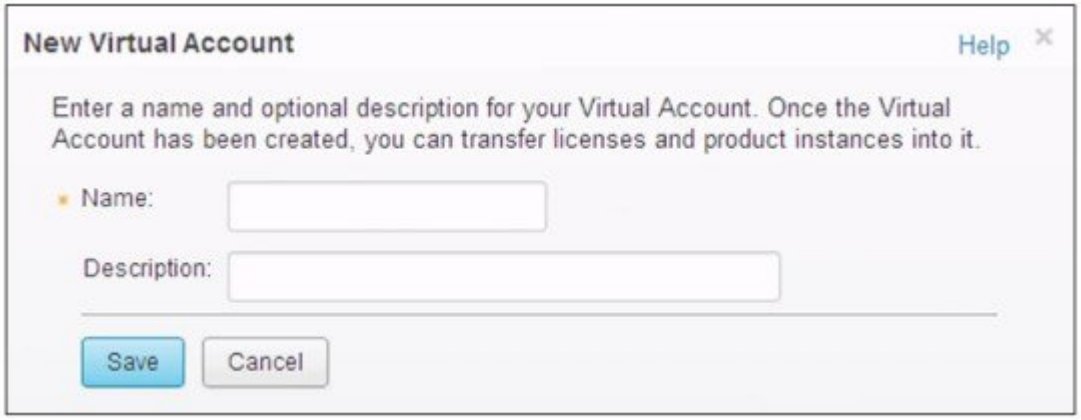
Step 2 Click the + (plus) symbol to create a virtual account.

Figure 13: Creating a Virtual Account



Step 3 In the **New Virtual Account** dialog box, enter the Name and Description.

Figure 14: New Virtual Account Dialog Box



Step 4 Click **Save**.

Creating a Product Instance Registration Token

Product instance registration tokens are used to register and consume a product for Cisco Smart Licensing. You must generate a token to register the product and add the product instance to a specified virtual account. Registration tokens can be valid from 1 to 365 days.

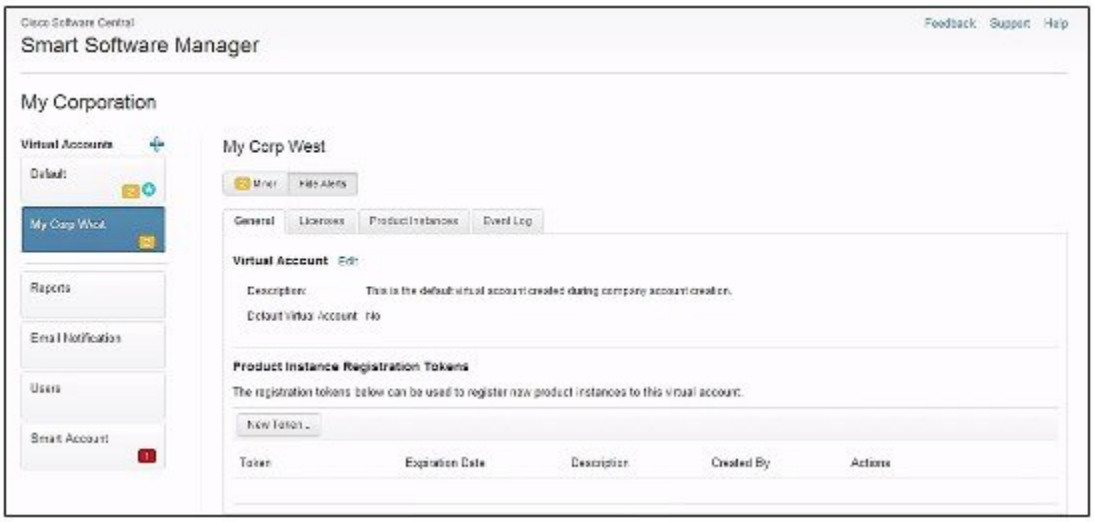
Procedure

Step 1 Log in to the Cisco Smart Software Manager.

Step 2 Click an existing virtual account.

Step 3 In the **General** tab, click **New Token**.

Figure 15: Creating a New Registration Token



Step 4 In the **Create Registration Token** dialog box, enter the Description and Expire After information and click **Create Token**.

Figure 16: Create Registration Token Dialog Box

What to do next

Register the router with the Cisco Licensing Cloud. For more details, see the Registering the Router with the Cisco Licensing Cloud Using the Registration Token, on page 44 section.

Registering the Router with the Cisco Licensing Cloud Using the Registration Token

The router registration is performed only once for each product instance.

Note Ensure that you have the product instance registration token.

To register the router with the Cisco Licensing Cloud using a registration token, use the following commands:

```
enable
license smart register idtoken id-token
```

For example:

```
Router#license smart register idtoken
YjBkOWM5YTItMDFiOS00ZjBmLTllY2YtODEzMzg1YTMyZDVhLTEz
ODE0MjE0%0ANzc5NDF8U1BDUTAySWFRTmJqa1NnbmlzRUIyaGlYU
053L0pHZTNvUW9VTFpE%0AekxCOD0%3D%0A
```

The system contacts the Cisco Smart Licensing servers to obtain authorization for Smart Licensing.

The license agent registers the product with Cisco and receives an identity certificate. This certificate is saved and automatically used for all future communications with Cisco. The license agent automatically renews the registration information with Cisco every 30 days.

Note Smart licensing may fail if IPv6 is configured on any interface, and the router does not have IPv6 connectivity to the Internet or Cisco Smart Software Agent (at tools.cisco.com). Log file error messages similar to the following may appear.

(These messages may also appear as a result of other conditions being true.)

```
%SMART_LIC-3-AGENT_REG_FAILED: Smart Agent for Licensing Registration with Cisco licensing
 cloud failed: Fail to send out Call Home HTTP message.
%SMART_LIC-3-COMM_FAILED: Communications failure with Cisco licensing cloud: Fail to send
out Call Home HTTP message.
```

If connectivity fails due to this issue, see the Re-establishing Connectivity to Cisco Smart Call Home Server section.

After connectivity is established, register the router with the Cisco Licensing Cloud.

How to Configure Cisco Smart Licensing using Transport Gateway Solution

The steps below describe how to configure Cisco smart licensing using transport gateway solution.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** **Example:** `Router(config)# crypto pki trustpoint cisco`	Declare the trustpoint that the router should use.
Step 4	**enrollment terminal** **Example:** `Router(ca-trustpoint)# enrollment terminal`	Specify manual cut-and-paste certificate enrollment.
Step 5	**revocation-check** *method* **Example:** `Router(ca-trustpoint)# revocation-check none`	Check the revocation status of a certificate. Method **none** means certificate checking is not required.
Step 6	**crypto pki authenticate** **Example:** `Router(config)# crypto pki authenticate cisco`	Authenticate the certification authority.

Step 7	**no reporting smart-licensing-data**
	Example:
	`Router(config)# call-home`
	`Router(cfg-call-home)# profile CiscoTAC-1`
	`Router(cfg-call-home-profile)# no reporting smart-licensing-data`
		Configure the default profile to not to communicate with tools.cisco.com.

Step 8	**destination address http** *address*
	Example:
	`Router(config)# call-home`
	`Router(cfg-call-home)# profile Custom-Profile-1`
	`Router(cfg-call-home-profile)# reporting smart-licensing-data`
	`Router(cfg-call-home-profile)# destination transport-method http`
	`Router(cfg-call-home-profile)# no`
		Configure the custom profile to communicate with the transport server, here we use Custom Profile 1 as the name of the custom profile.

	Command or Action	Purpose
	destination transport-method email Router(cfg-call-home-profile)# **destination address http** **https://TDS_IP_HERE:8443/Transportgateway/services/DeviceRequestHandler**	

Verifying Cisco Smart Licensing Configuration

Use the following commands to verify the Cisco Smart Licensing Configuration on the Cisco cBR router:

- **show license all**—Displays all the license information.

  The following is a sample output of this command:

```
Router# show license all

Smart Licensing Status
======================

Smart Licensing is ENABLED

Registration:
  Status: REGISTERED
  Virtual Account: auto-test-1
  Initial Registration: SUCCEEDED on Mar  5 02:01:03 2015 UTC
  Last Renewal Attempt: None
  Next Renewal Attempt: Sep  1 02:03:51 2015 UTC
  Registration Expires: Never

License Authorization:
  Status: OUT OF COMPLIANCE on Mar  5 03:34:54 2015 UTC
  Last Communication Attempt: SUCCEEDED on Mar  5 03:35:57 2015 UTC
  Next Communication Attempt: Mar  5 15:35:57 2015 UTC
  Communication Deadline: Jun  3 03:32:51 2015 UTC

License Usage
==============

(US_License):
  Description:
  Count: 64
  Version: 1.0
  Status: AUTHORIZED

(DS_License):
  Description:
  Count: 768
  Version: 1.0
  Status: AUTHORIZED

(WAN_License):
  Description:
  Count: 8
  Version: 1.0
  Status: OUT OF COMPLIANCE

Product Information
===================
UDI: PID:CBR-8-CCAP-CHASS,SN:FXS1739Q0NT
```

```
HA UDI List:
    Active:PID:CBR-8-CCAP-CHASS,SN:FXS1739Q0NT
    Standby:PID:CBR-8-CCAP-CHASS,SN:FXS1739Q0NT

Agent Version
=============
Smart Agent for Licensing: 1.2.1_throttle/5
Component Versions: SA:(1_2_1_throttle)1.1.0, SI:(rel20)1.0.1, CH:(rel4)1.0.15,
PK:(rel16)1.0.7
```

- **show license status**—Displays the license status information.

  The following is a sample output of this command:

```
Router# show license status

Smart Licensing is ENABLED

Registration:
  Status: REGISTERED
  Virtual Account: auto-test-1
  Initial Registration: SUCCEEDED on Mar  5 02:01:03 2015 UTC
  Last Renewal Attempt: None
  Next Renewal Attempt: Sep  1 02:03:51 2015 UTC
  Registration Expires: Never

License Authorization:
  Status: OUT OF COMPLIANCE on Mar  5 03:34:54 2015 UTC
  Last Communication Attempt: SUCCEEDED on Mar  5 03:35:57 2015 UTC
  Next Communication Attempt: Mar  5 15:35:56 2015 UTC
  Communication Deadline: Jun  3 03:32:50 2015 UTC
```

- **show license summary**—Displays the license summary information.

  The following is a sample output of this command:

```
Router# show license summary

Smart Licensing is ENABLED

Registration:
  Status: REGISTERED
  Virtual Account: auto-test-1
  Last Renewal Attempt: None
  Next Renewal Attempt: Sep  1 02:03:51 2015 UTC

License Authorization:
  Status: OUT OF COMPLIANCE
  Last Communication Attempt: SUCCEEDED
  Next Communication Attempt: Mar  5 15:35:56 2015 UTC

License Usage:
               License       Entitlement tag    Count Status
  -----------------------------------------------------------------
                               (US_License) 64       AUTHORIZED
                               (DS_License) 768      AUTHORIZED
                              (WAN_License) 8        OUT OF COMPLIANCE
```

- **show license tech support**—Displays the license technical support information.

  The following is a sample output of this command:

```
Router# show license tech support

Smart Licensing Tech Support info

Smart Licensing Status
======================

Smart Licensing is ENABLED

Registration:
  Status: REGISTERED
  Virtual Account: auto-test-1
  Initial Registration: SUCCEEDED on Mar  5 02:01:03 2015 UTC
  Last Renewal Attempt: None
  Next Renewal Attempt: Sep  1 02:03:51 2015 UTC
  Registration Expires: Never

License Authorization:
  Status: OUT OF COMPLIANCE on Mar  5 03:34:54 2015 UTC
  Last Communication Attempt: SUCCEEDED on Mar  5 03:35:57 2015 UTC
  Next Communication Attempt: Mar  5 15:35:57 2015 UTC
  Communication Deadline: Jun  3 03:32:51 2015 UTC

Evaluation Period:
  Evaluation Mode: Not In Use
  Evaluation Period Remaining: 89 days, 23 hours, 25 minutes, 40 seconds

License Usage
=============
Handle: 1
    License: 'nullPtr'
    Entitlement Tag:
regid.2014-11.com.cisco.US_License,1.0_a3f32909-2c71-426c-b3e0-eeefc946f9b3
    Description: <empty>
    Count: 64
    Version: 1.0
    Status: AUTHORIZED(3)
    Status time: Mar  5 03:34:54 2015 UTC
    Request Time: Mar  5 03:34:17 2015 UTC

Handle: 2
    License: 'nullPtr'
    Entitlement Tag:
regid.2014-11.com.cisco.DS_License,1.0_71ad0ae1-5e5e-4f02-b380-d2e1b8dcfa03
    Description: <empty>
    Count: 768
    Version: 1.0
    Status: AUTHORIZED(3)
    Status time: Mar  5 03:34:54 2015 UTC
    Request Time: Mar  5 03:34:17 2015 UTC

Handle: 3
    License: 'nullPtr'
    Entitlement Tag:
regid.2014-11.com.cisco.WAN_License,1.0_3d8bb7ba-1a92-4f01-a4aa-a4479f1d7612
    Description: <empty>
    Count: 8
    Version: 1.0
    Status: OUT OF COMPLIANCE(4)
    Status time: Mar  5 03:34:54 2015 UTC
    Request Time: Mar  5 03:34:17 2015 UTC

Product Information
===================
```

```
UDI: PID:CBR-8-CCAP-CHASS,SN:FXS1739Q0NT

HA UDI List:
    Active:PID:CBR-8-CCAP-CHASS,SN:FXS1739Q0NT
    Standby:PID:CBR-8-CCAP-CHASS,SN:FXS1739Q0NT

Agent Version
=============
Smart Agent for Licensing: 1.2.1_throttle/5
Component Versions: SA:(1_2_1_throttle)1.1.0, SI:(rel20)1.0.1, CH:(rel4)1.0.15,
PK:(rel16)1.0.7

Upcoming Scheduled Jobs
=======================
Current time: Mar  5 03:37:46 2015 UTC
IdCert Expiration Warning: Jan  4 02:00:41 2016 UTC (304 days, 22 hours, 22 minutes,
55 seconds remaining)
Daily: Mar  6 03:21:11 2015 UTC (23 hours, 43 minutes, 25 seconds remaining)
Certificate Renewal: Sep  1 02:03:51 2015 UTC (179 days, 22 hours, 26 minutes, 5 seconds
 remaining)
Certificate Expiration Check: Mar  4 02:00:41 2016 UTC (364 days, 22 hours, 22 minutes,
 55 seconds remaining)
Authorization Renewal: Mar  5 15:35:57 2015 UTC (11 hours, 58 minutes, 11 seconds
remaining)
Authorization Expiration Check: Jun  3 03:32:51 2015 UTC (89 days, 23 hours, 55 minutes,
 5 seconds remaining)
Init Flag Check: Not Available

License Certificates
====================
Production Cert: True
PIID: 36bf91ae-0577-4213-9e62-1b6ee0add02f
Licensing Certificated:
    Id certificate Info:
        Start Date: Mar  5 01:57:54 2015 UTC
        Expiry Date: Mar  4 01:57:54 2016 UTC
        Version Number: 3
        Serial Number: 134418
        Common Name: 05FB26B1A58A106DEA6878C346432186D08BC1C5::1,2

    Signing certificate Info:
        Start Date: Jun 14 20:18:52 2013 UTC
        Expiry Date: Apr 24 21:55:42 2033 UTC
        Version Number: 3
        Serial Number: 3
        Common Name: MMI Signer

    Sub CA Info:
        Start Date: Apr 24 22:19:15 2013 UTC
        Expiry Date: Apr 24 21:55:42 2033 UTC
        Version Number: 3
        Serial Number: 2
        Common Name: Smart Licensing CA - DEV


HA Info
==========
RP Role: Active
Chassis Role: Active
Behavior Role: Active
RMF: True
CF: True
CF State: Stateless
```

```
Other Info
==========
Software ID: regid.2014-12.com.cisco.CBR8V1,1.0_95948658-0b8b-4e8f-838d-b17020364ca9
Agent State: OOC
TS enable: True
Transport: Callhome
Locale: en_US.UTF-8
Debug flags: 0x7
Privacy Send Hostname: True
Privacy Send IP: True
Build type:: Production
sizeof(char)  : 1
sizeof(int)   : 4
sizeof(long)  : 4
sizeof(char *): 8
sizeof(time_t): 4
sizeof(size_t): 8
Endian: Big
enableOnInit: True
routingReadyByEvent: True
systemInitByEvent: True
WaitForHaRole: False
standbyIsHot: True
chkPtType: 2
delayCommInit: False
roleByEvent: True
maxTraceLength: 150
traceAlwaysOn: False
debugFlags: 7
```

- **show license udi**—Displays the license Unique Device Identifier (UDI) information.

  The following is a sample output of this command:

```
Router# show license udi

UDI: PID:CBR-8-CCAP-CHASS,SN:FXS1739Q0NT

HA UDI List:
    Active:PID:CBR-8-CCAP-CHASS,SN:FXS1739Q0NT
    Standby:PID:CBR-8-CCAP-CHASS,SN:FXS1739Q0NT
```

- **show license usage**—Displays the license usage information.

  The following is a sample output of this command:

```
Router# show license usage

License Authorization:
  Status: OUT OF COMPLIANCE on Mar  5 03:34:54 2015 UTC

(US_License):
  Description:
  Count: 64
  Version: 1.0
  Status: AUTHORIZED

(DS_License):
  Description:
  Count: 768
  Version: 1.0
  Status: AUTHORIZED

(WAN_License):
```

```
  Description:
  Count: 8
  Version: 1.0
  Status: OUT OF COMPLIANCE
```

- **show call-home profile all**—Displays the call home profile information for all configured profiles.

  The following is a sample output of this command:

```
Router# show call-home profile all

Profile Name: CiscoTAC-1
    Profile status: ACTIVE
    Profile mode: Full Reporting
    Reporting Data: Smart Call Home, Smart Licensing
    Preferred Message Format: xml
    Message Size Limit: 3145728 Bytes
    Transport Method: http
    Email address(es): callhome@cisco.com
    HTTP  address(es): https://tools.cisco.com/its/service/oddce/services/DDCEService

    Periodic configuration info message is scheduled every 25 day of the month at 10:03


    Periodic inventory info message is scheduled every 25 day of the month at 09:48

    Alert-group               Severity
    ------------------------  ------------
    crash                     debug
    diagnostic                minor
    environment               minor
    inventory                 normal

    Syslog-Pattern            Severity
    ------------------------  ------------
    .*                        major
```

- **show call-home smart-licensing statistics**—Displays the call home smart licensing statistics information.

  The following is a sample output of this command:

```
Router# show call-home smart-licensing statistics

Success: Successfully sent and response received.
Failed : Failed to send or response indicated error occurred.
Inqueue: In queue waiting to be sent.
Dropped: Dropped due to incorrect call-home configuration.

Msg Subtype      Success Failed  Inqueue Dropped Last-sent (GMT-06:00)
-------------------------------------------------------------------------
REGISTRATION     1       0       0       0       2015-03-13 13:12:13
ACKNOWLEDGEMENT  1       0       0       0       2015-03-13 13:12:20
ENTITLEMENT      5       0       0       0       2015-03-13 13:22:18
```

Use the following commands to verify the DOCSIS 3.1 Downstream License on the Cisco cBR router:

- **show cable license all | begin D3.1**—Displays all the DOCSIS 3.1 downstream license information.

  The following is a sample output of this command:

```
Router# show cable license all | begin D3.1
```

```
Load for five secs: 21%/1%; one minute: 52%; five minutes: 52%
Time source is NTP, 10:41:11.175 PST Mon May 9 2016
------------------------------------------
Entitlement:  DOCSIS 3.1 Downstream Channel License
Consumed count: 31
Consumed count reported to SmartAgent: 0
Enforced state: No Enforcement
```

Troubleshooting Cisco Smart Licensing

Before taking the steps below to troubleshoot the Cisco Smart Licensing, the customers should first make sure the configuration is correct and see if they are able to ping the HTTP address they have configured for the smart license. The output of the **show call-home smart-licensing statistics** command should have REGISTERED and ACKNOWLEDGE information. And check the output of **show logging | include SMART | CALL**.

Manually Renewing the Smart License Registration

The license agent automatically renews the registration information with Cisco every 30 days. You may need to manually renew the registration if the license is out of compliance and it needs to be registered immediately.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**license smart renew** **Example:** `Router# license smart renew`	Manually renews the license registration of the device instance with Cisco.

Unregistering the Router from Cisco Smart Licensing

You can unregister the router from Cisco Smart Licensing. You may need to unregister the router for the Return Material Authorization (RMA) of the router.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**license smart deregister** **Example:** `Router# license smart deregister`	Removes the Cisco Smart Licensing registration for the device instance. All Cisco Smart Licensing certificates and entitlements are removed.

Additional References

Related Documents

Related Topic	Document Title
Cisco Smart Licensing	Cisco Smart Software Licensing

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

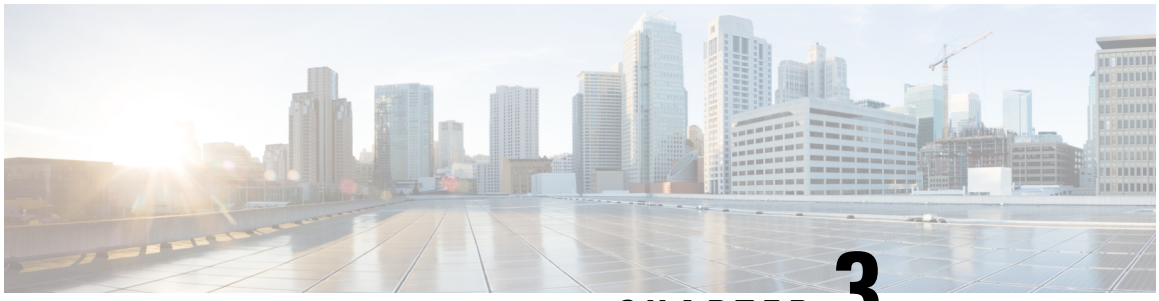
Feature Information for Cisco Smart Licensing

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 2: Feature Information for Cisco Smart Licensing

Feature Name	Releases	Feature Information
Cisco Smart Licensing	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 3

Consolidated Packages and SubPackages Management

This document discusses how consolidated packages and software subpackages (individual and optional) are run and managed on the Cisco cBR Series Converged Broadband Router . It contains the following sections:

- Finding Feature Information, on page 55
- Running the Cisco cBR Series Routers Using Individual and Optional SubPackages: An Overview , on page 56
- Running the Cisco cBR Series Routers Using a Consolidated Package: An Overview , on page 56
- Running the Cisco cBR Series Routers: A Summary , on page 57
- Software File Management Using Command Sets , on page 58
- Managing and Configuring the Router to Run Using Consolidated Packages and Individual SubPackages, on page 59
- Upgrading Individual SubPackages, on page 79
- Additional References, on page 87
- Feature Information for Consolidated Packages and SubPackages Management, on page 87

Finding Feature Information

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Running the Cisco cBR Series Routers Using Individual and Optional SubPackages: An Overview

The Cisco cBR Series Converged Broadband Router can be configured to run using individual subpackages and optional subpackages.

When the router is configured to run using individual and optional subpackages:

- Each individual subpackage within a consolidated package is extracted onto the router as its own file.
- Additionally, any optional subpackages must be separately downloaded and stored in the same directory with the provisioning file and the other individual subpackages that have been extracted.
- The router then runs by accessing each file as needed for operational purposes. All individual and optional subpackage files must be stored in the same directory on the router for the router to run properly using individual subpackages.

When the router runs using the individual and optional subpackages, the router needs to be configured to boot using the provisioning file that was included in the consolidated package with the individual subpackage files. This provisioning file must also be in the same directory as the individual and optional subpackage files. The router boots faster when configured to run using individual and optional subpackages than it does when configured to run using a consolidated package.

A Cisco cBR Series Router cannot be configured to run individual and optional subpackages stored on a TFTP or any other network server. To use this method of running the router, copy the individual and optional subpackages along with the provisioning file onto the bootflash: file system.

Running the Cisco cBR Series Routers Using a Consolidated Package: An Overview

The Cisco cBR Series Converged Broadband Router can also be configured to run using a consolidated package.

Note Booting the router from a consolidated package is not supported for installation of optional subpackages.

When the router is configured to run using a consolidated package, the entire consolidated package file is copied onto the router or accessed by the router via TFTP or another network transport method. The router runs using the consolidated package file.

A router configured to run using a consolidated package is booted by booting the consolidated package file. Because this file is large, the boot process for routers running using the consolidated package is slower than the boot process for routers running individual subpackages.

A router configured to run using a consolidated package does have some advantages over a router configured to run individual subpackages. First, a consolidated package can be booted and utilized using TFTP or another network transport method. Secondly, configuring the router to use the one consolidated package file is easier

than managing several individual subpackage files. Running the router using a consolidated package may be the right method of running the router in certain networking environments.

The consolidated package should be stored on bootflash:, usb[0-1]:, or a remote file system when this method is used to run the router.

Running the Cisco cBR Series Routers: A Summary

The advantages of running your router using individual subpackages include:

- The router boots fastest when booted using the individual subpackage boot approach.
- Individual subpackages can be upgraded instead of the complete consolidated image.

The advantages of running your router using a consolidated package include:

- Simplified installation—Only one software file needs to be managed instead of several separate images.
- Storage—A consolidated package, unlike individual subpackages, can be used to run the router while being stored in bootflash:, on a USB Flash disk, or on a network server. A consolidated package can be booted and utilized using TFTP or another network transport method, while the individual subpackage method requires the individual subpackage files to be copied into the bootflash: file directory on the router.

Approach	Advantages	Disadvantages
Individual and optional subpackages **Note** This method is required if you need to install any optional subpackages for your system.	Faster boot time.	• Multiple software subpackages more difficult to manage. • Cannot be booted from TFTP or any other network server. If you are going to use the individual subpackage boot method, each individual subpackage file must be placed in the bootflash: directory. • Individual and optional subpackage files and the provisioning file must be stored in bootflash:.
Consolidated Package	• Easier management. Only have to manage one file instead of many files. • A consolidated package file can be stored in bootflash:, on any TFTP or other network server.	Slower boot times and lessened maximum system scalability because the larger image must be processed at all times.

Software File Management Using Command Sets

Software files can be managed on the Cisco cBR Series Converged Broadband Router using the following distinct command sets.

The request platform Command Set

The **request platform software package** command is part of the larger request platform command set being introduced on the Cisco cBR Series Converged Broadband Router .

The **request platform software package** command, which can be used to upgrade individual subpackages and a complete consolidated package, is used to upgrade software on the Cisco cBR Series Converged Broadband Router . Notably, the **request platform software package** command is the recommended way of performing an individual subpackage upgrade, and also provides the only method of no-downtime upgrades of individual subpackages on the router when the router is running individual subpackages.

The **request platform software package** command requires that the destination device or process be specified in the command line, so the commands can be used to upgrade software on both an active or a standby processor.

Command Syntax

request platform software package install rp *rp-slot-number* **file** *file-URL*

where

- *rp-slot-number* is the number of the RP slot and
- *file-URL* is the path to the file being used to upgrade the Router.

The copy Command

The **copy** command can be used to move consolidated packages and individual subpackages onto the router, though using this command to move individual subpackage files from one storage area to another is often inefficient (in these scenarios, it is almost always preferable to move the consolidated package, then extract the subpackages, or to extract the subpackages without moving the consolidated package).

To upgrade a consolidated package on the Cisco cBR Series Converged Broadband Router , copy the consolidated package onto a file system, usually bootflash: , using the **copy** command as you would on most other Cisco routers. After making this copy, configure the router to boot using the consolidated package file.

To upgrade the router and reboot using individual subpackages, copy the consolidated package onto the router using the **copy** command, enter the **request platform software package expand** command to extract the individual subpackages, and configure the router to boot using subpackages. Other methods, such as copying each individual subpackage in the same consolidated package from a directory or using the request platform software package command to extract the subpackages onto a router directory are also usable, though copying individual subpackages is often inefficient.

Managing and Configuring the Router to Run Using Consolidated Packages and Individual SubPackages

This section describes methods that are used to manage and configure the packages, sub-packages and patches on the router.

Cable Line Card Process Restart

The last step, in many of the methods used to upgrade and manage the packages and sub-packages, is to reload the router or the component for which the software is being configured or upgraded. Reloading the router or the components involves the following disadvantages:

- service disruption (limited)
- loss of modem configuration data
- time consumption in rebooting the line cards and other components

The N+1 line card high availability (LCHA) system reboots the active and the standby line cards whenever a package is upgraded on a line card. This occurs for sub-package upgrade for Field Replaceable Units (FRUs) on the line card as well. Every time an upgrade is done to a package or sub-package, the line card must be rebooted. The time taken for the package upgrade on N number of active line cards, the total number of reboots would be 2xN. This is time-consuming and may affect services on the rebooting line cards.

To avoid the disadvantages of reloading the router and the line cards, use the Cable Line Card Process Restart features when you upgrade packages on the RF line cards.

Note Do not use the process restart features without upgrading or installing packages on the RF line cards.

Primarily there are two features that allow you to restart Cable Line Card processes:

- Cable Line Card Control Plane Process Restart
- Cable Line Card Upstream Scheduler Process Restart

Cable Line Card Control Plane Process Restart

The Cable Line Card Control Plane Process Restart feature provides the following advantages:

- Simplified package upgrade without LCHA based reboot of active and standby line cards.
- Restart of specific processes without service disruption.
- All the modem configuration data is recovered after the IOSd process restarts.
- Changes in the modem configuration data during the IOSd process restart is reconciled after the IOSd process is restarted.
- The secondary line card shutdown and un-shutdown processes occur automatically.

Restrictions:

- To upgrade a line card IOS using the Cable Line Card Control Plane Process Restart feature, the sub-package must have the patches for cbrsup-clcios and cbrsup-clciosdb.
- IOSd and us-scheduler restart are supported. You can restart IOSd only, which requires IOSd/IOSdb packages or restart the us-scheduler only, which requires clc-docsis package. You can also restart both using a single command, where the us-scheduler is restarted first and then the IOSd is restarted. In this case, IOSd/IOSdb and clc-docsis packages are required.

Other restrictions include the following:

- The IOSd process can be restarted only on the primary active RF line cards.
- Any secondary RF line card must be shut down before restarting the IOSd process.
- The IOSd process restart does not work when double failures (i.e. termination of IOSd and one or more process at the same time) occur. The double failures result in line card reload.
- The restart of the next IOSd (i.e. IOSd process on the next RF line card) does not occur until the current IOSd process recovers fully.

The Cable Line Card Control Plane Process Restart feature provides the following restart options:

- Restart a specific slot using the **request platform software process restart** command with the **slot** *slot number* option.
- Restart all the line cards without specifying a specific slot, using the **request platform software process restart** command without the **slot** *slot number* option.
- Interval based restart option of all line card using the **request platform software process restart** command with the **interval** *secs* option.

Important points to remember:

- Before using the Cable Line Card Control Plane Process Restart feature, shut down the secondary line card, using the **hw-module slot shutdown** command.
- When the Cable Line Card Control Plane Process Restart feature is started, the secondary line card shuts down automatically.
- Before using the Cable Line Card Control Plane Process Restart feature, shut down the secondary line card, using the **hw-module slot shutdown** command.
- When the Cable Line Card Control Plane Process Restart feature is started, the secondary line card shuts down automatically.
- To prevent reloading the line card after sub-package upgrade, use the **request platform software package install node file** command with the **noreload linecard** option.
- Verify the restart process using the **show platform software ios slot** *slot number* **restart info** command.

Restart on Crash

The cable line card control plane and upstream scheduler process restarts automatically after a crash. After restarting process, secondary line card is reset.

Restrictions:

- Restart on crash supported only on primary active line cards

The table below lists the cable line card behaviors when crash happens under different rules.

Table 3: Cable Line Card Process Restart Policy Matrix

	Secondary Cable Line Card Present	Crash	Secondary Cable Line Card state
LCHA Preferred	Yes	Cable line card reset	Active after switchover
Process restart Enabled	No	Restart	N/A
LCHA Preferred	Yes	Cable line card reset	Active after switchover
Process restart Disabled	No	Cable line card reset	N/A
No LCHA Preferred	Yes	Process restart	Secondary cable line card resets after process restart on primary cable line card
Process restart Enabled	No	Process restart	N/A
No LCHA Preferred	Yes	Cable line card reset	Active after switchover
Process restart Disabled	No	Cable line card reset	N/A

Note

- Manual process restart does not depend on policy configured, and will shut/unshut secondary cable line card if present.
- "LCHA preferred" and "process restart enable" are default. User can set these two parameters using disable-auto-restart and lcha-preferred commands. For the details, see http://www.cisco.com/c/en/us/td/docs/cable/cmts/cmd_ref/b_cmts_cable_cmd_ref.html.
- Behavior is the same for Control Plane Process Restart and Upstream Scheduler Process Restart.
- Secondary cable line card in standby mode is considered present.

The restart retry limit feature is added to the Cable Line Card Process Restart, it is applicable only to restart on crash. Using this feature, the customer can set a restart retry time limit, if the process cannot restart successfully within this limit, the line card will reload. This feature can prevent the line card from continuous restart when restart failed.

Using the Cable Line Card Control Plane Process Restart Feature

To use the Cable Line Card Control Plane Process Restart Feature, install the RF line card sub-package upgrade using the command.

Procedure

Step 1 Install the RF line card sub-package upgrade using the **request platform software package install node file noreload linecard** command.

```
Router#request platform software package install node file bootflash:sp/cbr_patch.9.0.tar
noreload linecard
```

Step 2 Use the **request platform software process restart** command to restart the RF line card IOSd process on all the cable line cards sequentially.

```
Router# request platform software process restart
```

Step 3 Use the **request platform software process restart slot** *slot#* command to restart the RF line card IOSd process on a specific cable line card.

```
Router# request platform software process restart slot 7
```

Configuring the Cable Line Card Control Plane Process Restart Retry Limit

To configure the Cable Line Card Control Plane Process Restart Retry Limit, complete the following procedure:

```
enable
configure terminal
process-restart
lc-control-plane-timeout time
restart-retry retry-times
exit
```

Examples for Cable Line Card Control Plane Process Restart Feature

This section provides the sample outputs for the commands used in the Cable Line Card Control Plane Process.

This example shows the output of the **request platform software package install node file** command with the **noreload linecard** option that installs the sub-package upgrade:

```
Router# request platform software package install node file bootflash:sp/cbr_patch.9.0.tar
noreload linecard
Image file expanded and copied
Expanding image file: stby-bootflash:sp/cbr_patch.9.0.tar
Image file expanded and copied
Finished image file expansion
Found clc package
STAGE 1: Installing software on standby RP
=========================================
--- Starting local lock acquisition on R0 ---
Finished local lock acquisition on R0
--- Starting installation state synchronization ---
Finished installation state synchronization
--- Starting local lock acquisition on R1 ---
Finished local lock acquisition on R1
```

```
--- Starting file path checking ---
Finished file path checking
--- Starting image file verification ---
Checking image file names
Locating image files and validating name syntax
Found cbrsup-clcios.2015-03-23_17.53_haolin2.SSA.pkg
Found cbrsup-clciosdb.2015-03-23_17.53_haolin2.SSA.pkg
Verifying image file locations
Inspecting image file types
Processing image file constraints
Creating candidate provisioning file
Finished image file verification
--- Starting candidate package set construction ---
Verifying existing software set
Processing candidate provisioning file
Constructing working set for candidate package set
Constructing working set for running package set
Checking command output
Constructing merge of running and candidate packages
Checking if resulting candidate package set would be complete
Finished candidate package set construction
--- Starting ISSU compatibility verification ---
Verifying image type compatibility
Checking IPC compatibility with running software
Checking candidate package set infrastructure compatibility
Checking infrastructure compatibility with running software
Checking package specific compatibility
Finished ISSU compatibility verification
--- Starting list of software package changes ---
Old files list:
Removed cbrsup-clcios.2015-03-23_17.28_haolin2.SSA.pkg
Removed cbrsup-clciosdb.2015-03-23_17.28_haolin2.SSA.pkg
New files list:
Added cbrsup-clcios.2015-03-23_17.53_haolin2.SSA.pkg
Added cbrsup-clciosdb.2015-03-23_17.53_haolin2.SSA.pkg
Finished list of software package changes
--- Starting commit of software changes ---
Updating provisioning rollback files
Creating pending provisioning file
Committing provisioning file
Finished commit of software changes
--- Starting analysis of software changes ---
Finished analysis of software changes
--- Starting update running software ---
Blocking peer synchronization of operating information
Creating the command set placeholder directory
Finding latest command set
Finding latest command shortlist lookup file
Finding latest command shortlist file
Assembling CLI output libraries
Assembling CLI input libraries
Assembling Dynamic configuration files
Applying interim IPC and database definitions
Replacing running software
Replacing CLI software
Restarting software
Restarting software: target frus filtered out ... skipped
Applying final IPC and database definitions
Generating software version information
Notifying running software of updates
Unblocking peer synchronization of operating information
Unmounting old packages
Cleaning temporary installation files
Finished update running software
```

```
SUCCESS: Finished installing software.
STAGE 2: Installing software on active RP
==========================================
--- Starting local lock acquisition on R0 ---
Finished local lock acquisition on R0
--- Starting installation state synchronization ---
Finished installation state synchronization
--- Starting file path checking ---
Finished file path checking
--- Starting image file verification ---
Checking image file names
Locating image files and validating name syntax
Found cbrsup-clcios.2015-03-23_17.53_haolin2.SSA.pkg
Found cbrsup-clciosdb.2015-03-23_17.53_haolin2.SSA.pkg
Verifying image file locations
Inspecting image file types
Processing image file constraints
Creating candidate provisioning file
Finished image file verification
--- Starting candidate package set construction ---
Verifying existing software set
Processing candidate provisioning file
Constructing working set for candidate package set
Constructing working set for running package set
Checking command output
Constructing merge of running and candidate packages
Checking if resulting candidate package set would be complete
Finished candidate package set construction
--- Starting ISSU compatibility verification ---
Verifying image type compatibility
Checking IPC compatibility with running software
Checking candidate package set infrastructure compatibility
Checking infrastructure compatibility with running software
Checking package specific compatibility
Finished ISSU compatibility verification
--- Starting impact testing ---
Checking operational impact of change
Finished impact testing
--- Starting list of software package changes ---
Old files list:
Removed cbrsup-clcios.2015-03-23_17.28_haolin2.SSA.pkg
Removed cbrsup-clciosdb.2015-03-23_17.28_haolin2.SSA.pkg
New files list:
Added cbrsup-clcios.2015-03-23_17.53_haolin2.SSA.pkg
Added cbrsup-clciosdb.2015-03-23_17.53_haolin2.SSA.pkg
Finished list of software package changes
--- Starting commit of software changes ---
Updating provisioning rollback files
Creating pending provisioning file
Committing provisioning file
Finished commit of software changes
--- Starting analysis of software changes ---
Finished analysis of software changes
--- Starting update running software ---
Blocking peer synchronization of operating information
Creating the command set placeholder directory
Finding latest command set
Finding latest command shortlist lookup file
Finding latest command shortlist file
Assembling CLI output libraries
Assembling CLI input libraries
Assembling Dynamic configuration files
Applying interim IPC and database definitions
Replacing running software
```

```
Replacing CLI software
Restarting software
Restarting software: target frus filtered out ... skipped
Applying final IPC and database definitions
Generating software version information
Notifying running software of updates
Unblocking peer synchronization of operating information
Unmounting old packages
Cleaning temporary installation files
Finished update running software
SUCCESS: Finished installing software.
Found clc package
SUCCESS: node ISSU finished successfully.
Invoking cleanup routine
```

This example shows the output of the **show platform software ios restart info** command:

```
Router#show platform software ios 6 restart info
IOSD process restart info:
Process restartable: Yes
IOSD restart state : ACTIVE
Total Modem Count : 31
Active Modem Count : 31
```

This example shows the output of the **request platform software process restart** command:

```
Router# request platform software process restart
--- Upgrading/Restarting LineCard-2 Packages/Processes ---
NOTICE: No upgrades available.
Provide process name in cli if you wish to restart a process
--- Upgrading/Restarting LineCard-3 Packages/Processes ---
Available upgrades
cbrsup-clcios.2015-03-23_17.53_haolin2.SSA.pkg
cbrsup-clciosdb.2015-03-23_17.53_haolin2.SSA.pkg
--- Checking for ready state before IOSD upgrade on LineCard-3
Updating Package cbrsup-clcios.2015-03-23_17.28_haolin2.SSA.pkg
---> cbrsup-clcios.2015-03-23_17.53_haolin2.SSA.pkg
Restarting ubrclc_k9lc_ms
--- Checking for ready state before IOSDB upgrade on LineCard-3
Updating Package cbrsup-clciosdb.2015-03-23_17.28_haolin2.SSA.pkg
---> cbrsup-clciosdb.2015-03-23_17.53_haolin2.SSA.pkg
Restarting iosdb
SUCCESS: Finished
```

```
Router#show platform software patch 2 info
cbrsup-clciosdb: 3.17.0 (3.0)
cbrsup-clc-firmware: 3.17.0 (0.0)
cbrsup-clcvideo: 3.17.0 (0.0)
cbrsup-clcios: 3.17.0 (3.0)
cbrsup-clccontrol: 3.17.0 (0.0)
cbrsup-clcdocsis: 3.17.0 (0.0)
```

Cable Line Card Upstream Scheduler Process Restart

The Cable Line Card Upstream Scheduler Process Restart feature is used to restart the upstream scheduler process on the RF line cards.

The Cable Line Card Upstream Scheduler Process Restart feature provides the following advantages:

- Allows to restart line card US- Scheduler (CDMAN) process with minimum impact to upstream traffic and no impact to downstream traffic.
- All modem configuration data recovered to the state before restart.
- Changes in modem configuration data during restart are reconciled.
- New modems coming online are blocked during restart.
- The **request platform software package restart** command is used to upgrade the new DOCSIS sub-pkg patch without reloading the line card.
- Effective with Cisco IOS-XE Release 3.18.0S, the card restarts automatically after a crash. For more information, see Restart on Crash, on page 60 section.

Restrictions:

The following restrictions apply to the :

- The Upstream Scheduler process can be restarted only on the primary active RF line cards.
- The Upstream Scheduler process restart does not work when double failures (i.e. termination of Upstream Scheduler and one or more process at the same time) occur. The double failures result in line card reload.
- The restart of the next Upstream Scheduler (i.e. Upstream Scheduler process on the next RF line card) does not occur until the current Upstream Scheduler process recovers fully.

Important points to remember:

- With Cisco IOS-XE Release 3.17.0S, when the Cable Line Card Upstream Scheduler Process Restart feature is started, the secondary line card shuts down automatically.
- To prevent reloading the line card after sub-package upgrade, use the **request platform software package install node file** command with the **noreload linecard** option.
- Verify the restart process using the **show platform software us-scheduler restart info** command.

```
Router# show platform software us-scheduler 3 restart info
us-scheduler process restart info:
Process restartable : Yes
us-scheduler state : RESTART_OPERATIONAL
Features bit map : 0x001e
us-scheduler restart count : 4
```

Using the Cable Line Card Upstream Scheduler Process Restart Feature

To use the Cable Line Card Upstream Scheduler Process Restart Feature, install the RF line card sub-package upgrade using the command.

Procedure

Step 1 Install the RF line card sub-package upgrade using the **request platform software package install node file noreload linecard** command.

```
Router#request platform software package install node file bootflash:sp/cbr_patch.9.0.tar
noreload linecard
```

Step 2 Use the **request platform software process restart** command to restart the Upstream Scheduler process on all the cable line cards sequentially.

```
Router# request platform software process restart
```

Step 3 Use the **request platform software process restart slot** *slot#* command to restart the Upstream Scheduler process on a specific cable line card.

```
Router# request platform software process restart slot 7
```

Configuring the Cable Line Card Control Plane Process Restart Retry Limit

To configure the Cable Line Card Control Plane Process Restart Retry Limit, complete the following procedure:

```
enable
configure terminal
process-restart
lc-us-scheduler-timeout time
restart-retry retry-times
exit
```

Examples for Cable Line Card Upstream Scheduler Process Restart Feature

This section provides the sample outputs for the commands used in the Cable Line Card Upstream Scheduler Process Restart Feature.

This example shows the output of the **show platform software us-scheduler restart info** command.

```
Router#show platform software us-scheduler 6 restart info
us-scheduler process restart info:
Process restartable : Yes
us-scheduler state : RESTART_OPERATIONAL
Features bit map : 0x001e
us-scheduler restart count : 1
```

This example shows the output when the upstream scheduler process is restarted.

```
Router# request platform software package install node file
bootflash:subpkg/cbr_patch-3.17.0-patch2.tar noreload linecard
NOTE: Currently node has booted from a provisioning file
NOTE: Going to start a dual rp sub-packages node ISSU install
--- Starting initial file path checking --- Copying
bootflash:subpkg/cbr_patch-3.17.0-patch2.tar to
stby-bootflash:subpkg/cbr_patch-3.17.0-patch2.tar
Finished initial file path checking
--- Starting config-register verification --- Finished config-register verification
```

```
--- Starting Checking noreload options --- Finished Checking noreload options
--- Starting image file expansion ---
Expanding image file: bootflash:subpkg/cbr_patch-3.17.0-patch2.tar
Image file expanded and copied
Expanding image file: stby-bootflash:subpkg/cbr_patch-3.17.0-patch2.tar
Image file expanded and copied
Finished image file expansion
Found clc package
STAGE 1: Installing software on standby RP ========================================
--- Starting local lock acquisition on R0 --- Finished local lock acquisition on R0
--- Starting installation state synchronization --- Finished installation state
synchronization
--- Starting local lock acquisition on R1 --- Finished local lock acquisition on R1
--- Starting file path checking ---
Finished file path checking
--- Starting image file verification --- Checking image file names Locating image files and
validating name syntax
Found cbrsup-clcdocsis.2015-10-08_18.10_haolin2.SSA.pkg
Verifying image file locations
Inspecting image file types
Processing image file constraints
Creating candidate provisioning file
Finished image file verification
--- Starting candidate package set construction --- Verifying existing software set Processing
candidate provisioning file Constructing working set for
candidate package set Constructing working set for running package set Checking command
output Constructing merge of running and candidate packages
Checking if resulting candidate package set would be complete Finished candidate package
set construction
--- Starting ISSU compatibility verification ---
WARNING:
WARNING: ISSU between engineering builds with release strings in non-standard fo rmat.
Skipping ISSU Software Compatibility checks.
WARNING:
WARNING:
WARNING: ISSU between engineering builds with release strings in non-standard fo rmat.
Skipping ISSU Software Compatibility checks.
WARNING:
WARNING:
WARNING: ISSU between engineering builds with release strings in non-standard fo rmat.
Skipping ISSU Software Compatibility checks.
WARNING:
WARNING:
WARNING: ISSU between engineering builds with release strings in non-standard fo rmat.
Skipping ISSU Software Compatibility checks.
WARNING:
Verifying image type compatibility
Checking IPC compatibility with running software Checking candidate package set infrastructure
compatibility Checking infrastructure compatibility with running
software Checking package specific compatibility Finished ISSU compatibility verification
--- Starting list of software package changes --- Old files list:
Removed cbrsup-clcdocsis.BLD_MCP_DEV_LATEST_20151006_133623.SSA.pkg
New files list:
Added cbrsup-clcdocsis.2015-10-08_18.10_haolin2.SSA.pkg
Finished list of software package changes
--- Starting commit of software changes --- Updating provisioning rollback files Creating
pending provisioning file Committing provisioning file Finished
commit of software changes
--- Starting analysis of software changes --- Finished analysis of software changes
--- Starting update running software --- Blocking peer synchronization of operating
information Creating the command set placeholder directory
Finding latest command set
Finding latest command shortlist lookup file
Finding latest command shortlist file
```

```
Assembling CLI output libraries
Assembling CLI input libraries
Assembling Dynamic configuration files
Applying interim IPC and database definitions
Replacing running software
Replacing CLI software
Restarting software
Restarting software: target frus filtered out ... skipped
Applying final IPC and database definitions
Generating software version information
Notifying running software of updates
Unblocking peer synchronization of operating information Unmounting old packages Cleaning
temporary installation files
Finished update running software
SUCCESS: Finished installing software.
STAGE 2: Installing software on active RP ========================================
--- Starting local lock acquisition on R0 --- Finished local lock acquisition on R0
--- Starting installation state synchronization --- Finished installation state
synchronization
--- Starting file path checking ---
Finished file path checking
--- Starting image file verification --- Checking image file names Locating image files and
validating name syntax
Found cbrsup-clcdocsis.2015-10-08_18.10_haolin2.SSA.pkg
Verifying image file locations
Inspecting image file types
Processing image file constraints
Creating candidate provisioning file
Finished image file verification
--- Starting candidate package set construction --- Verifying existing software set Processing
candidate provisioning file Constructing working set for candidate
package set Constructing working set for running package set Checking command output
Constructing merge of running and candidate packages Checking if resulting
candidate package set would be complete Finished candidate package set construction
--- Starting ISSU compatibility verification ---
WARNING:
WARNING: ISSU between engineering builds with release strings in non-standard fo rmat.
Skipping ISSU Software Compatibility checks.
WARNING:
WARNING:
WARNING: ISSU between engineering builds with release strings in non-standard fo rmat.
Skipping ISSU Software Compatibility checks.
WARNING:
WARNING:
WARNING: ISSU between engineering builds with release strings in non-standard fo rmat.
Skipping ISSU Software Compatibility checks.
WARNING:
WARNING:
WARNING: ISSU between engineering builds with release strings in non-standard fo rmat.
Skipping ISSU Software Compatibility checks.
WARNING:
Verifying image type compatibility
Checking IPC compatibility with running software Checking candidate package set infrastructure
compatibility Checking infrastructure compatibility with running
software Checking package specific compatibility Finished ISSU compatibility verification
--- Starting impact testing ---
Checking operational impact of change
Finished impact testing
--- Starting list of software package changes --- Old files list:
Removed cbrsup-clcdocsis.BLD_MCP_DEV_LATEST_20151006_133623.SSA.pkg
New files list:
Added cbrsup-clcdocsis.2015-10-08_18.10_haolin2.SSA.pkg
Finished list of software package changes
--- Starting commit of software changes --- Updating provisioning rollback files Creating
```

```
pending provisioning file Committing provisioning file Finished
commit of software changes
--- Starting analysis of software changes --- Finished analysis of software changes
--- Starting update running software --- Blocking peer synchronization of operating
information Creating the command set placeholder directory
Finding latest command set
Finding latest command shortlist lookup file
Finding latest command shortlist file
Assembling CLI output libraries
Assembling CLI input libraries
Assembling Dynamic configuration files
Applying interim IPC and database definitions
Replacing running software
Replacing CLI software
Restarting software
Restarting software: target frus filtered out ... skipped
Applying final IPC and database definitions
Generating software version information
Notifying running software of updates
Unblocking peer synchronization of operating information Unmounting old packages Cleaning
temporary installation files
Finished update running software
SUCCESS: Finished installing software.
Found clc package
SUCCESS: node ISSU finished successfully.
Invoking cleanup routine
Router#

Router#show platform software us-scheduler 2 restart info
us-scheduler process restart info:
Process restartable : Yes
us-scheduler state : RESTART_OPERATIONAL
Features bit map : 0x001e
us-scheduler restart count : 1

Router#request platform software process restart slot 2
--- Upgrading/Restarting LineCard-2 Packages/Processes ---
Available upgrades
cbrsup-clcdocsis.BLD_MCP_DEV_LATEST_20151006_13362
Updating Package cbrsup-clcdocsis.BLD_MCP_DEV_LATEST_20151006_13362
---> cbrsup-clcdocsis.2015-10-08_18.10_haolin2.SSA.pkg
Restarting us-scheduler
SUCCESS: Finished upgrading the LineCard-2

Router#show platform software patch 2 info
cbrsup-clciosdb: 3.17.0 (0.0)
cbrsup-clc-firmware: 3.17.0 (0.0)
cbrsup-clcvideo: 3.17.0 (0.0)
cbrsup-clcios: 3.17.0 (0.0)
cbrsup-clccontrol: 3.17.0 (0.0)
cbrsup-clcdocsis: 3.17.0 (2.0)
```

Note If the upgrade package includes both IOSD-CLC and US-scheduler sub-packages, the **request platform software process restart** command first restarts the Cable Line Card Upstream Scheduler process and then the Cable Line Card Control Plane process.

This example shows the output of the **request platform software process restart** command when the Control Plane and the upstream scheduler process are restarted:

Note If the **slot** keyword is not used, the upstream scheduler process on all the line cards are restarted sequentially.

```
Router#request platform software process restart


--- Upgrading/Restarting LineCard-1 Packages/Processes ---

    Available upgrades
        cbrsup-clcdocsis.2015-09-24_03.09_johuynh.SSA.pkg
        cbrsup-clcios.2015-09-24_03.09_johuynh.SSA.pkg
        cbrsup-clciosdb.2015-09-24_03.09_johuynh.SSA.pkg

    Updating Package  cbrsup-clcdocsis.2015-09-24_03.09_johuynh.SSA.pkg
              ---> cbrsup-clcdocsis.2015-09-24_19.04_haolin2.SSA.pkg
    Restarting us-scheduler
--- Checking for ready state before IOSD upgrade on LineCard-1
    Updating Package  cbrsup-clcios.2015-09-24_03.09_johuynh.SSA.pkg
              ---> cbrsup-clcios.2015-09-24_19.04_haolin2.SSA.pkg
    Restarting ubrclc_k9lc_ms
--- Checking for ready state before IOSDB upgrade on LineCard-1
    Updating Package  cbrsup-clciosdb.2015-09-24_03.09_johuynh.SSA.pkg
              ---> cbrsup-clciosdb.2015-09-24_19.04_haolin2.SSA.pkg
    Restarting iosdb

    SUCCESS: Finished upgrading the LineCard-1


--- Upgrading/Restarting LineCard-2 Packages/Processes ---

    Available upgrades
        cbrsup-clcdocsis.2015-09-24_03.09_johuynh.SSA.pkg
        cbrsup-clcios.2015-09-24_03.09_johuynh.SSA.pkg
        cbrsup-clciosdb.2015-09-24_03.09_johuynh.SSA.pkg

    Updating Package  cbrsup-clcdocsis.2015-09-24_03.09_johuynh.SSA.pkg
              ---> cbrsup-clcdocsis.2015-09-24_19.04_haolin2.SSA.pkg
    Restarting us-scheduler
--- Checking for ready state before IOSD upgrade on LineCard-2
    Updating Package  cbrsup-clcios.2015-09-24_03.09_johuynh.SSA.pkg
              ---> cbrsup-clcios.2015-09-24_19.04_haolin2.SSA.pkg
    Restarting ubrclc_k9lc_ms
--- Checking for ready state before IOSDB upgrade on LineCard-2
    Updating Package  cbrsup-clciosdb.2015-09-24_03.09_johuynh.SSA.pkg
              ---> cbrsup-clciosdb.2015-09-24_19.04_haolin2.SSA.pkg
    Restarting iosdb

    SUCCESS: Finished upgrading the LineCard-2
Router#
```

You can specify a time interval in seconds using the **interval** keyword, between the restarting of two line card processes in the sequence. The default interval is five seconds.

This example shows the configuration of an interval of six seconds.

```
Router#request platform software process restart interval 6
```

Quick Start Software Upgrade

The following instructions provide a quick start version of upgrading the software running the Cisco cBR Series Converged Broadband Router . These instructions assume you have access to the consolidated package and that the files will be stored in a bootflash: file system that is not storing any previously installed subpackages or consolidated packages and that has enough room for the file or files.

Procedure

Step 1 Copy the consolidated package into bootflash: using the **copy** *URL-to-image* **bootflash:** command.

Step 2 If you want to run the router using individual subpackages, enter the **request platform software package expand file bootflash:/***sub_dir*/*base_image* command. If you want to run the router using a consolidated package, skip this step.

Step 3 Enter the **dir bootflash:** command to verify your consolidated package or your extracted subpackages are in the directory.

Step 4 If you are trying to run individual subpackages, use the **delete bootflash:***base_image* to delete the consolidated package. If you want to run the router using the consolidated package, skip this step.

Step 5 Set up the boot parameters for your boot. Set the configuration register to 0x2 by entering the **config-register 0x2102** global configuration command, and enter the **boot system flash bootflash:***base_image* (if running using the consolidated package) or **boot system flash bootflash:***provisionging-file-name* (if running using individual subpackages) global configuration command.

Step 6 Enter **copy running-config startup-config** to save your configuration.

Step 7 Enter the **reload** command to reload the router and finish the boot. The upgraded software should be running when the reload completes.

Managing and Configuring a Consolidated Package Using the copy Command

To upgrade a consolidated package on the Cisco cBR Series Routers using the **copy** command, copy the consolidated package into the bootflash: directory on the router using the **copy** command as you would on most other Cisco routers. After making this copy, configure the router to boot using the consolidated package file.

In the following example, the consolidated package file is copied onto the bootflash: file system from TFTP. The config-register is then set to boot using **boot system** commands, and the **boot system** commands instruct the router to boot using the consolidated package stored in the bootflash: file system. The new configuration is then saved using the **copy running-config startup-config** command, and the system is then reloaded to complete the process.

```
Router# dir bootflash:
Directory of bootflash:/

11 drwx 16384 Dec 4 2007 04:32:46 -08:00 lost+found
86401 drwx 4096 Dec 4 2007 06:06:24 -08:00 .ssh
14401 drwx 4096 Dec 4 2007 06:06:36 -08:00 .rollback_timer
28801 drwx 4096 Mar 18 2008 17:31:17 -07:00 .prst_sync
43201 drwx 4096 Dec 4 2007 04:34:45 -08:00 .installer
13 -rw- 45977 Apr 9 2008 16:48:46 -07:00 target_support_output.tgz.tgz
```

```
928862208 bytes total (712273920 bytes free)

Router# copy tftp bootflash:
Address or name of remote host []? 172.17.16.81
Source filename []?
/auto/tftp-users/user/cbrsup-universal*.bin Destination filename [cbrsup-universal*.bin]?
Accessing
tftp://172.17.16.81//auto/tftp-users/user/ cbrsup-universal*.bin...

Loading /auto/tftp-users/user/cbrsup-universal*.bin from
172.17.16.81 (via GigabitEthernet0):
!!!
```

Extracting a Consolidated Package and Booting Using the Provisioning File

T o extract a consolidated package and to boot using provisioning file, perform the following steps:

Procedure

Step 1 Perform one of the following tasks:

a) Copy the consolidated package file (or, in cases where you have every individual subpackage and a provisioning file for the subpackages available, each individual subpackage and the provisioning file) onto the bootflash: file system using the **copy** command. Make sure to copy the consolidated package into the bootflash: file system and directory where you want to store the provisioning file and the individual image subpackages. Enter the **request platform software package expand file bootflash:***url-to-Cisco-IOS-XE-imagename* command with no other option to extract the provisioning file and the individual subpackages out of the consolidated package file and into the current directory in bootflash:. We recommend that you extract the sub packages to an empty directory to simplify the management of these files.

b) Copy the consolidated package file onto any file system on your router, then enter the **request platform software package expand file file-system:***url-to-Cisco-IOS-XE-imagename* **to bootflash:** command to extract the provisioning file and the individual image subpackages onto the bootflash: file system.

Note After performing this step, do not move any of the files. The bootup process cannot function properly unless all of the subpackages and the provisioning file are located in the same directory. Also, do not rename the subpackage files. Only the provisioning file can be renamed, and the renaming of the provisioning file, if desired, should be done at this step before the router is rebooted.

Step 2 Configure the router to boot using the provisioning file.

The sequence below provides an example that would boot the router using the provisioning file named packages.conf that was stored with the other subpackages in the bootflash: file system:

```
Router(config)# no boot system
Router(config)# config-register 0x2102
Router(config)# boot system flash bootflash:packages.conf
Router(config)# exit
*May 11 01:31:04.815: %SYS-5-CONFIG_I: Configured from console by con
Router# copy running-config startup-config
Building configuration... [OK]
Router# reload
```

Extracting the SubPackages and the Provisioning File: Example 1

The following example shows how to extract the individual subpackages and the provisioning file from a consolidated package that has already been placed in the directory where you want to store the individual subpackages and the provisioning file. We recommend that you extract the sub packages to an empty directory to simplify the management of these files.

Output of the directory before and after the extraction is given to confirm the files were extracted.

```
Router# dir bootflash:
Directory of bootflash:/

11 drwx 16384 Dec 4 2007 04:32:46 -08:00 lost+found
```

```
86401 drwx 4096 Dec 4 2007 06:06:24 -08:00 .ssh
14401 drwx 4096 Dec 4 2007 06:06:36 -08:00 .rollback_timer
28801 drwx 4096 Mar 18 2008 17:31:17 -07:00 .prst_sync
43201 drwx 4096 Dec 4 2007 04:34:45 -08:00 .installer
12 -rw- 208904396 May 9 2008 14:36:31 -07:00
cbrsup-universal*.bin
13 -rw- 45977 Apr 9 2008 16:48:46 -07:00 target_support_output.tgz.tgz

928862208 bytes total (503156736 bytes free)

Router# request platform software package expand file bootflash:cbrsup-universal*.bin
Verifying parameters
Validating package type
Copying package files
SUCCESS: Finished expanding all-in-one software package.

Router# dir bootflash:
Directory of bootflash:/

11 drwx 16384 Dec 4 2007 04:32:46 -08:00 lost+found
86401 drwx 4096 Dec 4 2007 06:06:24 -08:00 .ssh
14401 drwx 4096 Dec 4 2007 06:06:36 -08:00 .rollback_timer
28801 drwx 4096 Mar 18 2008 17:31:17 -07:00 .prst_sync
43201 drwx 4096 Dec 4 2007 04:34:45 -08:00 .installer
12 -rw- 208904396 May 9 2008 14:36:31 -07:00
cbrsup-universal*.bin
57611 -rw- 47071436 May 22 2008 11:26:23 -07:00 cbr000rp1-espbase.02.01.00.122-33.XNA.pkg
57602 -rw- 5740 May 22 2008 11:26:22 -07:00
cbr000rp1-packages-adventerprisek9.02.01.00.122-33.XNA.conf
57612 -rw- 20334796 May 22 2008 11:26:24 -07:00
cbr000rp1-rpaccess.02.01.00.122-33.XNA.pkg
57613 -rw- 22294732 May 22 2008 11:26:24 -07:00 cbr000rp1-rpbase.02.01.00.122-33.XNA.pkg
57614 -rw- 21946572 May 22 2008 11:26:25 -07:00 cbr000rp1-rpcontrol.02.01.00.122-33.XNA.pkg
57615 -rw- 48099532 May 22 2008 11:26:26 -07:00
cbr000rp1-rpios-adventerprisek9.02.01.00.122-33.XNA.pkg
57616 -rw- 34324684 May 22 2008 11:26:27 -07:00 cbr000rp1-sipbase.02.01.00.122-33.XNA.pkg
57617 -rw- 22124748 May 22 2008 11:26:28 -07:00 cbr000rp1-sipspa.02.01.00.122-33.XNA.pkg
57603 -rw- 6256 May 22 2008 11:26:28 -07:00 packages.conf
13 -rw- 45977 Apr 9 2008 16:48:46 -07:00 target_support_output.tgz.tgz

928862208 bytes total (286662656 bytes free)
```

Extracting the SubPackages, Configuring the Router to Boot Using the Provisioning File, and Reloading the Router: Example 2

In the following example, the provisioning file and the individual subpackages are extracted from a consolidated package. The router is then configured to boot using the provisioning file. This example also shows the config-register being set and the running configuration being saved because these tasks must be performed for the router to reload properly. The router is then reloaded to complete the process.

```
Router# dir bootflash:
Directory of bootflash:/

11 drwx 16384 Dec 4 2007 04:32:46 -08:00 lost+found
86401 drwx 4096 Dec 4 2007 06:06:24 -08:00 .ssh
14401 drwx 4096 Dec 4 2007 06:06:36 -08:00 .rollback_timer
28801 drwx 4096 Mar 18 2008 17:31:17 -07:00 .prst_sync
43201 drwx 4096 Dec 4 2007 04:34:45 -08:00 .installer
12 -rw- 208904396 May 9 2008 14:36:31 -07:00
cbrsup-universal*.bin
13 -rw- 45977 Apr 9 2008 16:48:46 -07:00 target_support_output.tgz.tgz
```

```
928862208 bytes total (503156736 bytes free)

Router# request platform software package expand file bootflash:cbrsup-universal*.bin
Verifying parameters
Validating package type
Copying package files
SUCCESS: Finished expanding all-in-one software package.

Router# dir bootflash:
Directory of bootflash:/

11 drwx 16384 Dec 4 2007 04:32:46 -08:00 lost+found
86401 drwx 4096 Dec 4 2007 06:06:24 -08:00 .ssh
14401 drwx 4096 Dec 4 2007 06:06:36 -08:00 .rollback_timer
28801 drwx 4096 Mar 18 2008 17:31:17 -07:00 .prst_sync
43201 drwx 4096 Dec 4 2007 04:34:45 -08:00 .installer
12 -rw- 208904396 May 9 2008 14:36:31 -07:00
cbrsup-universal*.bin
57611 -rw- 47071436 May 22 2008 11:26:23 -07:00 cbr000rp1-espbase.02.01.00.122-33.XNA.pkg
57602 -rw- 5740 May 22 2008 11:26:22 -07:00
cbr000rp1-packages-adventerprisek9.02.01.00.122-33.XNA.conf
57612 -rw- 20334796 May 22 2008 11:26:24 -07:00 cbr000rp1-rpaccess.02.01.00.122-33.XNA.pkg
57613 -rw- 22294732 May 22 2008 11:26:24 -07:00 cbr000rp1-rpbase.02.01.00.122-33.XNA.pkg
57614 -rw- 21946572 May 22 2008 11:26:25 -07:00
cbr000rp1-rpcontrol.02.01.00.122-33.XNA.pkg
57615 -rw- 48099532 May 22 2008 11:26:26 -07:00
cbr000rp1-rpios-adventerprisek9.02.01.00.122-33.XNA.pkg
57616 -rw- 34324684 May 22 2008 11:26:27 -07:00 cbr000rp1-sipbase.02.01.00.122-33.XNA.pkg
57617 -rw- 22124748 May 22 2008 11:26:28 -07:00
cbr000rp1-sipspa.02.01.00.122-33.XNA.pkg
57603 -rw- 6256 May 22 2008 11:26:28 -07:00 packages.conf
13 -rw- 45977 Apr 9 2008 16:48:46 -07:00 target_support_output.tgz.tgz

928862208 bytes total (286662656 bytes free)

Router(config)# no boot system
Router(config)# config-register 0x2102
Router(config)# boot system flash bootflash:packages.conf
Router(config)# exit
Router# copy run start
Router# reload
```

Copying a Set of Individual SubPackage Files, and Booting Using a Provisioning File

To copy a set of individual subpackage files and to boot using a provisioning file, perform the following steps:

Note Although this upgrade method works, it is less efficient than other methods of upgrading the router's software.

Procedure

Step 1 Copy each individual subpackage and the provisioning file into the bootflash: directory using the **copy** command. Note that this method of running the router will only work if all the individual subpackages for a

release and a provisioning file are downloaded onto the router and stored in the bootflash: directory. No other file directories should be used for booting the router using individual subpackages.

The files can also be moved on the router physically using a USB Flash drive.

Step 2 Configure the router to boot using the provisioning file.

The sequence below provides an example that describes how to boot the router using the provisioning file named packages.conf that was stored with the other subpackages in the bootflash: file system. The router runs using individual subpackages once the reload is complete.

```
Router(config)# no boot system
Router(config)# config-register 0x2102
Router(config)# boot system flash bootflash:packages.conf
Router(config)# exit
*May 11 01:31:04.815: %SYS-5-CONFIG_I: Configured from console by con
Router# write memory Building configuration... [OK]
Router# reload
```

Installing an Optional SubPackage

To run the router using an optional subpackage, perform the following steps for each Supervisor in the chassis:

Procedure

Step 1 Verify that the Supervisor is running in individual subpackage mode and was booted from a provisioning file.

Step 2 Verify that the version of the optional subpackage that you want to install is the same version as the software running on the active Supervisor.

Step 3 Download the optional subpackage that you want to install. Optional subpackages must be downloaded independently from consolidated packages for the Cisco cBR Series Routers.

Step 4 On each Supervisor, copy the optional subpackage to the directory where any other individual subpackages and the provisioning file is located.

Step 5 Run the **request platform software package install rp file** command, as shown in the following example.

Note Do not use the optional slot or bay keywords for the initial installation.

```
Router# request platform software package install rp 0 file
bootflash: cbrsup-universal*.bin
--- Starting local lock acquisition on R0 --- Finished local lock acquisition on R0

--- Starting file path checking --- Finished file path checking

--- Starting image file verification --- Checking image file names Verifying image file
locations Locating image files and validating name syntax
Found cbrsup-universal*.bin
Inspecting image file types Processing image file constraints Creating candidate provisioning
 file

WARNING: No package of type sipspawmak9 is installed.
WARNING: Package will be installed for all SIP slots and bays. Finished image file
verification
```

```
--- Starting candidate package set construction --- Verifying existing software set Processing
 candidate provisioning file Constructing working set for candidate package set Constructing
 working set for running package set Checking command output Constructing merge of running
 and candidate packages Checking if resulting candidate package set would be complete
Finished candidate package set construction

--- Starting compatibility testing ---
Determining whether candidate package set is compatible

WARNING:
WARNING: Candidate software combination not found in compatibility database
WARNING:

Determining whether installation is valid

WARNING:
WARNING: Candidate software combination not found in compatibility database
WARNING:

Software sets are identified as compatible Checking IPC compatibility with running software
 Checking candidate package set infrastructure compatibility Checking infrastructure
compatibility with running software Checking package specific compatibility Finished
compatibility testing

--- Starting impact testing --- Checking operational impact of change Finished impact testing

--- Starting list of software package changes --- No old package files removed New files
list:
Added cbrsup-universal*.bin Finished list of software package changes

--- Starting commit of software changes --- Updating provisioning rollback files Creating
pending provisioning file Committing provisioning file Finished commit of software changes

--- Starting analysis of software changes --- Finished analysis of software changes

--- Starting update running software --- Blocking peer synchronization of operating
information Creating the command set placeholder directory
Finding latest command set
Finding latest command shortlist lookup file Finding latest command shortlist file Assembling
 CLI output libraries
Assembling CLI input libraries
Applying interim IPC and database definitions
Replacing running software Replacing CLI software Restarting software
Restarting software: target frus filtered out ... skipped
Applying final IPC and database definitions Generating software version information Notifying
 running software of updates
Unblocking peer synchronization of operating information Unmounting old packages
Cleaning temporary installation files
Finished update running software

SUCCESS: Finished installing software.
```

Upgrading Individual SubPackages

Patch Installation

Individual subpackages can be upgraded in subpackage mode. Patch releases consist of one or more subpackages. After a patch has been installed, a message is displayed indicating whether the entire chassis or only the line cards must be rebooted for the updates to take effect.

Installing a Patch that Affects Both Line Card and Supervisor Card

Procedure

Step 1 The Cisco cBR router must be in subpackage mode.

Step 2 Copy the patch file to the same location as the active subpackages on both the active and standby supervisor cards.

Step 3 Install the patch using the **request platform software package install rp** *slot* **file** *patch file* command on the standby card.

Step 4 Install the patch using the**request platform software package install rp** *slot* **file** *patch file* command on the active card.

Step 5 Reload the chassis.

Installing a Patch that Affects Only Line Cards

Procedure

Step 1 The Cisco cBR router must be in subpackage mode.

Step 2 Copy the patch file to the same location as the active subpackages on both the active and standby supervisor cards.

Step 3 Install the patch using the **request platform software package install rp** *slot* **file** *patch file* command on the standby card.

Step 4 Install the patch using the**request platform software package install rp** *slot* **file** *patch file* command on the active card.

Step 5 Reload the line cards.

Installing a Patch that Affects Only Supervisor Cards

Procedure

Step 1 The Cisco cBR router must be in subpackage mode.

Step 2 Copy the patch file to the same location as the active subpackages on both the active and standby supervisor cards.

Step 3 Install the patch using the **request platform software package install rp** *slot* **file** *patch file* command on the standby card.

Step 4 Switch over to the standby card. This will make it the active card.

Step 5 Install the patch using the **request platform software package install rp** *slot* **file** *patch file* command on the standby card.

Upgrading a Line Card SubPackage

Use the **request platform software package install node file** *filename* command to upgrade a line card subpackage.

```
Router# request platform software package install node file
bootflash:/subpkg/cbr_patch.5.0.tar

NOTE: Currently node has booted from a provisioning file
NOTE: Going to start a dual rp sub-packages node ISSU install

--- Starting initial file path checking ---
Copying bootflash:/subpkg/cbr_patch.5.0.tar to stby-bootflash:/subpkg/cbr_patch.5.0.tar
Finished initial file path checking

--- Starting config-register verfication ---
Finished config-register verfication

--- Starting image file expansion ---
Expanding image file: bootflash:/subpkg/cbr_patch.5.0.tar
Image file expanded and copied
Expanding image file: stby-bootflash:/subpkg/cbr_patch.5.0.tar
Image file expanded and copied
Finished image file expansion


STAGE 1: Installing software on standby RP
==========================================
--- Starting local lock acquisition on R0 ---
Finished local lock acquisition on R0

--- Starting installation state synchronization ---
Finished installation state synchronization

--- Starting local lock acquisition on R1 ---
Finished local lock acquisition on R1

--- Starting file path checking ---
Finished file path checking

--- Starting image file verification ---
Checking image file names
Locating image files and validating name syntax
  Found cbrsup-clcdocsis.2015-02-20_01.02.SSA.pkg
Verifying image file locations
Inspecting image file types
Processing image file constraints
Creating candidate provisioning file
Finished image file verification
```

```
--- Starting candidate package set construction ---
Verifying existing software set
Processing candidate provisioning file
Constructing working set for candidate package set
Constructing working set for running package set
Checking command output
Constructing merge of running and candidate packages
Checking if resulting candidate package set would be complete
Finished candidate package set construction

--- Starting ISSU compatiblity verficiation ---
Verifying image type compatibility
Checking IPC compatibility with running software
Checking candidate package set infrastructure compatibility
Checking infrastructure compatibility with running software
Checking package specific compatibility
Finished ISSU compatiblity verficiation

--- Starting list of software package changes ---
Old files list:
  Removed cbrsup-clcdocsis.BLD_V155_2_S_XE315_THROTTLE_LATEST_20150217_110041-st
d.SSA.pkg
New files list:
  Added cbrsup-clcdocsis.2015-02-20_01.02.SSA.pkg
Finished list of software package changes

--- Starting commit of software changes ---
Updating provisioning rollback files
Creating pending provisioning file
Committing provisioning file
Finished commit of software changes

--- Starting analysis of software changes ---
Finished analysis of software changes

--- Starting update running software ---
Blocking peer synchronization of operating information
Creating the command set placeholder directory
  Finding latest command set
  Finding latest command shortlist lookup file
  Finding latest command shortlist file
  Assembling CLI output libraries
  Assembling CLI input libraries
  Assembling Dynamic configuration files
  Applying interim IPC and database definitions
  Replacing running software
  Replacing CLI software
  Restarting software
  Restarting software: target frus filtered out ... skipped
  Applying final IPC and database definitions
  Generating software version information
  Notifying running software of updates
  Unblocking peer synchronization of operating information
Unmounting old packages
Cleaning temporary installation files
  Finished update running software

SUCCESS: Finished installing software.

STAGE 2: Installing software on active RP
=========================================
--- Starting local lock acquisition on R0 ---
Finished local lock acquisition on R0
```

```
--- Starting installation state synchronization ---
Finished installation state synchronization

--- Starting file path checking ---
Finished file path checking

--- Starting image file verification ---
Checking image file names
Locating image files and validating name syntax
  Found cbrsup-clcdocsis.2015-02-20_01.02.SSA.pkg
Verifying image file locations
Inspecting image file types
Processing image file constraints
Creating candidate provisioning file
Finished image file verification

--- Starting candidate package set construction ---
Verifying existing software set
Processing candidate provisioning file
Constructing working set for candidate package set
Constructing working set for running package set
Checking command output
Constructing merge of running and candidate packages
Checking if resulting candidate package set would be complete
Finished candidate package set construction

--- Starting ISSU compatiblity verficiation ---
Verifying image type compatibility
Checking IPC compatibility with running software
Checking candidate package set infrastructure compatibility
Checking infrastructure compatibility with running software
Checking package specific compatibility
Finished ISSU compatiblity verficiation

--- Starting impact testing ---
Checking operational impact of change
Finished impact testing

--- Starting list of software package changes ---
Old files list:
  Removed cbrsup-clcdocsis.BLD_V155_2_S_XE315_THROTTLE_LATEST_20150217_110041-st
d.SSA.pkg
New files list:
  Added cbrsup-clcdocsis.2015-02-20_01.02.SSA.pkg
Finished list of software package changes

--- Starting commit of software changes ---
Updating provisioning rollback files
Creating pending provisioning file
Committing provisioning file
Finished commit of software changes

--- Starting analysis of software changes ---
Finished analysis of software changes

--- Starting update running software ---
Blocking peer synchronization of operating information
Creating the command set placeholder directory
  Finding latest command set
  Finding latest command shortlist lookup file
  Finding latest command shortlist file
  Assembling CLI output libraries
  Assembling CLI input libraries
```

```
  Assembling Dynamic configuration files
  Applying interim IPC and database definitions
  Replacing running software
  Replacing CLI software
  Restarting software
  Restarting software: target frus filtered out ... skipped
  Applying final IPC and database definitions
  Generating software version information
  Notifying running software of updates
  Unblocking peer synchronization of operating information
Unmounting old packages
Cleaning temporary installation files
  Finished update running software

SUCCESS: Finished installing software.
Found clc package
Found clc package
Found clcdocsis package
SUCCESS: Reload Cable Linecard at slot 1
SUCCESS: Reload Cable Linecard at slot 2
SUCCESS: node ISSU finished successfully.
Invoking cleanup routine
```

Use the **request platform software package install node file** *filename* **noreload linecard** command to upgrade a line card subpackage.

```
Router#request platform software package install node file bootflash:/subpkg/cbr_patch.5.0.tar
 noreload linecard
NOTE: Currently node has booted from a provisioning file
NOTE: Going to start a dual rp sub-packages node ISSU install

--- Starting initial file path checking ---
Copying bootflash:/subpkg/cbr_patch.5.0.tar to stby-bootflash:/subpkg/cbr_patch.
5.0.tar
Finished initial file path checking

--- Starting config-register verfication ---
Finished config-register verfication

--- Starting Checking noreload options ---
Finished Checking noreload options

--- Starting image file expansion ---
Expanding image file: bootflash:/subpkg/cbr_patch.5.0.tar
Image file expanded and copied
Expanding image file: stby-bootflash:/subpkg/cbr_patch.5.0.tar
Image file expanded and copied
Finished image file expansion

Found clc package

STAGE 1: Installing software on standby RP
==========================================
--- Starting local lock acquisition on R0 ---
Finished local lock acquisition on R0

--- Starting installation state synchronization ---
Finished installation state synchronization

--- Starting local lock acquisition on R1 ---
Finished local lock acquisition on R1
```

```
--- Starting file path checking ---
Finished file path checking

--- Starting image file verification ---
Checking image file names
Locating image files and validating name syntax
  Found cbrsup-clcdocsis.2015-03-01_01.40.SSA.pkg
Verifying image file locations
Inspecting image file types
Processing image file constraints
Creating candidate provisioning file
Finished image file verification

--- Starting candidate package set construction ---
Verifying existing software set
Processing candidate provisioning file
Constructing working set for candidate package set
Constructing working set for running package set
Checking command output
Constructing merge of running and candidate packages
Checking if resulting candidate package set would be complete
Finished candidate package set construction

--- Starting ISSU compatiblity verficiation ---
Verifying image type compatibility
Checking IPC compatibility with running software
Checking candidate package set infrastructure compatibility
Checking infrastructure compatibility with running software
Checking package specific compatibility
Finished ISSU compatiblity verficiation

--- Starting list of software package changes ---
Old files list:
  Removed cbrsup-clcdocsis.2015-03-01_03.43.SSA.pkg
New files list:
  Added cbrsup-clcdocsis.2015-03-01_01.40.SSA.pkg
Finished list of software package changes

--- Starting commit of software changes ---
Updating provisioning rollback files
Creating pending provisioning file
Committing provisioning file
Finished commit of software changes

--- Starting analysis of software changes ---
Finished analysis of software changes

--- Starting update running software ---
Blocking peer synchronization of operating information
Creating the command set placeholder directory
  Finding latest command set
  Finding latest command shortlist lookup file
  Finding latest command shortlist file
  Assembling CLI output libraries
  Assembling CLI input libraries
  Assembling Dynamic configuration files
  Applying interim IPC and database definitions
  Replacing running software
  Replacing CLI software
  Restarting software
  Restarting software: target frus filtered out ... skipped
  Applying final IPC and database definitions
  Generating software version information
```

```
  Notifying running software of updates
  Unblocking peer synchronization of operating information
Unmounting old packages
Cleaning temporary installation files
  Finished update running software

SUCCESS: Finished installing software.

STAGE 2: Installing software on active RP
========================================
--- Starting local lock acquisition on R0 ---
Finished local lock acquisition on R0

--- Starting installation state synchronization ---
Finished installation state synchronization

--- Starting file path checking ---
Finished file path checking

--- Starting image file verification ---
Checking image file names
Locating image files and validating name syntax
  Found cbrsup-clcdocsis.2015-03-01_01.40.SSA.pkg
Verifying image file locations
Inspecting image file types
Processing image file constraints
Creating candidate provisioning file
Finished image file verification

--- Starting candidate package set construction ---
Verifying existing software set
Processing candidate provisioning file
Constructing working set for candidate package set
Constructing working set for running package set
Checking command output
Constructing merge of running and candidate packages
Checking if resulting candidate package set would be complete
Finished candidate package set construction

--- Starting ISSU compatiblity verficiation ---
Verifying image type compatibility
Checking IPC compatibility with running software
Checking candidate package set infrastructure compatibility
Checking infrastructure compatibility with running software
Checking package specific compatibility
Finished ISSU compatiblity verficiation

--- Starting impact testing ---
Checking operational impact of change
Finished impact testing

--- Starting list of software package changes ---
Old files list:
  Removed cbrsup-clcdocsis.2015-03-01_03.43.SSA.pkg
New files list:
  Added cbrsup-clcdocsis.2015-03-01_01.40.SSA.pkg
Finished list of software package changes

--- Starting commit of software changes ---
Updating provisioning rollback files
Creating pending provisioning file
Committing provisioning file
Finished commit of software changes
```

```
--- Starting analysis of software changes ---
Finished analysis of software changes

--- Starting update running software ---
Blocking peer synchronization of operating information
Creating the command set placeholder directory
  Finding latest command set
  Finding latest command shortlist lookup file
  Finding latest command shortlist file
  Assembling CLI output libraries
  Assembling CLI input libraries
  Assembling Dynamic configuration files
  Applying interim IPC and database definitions


  Replacing running software
  Replacing CLI software
  Restarting software
  Restarting software: target frus filtered out ... skipped
  Applying final IPC and database definitions
  Generating software version information
  Notifying running software of updates
  Unblocking peer synchronization of operating information
Unmounting old packages
Cleaning temporary installation files
  Finished update running software

SUCCESS: Finished installing software.
Found clc package
SUCCESS: node ISSU finished successfully.
Invoking cleanup routine
```

Use the **show platform software patch** *n* **info** command to verify completion of this upgrade.

```
Router#show platform software patch 1 info
cbrsup-clciosdb: 3.15 (0.0)
cbrsup-clc-firmware: 3.15 (0.0)
cbrsup-clcvideo: 3.15 (0.0)
cbrsup-clcios: 3.15 (0.0)
cbrsup-clccontrol: 3.15 (0.0)
cbrsup-clcdocsis: 3.15 (1.0)
cbrsup-clcmipsbase: 3.15 (0.0)
```

```
Router#show platform software patch 2 info
cbrsup-clciosdb: 3.15 (0.0)
cbrsup-clc-firmware: 3.15 (0.0)
cbrsup-clcvideo: 3.15 (0.0)
cbrsup-clcios: 3.15 (0.0)
cbrsup-clccontrol: 3.15 (0.0)
cbrsup-clcdocsis: 3.15 (1.0)
cbrsup-clcmipsbase: 3.15 (0.0)
```

Use the **show platform software ios** *slot-number* **restart info** command to verify completion of this upgrade. This example shows the output of this **show** command for the RF line card slot number 2.

```
Router#show platform software ios 2 restart info
IOSD process restart info:
  Process restartable: Yes
```

```
  IOSD restart state : NOT_RESTARTED_YET
  Total Modem Count  : 251
  Active Modem Count : 251

Router#
```

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Consolidated Packages and SubPackages Management

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

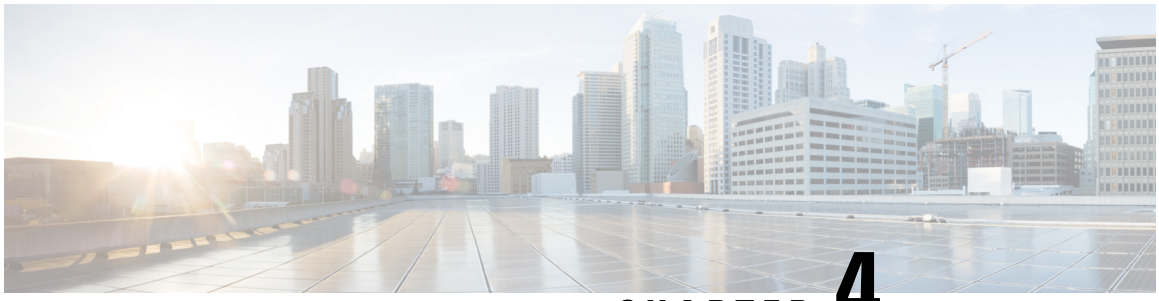
Table 4: Feature Information for Consolidated Packages and SubPackages Management

Feature Name	Releases	Feature Information
Consolidated Packages and SubPackages Management	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Routers.

PART II

High Availability Configuration

- Cisco IOS-XE In-Service Software Upgrade Process, on page 91
- Supervisor Redundancy, on page 97
- Line Card Redundancy, on page 113



CHAPTER 4

Cisco IOS-XE In-Service Software Upgrade Process

Cisco cBR-8 Routers support the In-Service Software Upgrades (ISSU) for redundant platforms.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 91
- Information about In-Service Software Upgrade, on page 92
- How to Configure In-Service Software Upgrade, on page 92
- Additional References, on page 94
- Feature Information for In-Service Software Upgrade, on page 95

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 5: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information about In-Service Software Upgrade

Cisco cBR-8 Routers support the In-Service Software Upgrades (ISSU) for redundant platforms. The ISSU process allows software to be updated or otherwise modified while packet forwarding continues with the benefit of LCHA.

For the Cisco cBR Series Routers, ISSU-compatibility depends on the software subpackage being upgraded and the hardware configuration.

The specific procedures in this document represent supported and tested installation sequences. The Cisco IOS-XE system software allows other installation sequences for special purposes under the guidance of Cisco customer support representatives, but the steps in this document should be followed otherwise. These steps should be followed completely, as the Cisco cBR Series Routers are designed to run one version of Cisco IOS-XE on an SUP, and running subpackages from different versions of Cisco IOS-XE can cause unexpected router behavior.

How to Configure In-Service Software Upgrade

This section describes the configuration of the ISSU feature:

Configuring Subpackage Upgrade

Subpackages Upgrade

Subpackage upgrade allows a subset of the running software to be upgraded. It is intended for patching small and targeted fix instead of full image upgrade. Subpackage upgrade supports both single and dual SUP setup.

Single SUP Subpackages Upgrade

Before you begin

Be sure to complete the following prerequisites before running the ISSU process:

- Config register autoboot enabled.
- Target patch copied to active SUP in the same directory of the packages.conf file system is booted up with.
- If needed copy patch info file to SUP.
- Enough bootflash disk space on SUP.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**request platform software package install rp** *rp-slot* **file bootflash:** **Example:** `Router# request platform software package install rp 1 file bootflash:cbrsup-universalk9.03.17.00.S.156-1.S-std.SPA.bin`	Upgrades the cBR-8 router with one SUP using subpackages ISSU procedure.

Dual SUPs Subpackages Upgrade

Before you begin

Be sure to complete the following prerequisites before running the ISSU process:

- Standby SUP must be in hot standby.
- Config register autoboot enabled.
- Both SUP in sub-package mode, running same base image and patches from same path.
- Target patch copied to active SUP in the same directory of the packages.conf file system is booted up with.
- If needed copy patch info file to both SUPs.
- Enough bootflash disk space on both SUPs.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**request platform software package install node file bootflash:** **Example:** `Router# request platform software package install node file bootflash:cbrsup-universalk9.03.17.00.S.156-1.S-std.SPA.bin`	Upgrades the cBR-8 router with dual SUPs using subpackages ISSU procedure.

Linecard Only In-Service Software Upgrade

Use **request platform software package install node linecard-only** command to upgrade only the linecard to the same version as the one in the current active SUP, the customer can choose to upgrade one linecard or all the linecards in the chassis.

This command can be used together with **request platform software package install node file** *file-path* **noreload linecard** command to upgrade SUP first, and then upgrade linecard.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**request platform software package install node linecard-only** **Example:** `Router# request platform software package install node linecard-only all`	Upgrade all the linecards to the same version as the one in the current active SUP.

Additional References

The following sections provide references related to the ISSU feature.

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

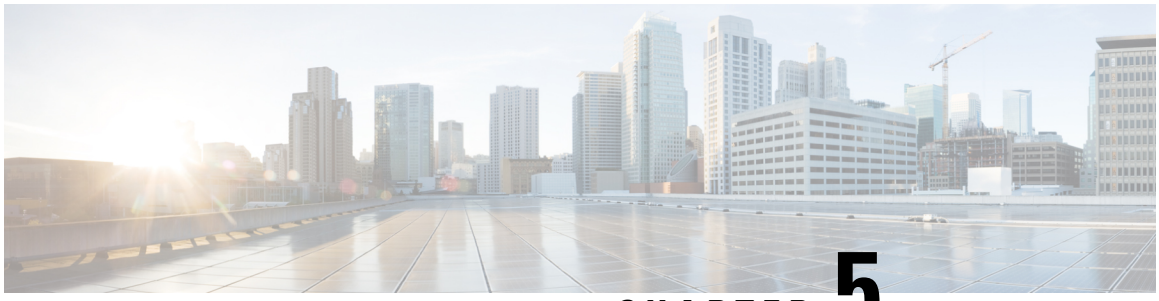
Feature Information for In-Service Software Upgrade

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 6: Feature Information for ISSU

Feature Name	Releases	Feature Information
ISSU	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the cisco cBR Series Converged Broadband Routers.



CHAPTER 5

Supervisor Redundancy

Supervisor redundancy reduces unplanned downtime. It enables a quicker switchover between active and standby Supervisor s in the event of a fatal error on the active Supervisor . When you configure Supervisor redundancy, the standby Supervisor is synchronized with the active Supervisor . In the event of a fatal error on the active Supervisor , the system immediately switches to the standby Supervisor .

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 97
- Prerequisites for Supervisor Redundancy, on page 98
- Information About Supervisor Redundancy, on page 98
- How to Configure Supervisor Redundancy, on page 102
- Verifying the Supervisor Redundancy Configuration, on page 107
- Configuration Example for Supervisor Redundancy, on page 110
- Additional References, on page 110
- Feature Information for Supervisor Redundancy, on page 110

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 7: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Supervisor Redundancy

- Two Supervisor s (that is, two Supervisor Card s and two Supervisor PIC s) must be installed in the Cisco cBR chassis.
- Both Supervisor s must be running identical software releases.

Information About Supervisor Redundancy

The Supervisor redundancy feature enables the Cisco cBR router to use two Supervisor s in a redundant configuration, so that if the active Supervisor fails or becomes inactive, the system automatically performs a switchover , where the standby Supervisor takes over and assumes full responsibility for systems operations.

The Supervisor redundancy feature does not require a full reboot of the system to perform a switchover. When the system boots up, the standby Supervisor performs full initialization, which includes self initialization, running configuration synchronization from the active Supervisor , and SSO feature data synchronization from the active Supervisor , then it enters into hot standby state and monitors the active Supervisor . If the standby Supervisor detects a failure in the active Supervisor , it can quickly assume the active responsibility for systems operations.

Each Supervisor contains all the resources required to operate the router, such as bootflash memory, hard disks, Ethernet ports, and console port. In the default operation, the standby Supervisor also synchronizes the major systems files, such as the running configuration file, so that during a switchover, the standby Supervisor can duplicate the active Supervisor 's configuration.

You can use Cisco IOS CLI commands to access the standby Supervisor resources, such as the bootflash and hard disk. For example, you can use the **dir** command to list the contents of a device, or use the **copy** command to transfer files between the active and standby Supervisor .

Switchover Procedure

A switchover occurs when the standby Supervisor takes over responsibilities from the active Supervisor . The switchover can occur automatically if the standby Supervisor has determined that the active Supervisor has failed, or an operator can initiate a manual switchover whenever desired.

A switchover triggers the following events:

1. If this is a manual switchover, the active Supervisor verifies that the standby Supervisor is present and has entered into SSO. If so, it instructs the standby Supervisor to begin switchover procedures, and the active Supervisor either attempts to reload its configured Cisco IOS software image or enters ROM monitor mode, depending on the setting of its configuration register.
2. The standby Supervisor assumes responsibility as the active Supervisor and brings the Cisco cBR chassis into active state, and continues the service as active Supervisor .
3. The new active Supervisor begins normal systems operations, including passing traffic.

Note The Supervisor does not begin functioning as a standby Supervisor until it is booted up with a proper Cisco IOS software.

Is Supervisor Switchover Failing?

The usual phenomenon for a Supervisor switchover to be affected is when the active Supervisor has these issues:

- Supervisor hangs
- Login to Supervisor console or Telnet to chassis fails
- Interface cards unable to connect to active Supervisor , hence crashing
- Cable modems drop offline
- Chassis reload required
- Reset of active Supervisor required to restore service

Note In case there is hardware issue with the Supervisor , do not reinsert the faulty Supervisor in the chassis. Inserting a faulty Supervisor (although a standby Supervisor) may cause the interface card to switch to the faulty Supervisor causing the interface card to crash and cable modems to go offline.

Using Redundant File Systems

Both the active and standby Supervisor s have active file systems that can be accessed to store and transfer files. The table below lists the available file systems, the filenames that you can use with CLI commands to access the file systems,and a short description of each.

File System	File Name for CLI Commands	Description
• Bootflash • Flash • Hard disk • USB • Standby bootflash • Standby hard disk • Standby USB	• bootflash: • flash: • harddisk: • usb0: • usb1: • stby-bootflash: • stby-harddisk: • stby-usb0: • stby-usb1:	Stores image, crash file, core files, saved configuration files, and various user files.
• System • Temporary system • Null • Tar • Syslog • CNS • RCSF	• system: • tmpsys: • null: • tar: • syslog: • cns: • revrcsf:	Stores the running configuration and other system files.
• NVRAM • Standby NVRAM • Standby RCSF	• nvram: • stby-nvram: • stby-rcsf:	Typically stores the system default configuration file and startup configuration file.
• TFTP • RCP • PRAM • FTP • HTTP • SCP • HTTPS	• tftp: • rcp: • pram: • ftp: • http: • scp: • https:	Protocols used to transfer files to and from remote devices.

You can use the privileged EXEC commands **dir**, **del**, and **copy** to manage the contents of the file systems. You can also use the commands **mkdir** and **rmdir** to create and remove directories on bootflash or hard disks.

Following is a sample output of the **show file systems** command on the Cisco cBR router:

```
Router# show file systems

File Systems:

Size(b)       Free(b)      Type  Flags  Prefixes
             -             -    opaque     rw   system:
             -             -    opaque     rw   tmpsys:
*   7800705024    1574408192      disk     rw   bootflash:
    7800705024    1574408192      disk     rw   flash:
   98394218496   79534682112      disk     rw   harddisk:
    8009056256    8009023488      disk     rw   usb1:
      33554432      33507452     nvram     rw   stby-nvram:
             -             -    opaque     rw   null:
             -             -    opaque     ro   tar:
             -             -   network     rw   tftp:
             -             -    opaque     wo   syslog:
      33554432      33508476     nvram     rw   nvram:
             -             -   network     rw   rcp:
             -             -   network     rw   pram:
             -             -   network     rw   ftp:
             -             -   network     rw   http:
             -             -   network     rw   scp:
             -             -   network     rw   https:
             -             -    opaque     ro   cns:
             -             -     nvram     rw   stby-rcsf:
    7800705024    1635270656      disk     rw   stby-bootflash:
   98394218496   89040576512      disk     rw   stby-harddisk:
             -             -      disk     rw   stby-usb0:
    1000787968     301559808      disk     rw   stby-usb1:
             -             -    opaque     rw   revrcsf:
```

Console Port Usage After Supervisor Switchover

When an active Supervisor fails, and the standby Supervisor becomes the active Supervisor , you must use the console port on the new active Supervisor to give CLI commands and display statistics for the router. The standby Supervisor console is disabled by default and cannot be used to run any CLI commands. Following is an sample output of the standby Supervisor console:

```
Router-stby>
Standby console disabled
Router-stby>
```

To access the console, move the PC or terminal's serial cable to the console port on the other Supervisor , which is now acting as the active Supervisor .

Benefits

- The Supervisor is not a single point of hardware failure. If a permanent hardware failure in the active Supervisor occurs, the standby Supervisor recovers the system, increasing the level of network service and reliability.

- The standby Supervisor can become the active Supervisor without the manual intervention of a system operator. This reduces the recovery time and the need for an instant response from the network administrators.
- The active Supervisor continues to dynamically synchronize the changed configuration and feature data with the standby Supervisor after the system reaches SSO. Therefore, the standby Supervisor always operates as a hot standby and ready to take over.

How to Configure Supervisor Redundancy

The Supervisor redundancy feature is automatically enabled when two Supervisor are installed in the Cisco cBR chassis. The active Supervisor automatically synchronizes the running configuration file with the standby Supervisor during the bootup of standby Supervisor .

Note The Cisco cBR router supports only the SSO mode for Supervisor redundancy. The default redundancy mode is SSO and this mode does not need any new configurations.

This section contains the following:

Forcing Switchover

To manually force a switchover, so that the standby Supervisor becomes active, use the **redundancy force-switchover** command in privileged EXEC mode on the active Supervisor . Manually forcing a switchover is useful in the following situations:

- You need to remove, replace, or upgrade the currently active Supervisor .
- A previous switchover has activated the standby Supervisor and you now want to restore the previously active Supervisor .

Tip Simply removing the active Supervisor also triggers a switchover, but using the **redundancy force-switchover** command does not generate a hardware alarm.

Before you begin

Ensure that the standby Supervisor is in the SSO state using the **show redundancy** command. For more information, see Verifying Supervisor Redundancy , on page 107.

Procedure

Step 1 Set the configuration register as 0x02 and the load the appropriate image on both the Supervisor s

Example:

```
Router# configure terminal
Router(config)# config-register 0x02
```

```
Router(config)# boot system bootflash:cbrsup-universalk9.2015-03-08_01.38_xxxxx.SSA.bin
```

Note Do not perform this step if you want to set the previous active Supervisor to stay in ROM monitor mode or manually boot it up after the switchover.

Step 2 Use the **redundancy force-switchover** command to force the switchover.

Example:

```
Router# redundancy force-switchover

Proceed with switchover to standby RP? [confirm]
Manual Swact = enabled

Jan 1 19:23:22.483 R0/0: %PMAN-5-EXITACTION: Process manager is exiting: process exit with
 reload fru code

Initializing Hardware ...

System Bootstrap, Version 12.2(20141120:061458) [153], DEVELOPMENT SOFTWARE
Copyright (c) 1994-2014 by cisco Systems, Inc.
Compiled Thu 11/20/2014 18:04:24.91 by xxxxx
```

The standby Supervisor becomes the active Supervisor .

Step 3 (Optional) If you have not performed Step 1, on page 102, the previous active Supervisor is put into the ROM monitor mode after the switchover. To enable the previous active Supervisor to become the new standby Supervisor , manually boot up the new standby Supervisor to enter into SSO mode.

Changing the System Boot Behavior

This section describes how to change the Cisco IOS software configuration register to modify how the system behaviors at power-on or reboot. The software configuration register is a 16-bit register in NVRAM that controls the following boot functions:

- Specifies the source of the Cisco IOS software image to be loaded
- Specifies whether the Cisco IOS software should ignore the contents of the saved configuration file in NVRAM memory
- Enables or disables the use of the Break function

Use the following procedure to change the software configuration register settings:

Procedure

Step 1 Enter global configuration mode and use the **config-register** command to set the contents of the software configuration register to a new value. You must specify the new value as a 16-bit hexadecimal bitmask, using the values shown in the table below.

Table 8: Definition of Bits in the Software Configuration Register

Bit No.	Hex Value	Meaning/Function
00 to 03	0x0000 to 0x000F	Defines the source of a default Cisco IOS software image required to run the router: • 00—On powerup, the system remains at the ROM monitor prompt (rommon>), awaiting a user command to boot the system manually by means of the rommon **boot** command. • 01—On powerup, the system automatically boots the first system image found in the Flash memory single inline memory module (SIMM) on the Supervisor . • 02 to 0F—On powerup, the system automatically boots from a default Cisco IOS software image stored on a TFTP server in the network. For this setting, the Network Management Ethernet port on the Supervisor must be configured and operational. This setting also enables boot system commands that override the default filename.
06	0x0040	Causes system software to ignore the contents of the NVRAM configuration file.
07	0x0080	Enables the original equipment manufacturer (OEM) bit.
08	0x0100	Disables the Break function after 30 seconds.
09	0x0200	Not used.
10	0x0400	Specifies that broadcast packets are based on the 0.0.0.0 IP address.
11 and 12	0x0800 to 0x1000	Defines the console baud rate (the default setting is 9600 baud).
13	0x2000	Boots an image from the bootflash memory.
14	0x4000	Specifies that broadcast packets use the subnet broadcast address.
15	0x8000	Enables diagnostic messages and ignores the contents of the NVRAM configuration file.

For example, to configure the router to boot to the ROM monitor prompt, set the configuration register to **0x2100** with the following commands:

Example:

```
Router# config t
Router(config)# config-register 0x2100
Router(config)#
```

Tip The typical bitmask for normal use is 0x2102, which specifies that the router loads the Cisco IOS software from the Flash memory and boots to the Cisco IOS CLI prompt. The Break key is enabled for only 30 seconds, so that the user can break to the ROM monitor prompt if desired.

Step 2 Exit the global configuration mode by entering the **exit** command.

Example:

```
Router(config)# exit
Router#
```

Step 3 Display the new software configuration register setting using the **show version** command. The last line shows the settings of the configuration register:

Example:

```
Router# show version
Cisco IOS XE Software, Version 2015-03-04_00.38_xxxxx
Cisco IOS Software, IOS-XE Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Experimental  \
Version 15.5(20150302:044048) [v155_2_s_xe315_throttle-xxxxx-XE315_0301 121]
This software is an Engineering Special
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Wed 04-Mar-15 00:21 by xxxxx


Cisco IOS-XE software, Copyright (c) 2005-2015 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0.  The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.  For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.


ROM: IOS-XE ROMMON

Router uptime is 14 minutes
Uptime for this control processor is 17 minutes
System returned to ROM by SSO Switchover
System image file is "bootflash:cbrsup-universalk9.2015-03-04_00.38_xxxxx.SSA.bin"
Last reload reason: Reload Command



This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

cisco cBR1013 (CBR) processor (revision CBR) with 3647635K/6147K bytes of memory.
Processor board ID CSJ13152101
16 Gigabit Ethernet interfaces
32768K bytes of non-volatile configuration memory.
50331648K bytes of physical memory.
7739391K bytes of eUSB flash at bootflash:.
97620247K bytes of SATA hard disk at harddisk:.
979258K bytes of USB flash at usb1:.
```

```
Configuration register is 0x2
```

When you change the configuration register, the **show version** command shows both the current value of the register, as well as the value that will be used on the next reboot or reload.

Step 4 Perform one of the following to save the configuration file to preserve the new software configuration register settings:

- Use the **copy running-config startup-config** command.
- Use the **write** command.

Example:

```
Router# copy running-config startup-config
```

```
Router# write
Building configuration...
[OK]
```

Step 5 The changes to the software configuration register will take effect the next time the router is rebooted or restarted. To manually reboot the router, use the **reload** command:

Example:

```
Router# reload
System configuration has been modified. Save? [yes/no]: yes
Proceed with reload? [confirm]
```

Saving a Configuration File to the Bootflash or Hard Disk

This section describes how to copy a configuration file to a bootflash or hard disk and configure the Cisco cBR router.

Procedure

Step 1 Copy the configuration file to the bootflash or hard disks in both Supervisor s.

Example:

```
Router# copy running-config bootflash:cbr8-config
Router# copy running-config stby-bootflash:cbr8-config
Router# copy running-config harddisk:cbr8-config
Router# copy running-config stby-harddisk:cbr8-config
```

Step 2 If the configuration file is currently on a TFTP server, copy the file from the TFTP server to the bootflash or hard disk in each Supervisor .

Example:

```
Router# copy tftp://192.168.100.10/router-config bootflash:cbr8-config
Router# copy tftp://192.168.100.10/router-config stby-bootflash:cbr8-config
Router# copy tftp://192.168.100.10/router-config harddisk:cbr8-config
```

```
Router# copy tftp://192.168.100.10/router-config stby-harddisk:cbr8-config
```

Verifying the Supervisor Redundancy Configuration

This section contains the following:

Verifying Supervisor Redundancy

Procedure

Step 1 Display the startup configuration and verify that the lines configuring redundancy appear:

Example:

```
Router# show startup-config

...
redundancy
mode sso
...
```

Step 2 Display the current Supervisor redundancy state using the **show redundancy** command. The active Supervisor typically is shown in slot 4 (SUP0):

```
Router# show redundancy

Redundant System Information :
------------------------------
Available system uptime = 28 minutes
Switchovers system experienced = 0
Standby failures = 0
Last switchover reason = none

Hardware Mode = Duplex
Configured Redundancy Mode = sso
Operating Redundancy Mode = sso
Maintenance Mode = Disabled
Communications = Up

Current Processor Information :
-------------------------------
Active Location = slot 4
Current Software state = ACTIVE
Uptime in current state = 28 minutes
Image Version = Cisco IOS Software, IOS-XE Software (X86_64_LINUX_IOSD-UNIVERSALK9-M),
Experimental Version 15.5(20150302:044048) [v155_2_s_xe315_throttle-xxxxx-XE315_0301 121]
This software is an Engineering Special
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Wed 04-Mar-15 00:21 by xxxxx
BOOT = bootflash:cbrsup-universalk9.2015-03-04_00.38_xxxxx.SSA.bin,12;
CONFIG_FILE = bootflash:startup_config1419513118
Configuration register = 0x2
```

```
Peer Processor Information :
----------------------------
Standby Location = slot 5
Current Software state = STANDBY HOT
Uptime in current state = 24 minutes
Image Version = Cisco IOS Software, IOS-XE Software (X86_64_LINUX_IOSD-UNIVERSALK9-M),
Experimental Version 15.5(20150302:044048) [v155_2_s_xe315_throttle-xxxxx-XE315_0301 121]
This software is an Engineering Special
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Wed 04-Mar-15 00:21 by xxxxx
BOOT = bootflash:cbrsup-universalk9.2015-03-04_00.38_xxxxx.SSA.bin,12;
CONFIG_FILE = bootflash:startup_config1419513118
Configuration register = 0x2
```

If a switchover has occurred, the **show redundancy** command will produce a display similar to the following, showing that the active Supervisor has changed slots that is, moving from slot 4 (SUP0) to slot 5 (SUP1).

```
Router# show redundancy

Redundant System Information :
------------------------------
Available system uptime = 39 minutes
Switchovers system experienced = 1
Standby failures = 0
Last switchover reason = user forced

Hardware Mode = Duplex
Configured Redundancy Mode = sso
Operating Redundancy Mode = sso
Maintenance Mode = Disabled
Communications = Up

Current Processor Information :
-------------------------------
Active Location = slot 5
Current Software state = ACTIVE
Uptime in current state = 10 minutes
Image Version = Cisco IOS Software, IOS-XE Software (X86_64_LINUX_IOSD-UNIVERSALK9-M),
Experimental Version 15.5(20150302:044048) [v155_2_s_xe315_throttle-xxxxx-XE315_0301 121]
This software is an Engineering Special
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Wed 04-Mar-15 00:21 by xxxxx
BOOT = bootflash:cbrsup-universalk9.2015-03-04_00.38_xxxxx.SSA.bin,12;
CONFIG_FILE = bootflash:startup_config1419513118
Configuration register = 0x2

Peer Processor Information :
----------------------------
Standby Location = slot 4
Current Software state = STANDBY HOT
Uptime in current state = 4 minutes
Image Version = Cisco IOS Software, IOS-XE Software (X86_64_LINUX_IOSD-UNIVERSALK9-M),
Experimental Version 15.5(20150302:044048) [v155_2_s_xe315_throttle-xxxxx-XE315_0301 121]
This software is an Engineering Special
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Wed 04-Mar-15 00:21 by xxxxx
BOOT = bootflash:cbrsup-universalk9.2015-03-04_00.38_xxxxx.SSA.bin,12;
CONFIG_FILE = bootflash:startup_config1419513118
Configuration register = 0x2
```

If the standby Supervisor is not installed or is not operational, the **show redundancy** command will produce a display similar to the following:

```
Router# show redundancy

Redundant System Information :
------------------------------
Available system uptime = 31 minutes
Switchovers system experienced = 1
Standby failures = 0
Last switchover reason = user forced

Hardware Mode = Simplex
Configured Redundancy Mode = sso
Operating Redundancy Mode = Non-redundant
Maintenance Mode = Disabled
Communications = Down Reason: Failure

Current Processor Information :
-------------------------------
Active Location = slot 5
Current Software state = ACTIVE
Uptime in current state = 2 minutes
Image Version = Cisco IOS Software, IOS-XE Software (X86_64_LINUX_IOSD-UNIVERSALK9-M),
Experimental Version 15.5(20150302:044048) [v155_2_s_xe315_throttle-xxxxx-XE315_0301 121]
This software is an Engineering Special
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Wed 04-Mar-15 00:21 by xxxxx
BOOT = bootflash:cbrsup-universalk9.2015-03-04_00.38_xxxxx.SSA.bin,12;
CONFIG_FILE = bootflash:startup_config1419513118
Configuration register = 0x2

Peer (slot: 4) information is not available because it is in 'DISABLED' state
```

Note The **show redundancy** command shows the redundancy state, software state, system uptime, image version, boot, configuration file, and configuration register information.

Verifying Supervisor Switchover

Procedure

Step 1 Verify the LEDs on the Supervisor Card . When a Supervisor becomes active, the RP ACT and FP ACT LEDs on the Supervisor Card illuminate green to indicate that they have initialized and acting as the active Supervisor . The RP ACT and FP ACT on standby Supervisor Card are off. For more information, see Monitoring the Supervisor in the Cisco cBR Chassis.

Step 2 To verify that a switchover has occurred, use the **show redundancy switchover history** command. Assuming that the original Supervisor had been in slot 4 (SUP0), and that the standby Supervisor is in slot 5 (SUP1), the following is the sample output:

Example:

```
Router# show redundancy switchover history

Index Previous Current Switchover Switchover
      active   active  reason     time
----- -------- ------- ---------- ----------
```

```
1      48       49      user forced 19:23:11 CST Sun Jan 1 2012
```

where, 48 indicates SUP0 and 49 indicates SUP1.

Configuration Example for Supervisor Redundancy

The following example shows the relevant portion of the Cisco IOS configuration file for the default configuration for the Supervisor redundancy feature, which should be used for most applications:

```
Router# show running-config | sec redundancy

redundancy
 mode sso

Router#
```

Additional References

Related Documents

Related Topic	Document Title
CMTS commands	Cisco IOS CMTS Cable Command Reference
Stateful Switchover	Stateful Switchover

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Supervisor Redundancy

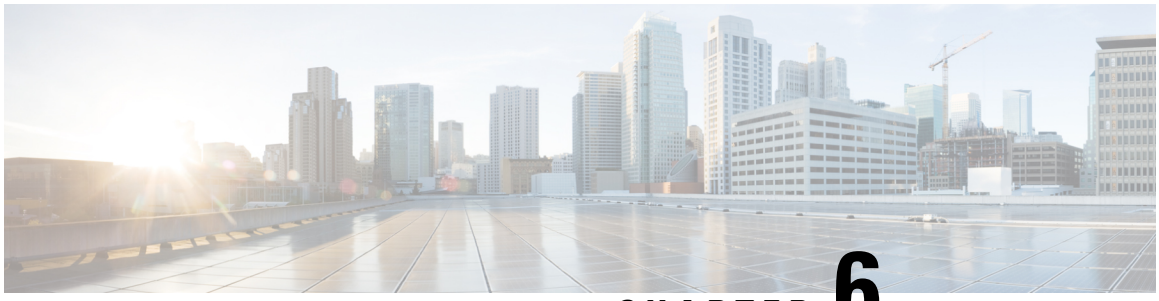
Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release,

feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 9: Feature Information for Supervisor Redundancy

Feature Name	Releases	Feature Information
Supervisor Redundancy	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 6

Line Card Redundancy

The line cards support high availability with redundancy schemes. Line card redundancy can help limit customer premises equipment (CPE) downtime by enabling robust automatic switchover and recovery in the event that there is a localized system failure.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 113
- Prerequisites for Line Card Redundancy, on page 114
- Restrictions for Line Card Redundancy, on page 114
- Information About Line Card Redundancy, on page 115
- How to Configure Line Card Redundancy, on page 116
- Verifying the Line Card Redundancy Configuration, on page 117
- Additional References, on page 121
- Feature Information for Line Card Redundancy, on page 122

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 10: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Line Card Redundancy

- At least one RF Through PIC and its corresponding interface line card must be installed in the chassis to be configured as the primary card.
- An RF Protect PIC and its corresponding interface line card must be installed in the chassis to be configured as the secondary card.

Restrictions for Line Card Redundancy

- The line cards installed in slot 3 and 6 of the Cisco cBR-8 router cannot be configured as the secondary card.
- The RF Protect PIC can send RF signals only to the lower slots (with larger slot number). So, the slot number of the secondary card must be the smallest in the redundancy group.

Note We recommend that you install the RF Protect PIC in the uppermost slot (slot 0) of the chassis and configure it as the secondary card.

- The RF Through PIC can send RF signal only from upper slot to lower slot. So, do not install any RF blank PICs between the secondary card and primary cards.
- You cannot change any configuration on the primary or secondary card when the secondary card is active.
- You cannot remove the last primary card if there is a secondary card in the redundancy group. You must remove the secondary card and then remove the primary card.
- If the primary card is in the standby role, you must revert to the primary card before removing it from the redundancy group.

Information About Line Card Redundancy

Line card redundancy reduces the unplanned downtime. When you configure line card redundancy, a protect zone (redundancy group) is created on the router and the configurations on the primary cards are synchronized with the secondary card.

The following events can trigger a switchover from an active card to a standby card:

- Manual switchover using the **redundancy linecard-group switchover from slot** *slot*command.
- Line card reload using the **hw-module slot reload** command.
- Line card crash.
- Line card Online Insertion and Removal (OIR).

The secondary card reloads after the switchover. The router can be configured to automatically revert to the primary card when it becomes hot standby after an unplanned switchover triggered by the line card OIR or crash.

Following are the line card redundancy states:

- **Unavail**—The line card state is not available.
- **Init**—The line card did not boot up.
- **Active Cold**—The active card is downloading the configuration.
- **Active**—The active card is fully configured and working.
- **Stdby Cold**—The standby card configuration is synchronizing with the active card.
- **Stdby Warm**—(Only for the secondary card) The standby card is fully synchronized and ready for switchover. It is the stable state of a secondary standby card.
- **Stdby Hot**—The primary standby card is fully synchronized. It is the stable state of a primary standby card. The secondary standby card is chosen to switchover for a primary card, and will be active soon. It is a transient state when secondary card is becoming active.

N+1 Line Card Redundancy

The Cisco cBR-8 router supports N+1 redundancy scheme for line cards. A single RF Protect PIC can be configured as a secondary card for multiple RF Through PICs (primary cards). In this redundancy scheme, when the secondary card becomes the active card for a primary card, the redundancy scheme is changed to 1+1 redundancy.

The Cisco cBR-8 router supports a single protect zone or redundancy group (group 0).

How to Configure Line Card Redundancy

This section contains the following:

Configuring Line Card Manual Switchover

Before you begin

The line card must be in active role, and warm standby or hot standby state. Use the **show redundancy linecard all** command to verify the role and state of the card.

Restrictions

- You cannot perform a manual switchover when the standby Supervisor is booting up and not yet entered into SSO.
- You cannot auto revert the switchover triggered manually.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**redundancy linecard-group switchover from slot** *slot* **Example:** `Router# redundancy linecard-group switchover from slot 9`	Manually switches over from the active line card.

Configuring N+1 Line Card Redundancy

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**redundancy** **Example:** `Router(config)# redundancy`	Enables redundancy and enters redundancy configuration mode.
Step 4	**linecard-group** *group-id* **internal-switch** **Example:** `Router(config-red)# linecard-group 0 internal-switch`	Configures the redundancy group and enters the line card redundancy configuration mode.
Step 5	**description** *group-description* **Example:** `Router(config-red-lc)# description RedundancyGroup0`	(Optional) Configures the redundancy group description.
Step 6	**class 1:N** **Example:** `Router(config-red-lc)# class 1:N`	Configures the N+1 redundancy class for the redundancy group.
Step 7	**revertive** *seconds* **Example:** `Router(config-red-lc)# revertive 60`	(Optional) Configures the auto revert time for the primary card, in seconds.
Step 8	**member slot** *slot* **primary** **Example:** `Router(config-red-lc)# member slot 1 primary`	Adds the line card as a primary card in the redundancy group. **Note** Repeat this step for each primary card to be added in the redundancy group.
Step 9	**member slot** *slot* **secondary** **Example:** `Router(config-red-lc)# member slot 0 secondary`	Adds the line card as a primary card in the redundancy group.
Step 10	**end** **Example:** `Router(config-red-lc)# end`	Returns to privileged EXEC mode.

Verifying the Line Card Redundancy Configuration

- **show redundancy linecard group all**—Displays the redundancy group information.

  The following is a sample output of this command:

  ```
  Router# show redundancy linecard group all
  ```

```
Group Identifier: 0
Revertive, Revert Timer: OFF (60000 sec)
Reserved Cardtype: 0xFFFFFFFF  4294967295
Group Redundancy Type: INTERNAL SWITCH
Group Redundancy Class: 1:N
Group Redundancy Configuration Type: LINECARD GROUP
Primary: Slot 6
Primary: Slot 7
Secondary: Slot 0
```

- **show redundancy linecard all**—Displays the role and state information for all line cards.

  Following is a sample output of this command:

```
Router# show redundancy linecard all

              LC     My         Peer       Peer     Peer
Slot Subslot  Group  State      State      Slot     Subslot  Role     Mode
-------------------------------------------------------------------------------
 9   -        0      Active     Stdby Cold 0        -        Active   Primary
 8   -        0      Active     Stdby Warm 0        -        Active   Primary
 7   -        0      Active     Stdby Warm 0        -        Active   Primary
 6   -        0      Active     Stdby Cold 0        -        Active   Primary
 3   -        0      Active     Stdby Cold 0        -        Active   Primary
 2   -        0      Active     Stdby Cold 0        -        Active   Primary
 1   -        0      Active     Stdby Cold 0        -        Active   Primary
 0   -        0      -          -          Multiple None     Standby  Secondary
```

  Note The secondary card does not have a valid *My State* when it is in *Standby* role as it is the peer for *N* primary cards. The secondary card has *N* peer states. For example, it can be cold standby for some primary cards and warm standby for the other primary card.

  Following is a sample output of the command when secondary card becomes active for a primary card, and the N+1 redundancy is changed to 1+1 redundancy:

```
Router# show redundancy linecard all

              LC     My         Peer       Peer     Peer
Slot Subslot  Group  State      State      Slot     Subslot  Role     Mode
-------------------------------------------------------------------------------
 9   -        0      Stdby Hot  Active     0        -        Standby  Primary
 8   -        0      Active     Unavail    0        -        Active   Primary
 7   -        0      Active     Unavail    0        -        Active   Primary
 6   -        0      Active     Unavail    0        -        Active   Primary
 3   -        0      Active     Unavail    0        -        Active   Primary
 2   -        0      Active     Unavail    0        -        Active   Primary
 1   -        0      Active     Unavail    0        -        Active   Primary
 0   -        0      Active     Stdby Hot  9        -        Active   Secondary
```

- **show redundancy linecard slot**—Displays the redundancy information for the line card.

  Following is a sample output of the command:

```
Router# show redundancy linecard slot 9

 LC Redundancy Is Configured:
 LC Group Number: 0
```

```
 LC Slot: 9 (idx=9)
 LC Peer Slot: 0
 LC Card Type: 0x4076 , 16502
 LC Name: 9
 LC Mode: Primary
 LC Role: Active
 LC My State: Active
 LC Peer State: Stdby Warm
```

- **show redundancy linecard history**—Displays the state change history for all line cards.

  Following is a sample output of the command:

```
Router# show redundancy linecard history

Jan 05 2012 12:24:27 20559 - st_mem(9): MY State Change, (Active Wait) -> (Active)
Jan 05 2012 12:24:27 20559 - st_mem(9): MY FSM execution, Active Wait:Init:State Ntfy
Jan 05 2012 12:24:27 20559 - st_mem(9): MY State Change, (Active LC Cfg Dnld) -> (Active
 Wait)
Jan 05 2012 12:24:27 20559 - st_mem(9): MY FSM execution, Active LC Cfg Dnld:Init:Cfg
Dnld Done
Jan 05 2012 12:24:27 20559 - st_mem(9): MY State Change, (Active Cold) -> (Active LC
Cfg Dnld)
Jan 05 2012 12:23:09 12763 - st_mem(9): MY FSM execution, Active Cold:Init:Cfg Dnld
Jan 05 2012 12:23:09 12760 - st_mem(9): MY State Change, (Init) -> (Active Cold)
Jan 05 2012 12:23:09 12760 - st_mem(9): MY FSM execution, Init:Init:Up
Jan 05 2012 12:21:39 3746 - st_mem(9): PEER FSM Execution , Init:Init:Reset
```

- **show lcha rfsw**—Displays the internal RF switch PIC state information.

  Following is a sample output of the command:

```
Router# show lcha rfsw

Slot 0 ===================================
Type : Secondary PIC State: normal
Slot 1 ===================================
Type : Primary PIC State: normal
```

- **show lcha logging level**—Displays the cable modem line card logs.

  Following is a sample output of the command:

```
Router# show lcha logging level noise

11:02:03.313 CST Tue Nov 18 2014 [error] [slot=3] [txn=229] Peer-Up Message [tag=1011]
 to slot 3 complete [36144 ms]; status=nak response
11:02:03.313 CST Tue Nov 18 2014 [error] [slot=0] [txn=229] Slot 0 downloaded
configuration for slot 3; result=peer-up notification failed
11:02:03.316 CST Tue Nov 18 2014 [noise] [slot=0] [txn=none]
lcha_plfm_get_max_port_count_for_slot: slot 0 maximum port count is 1794
11:02:03.316 CST Tue Nov 18 2014 [noise] [slot=0] [txn=none]
lcha_plfm_get_starting_port_index: slot 0 starting port count is 0
11:02:03.331 CST Tue Nov 18 2014 [note]  [slot=0] [txn=none] Slot 0 is being reset
11:02:04.352 CST Tue Nov 18 2014 [note]  [slot=0] [txn=none] slot 0 removed
```

- When the secondary card is active, you can use the slot number of either the primary or secondary card in the **show** commands.

  Following is a sample output of the **show interfaces** command after the primary card in slot 8 switches over to secondary card in slot 0:

```
Router# show interfaces c0/0/0

Cable0/0/0 is up, line protocol is up
  Hardware is CMTS MD interface, address is 0000.0000.031e (bia 0000.0000.031e)
  MTU 1500 bytes, BW 26000 Kbit/sec, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation MCNS, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 19500 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 13000 bits/sec, 17 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     140520 packets output, 14052672 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out

Router# show interfaces c8/0/0

Cable0/0/0 is up, line protocol is up
  Hardware is CMTS MD interface, address is 0000.0000.031e (bia 0000.0000.031e)
  MTU 1500 bytes, BW 26000 Kbit/sec, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation MCNS, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 19500 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 14000 bits/sec, 18 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     140616 packets output, 14062272 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
```

- When the secondary card is active, the **show running-config** command displays the output for the secondary card.

Note The output of the **show running-config** command is empty for the primary card when the secondary card is active.

Following is a sample output of the **show running-config** command after the primary card in slot 8 switches over to secondary card in slot 0:

```
Router# show running-config | begin controller Upstream-Cable 0

controller Upstream-Cable 0/0/0
 us-channel 0 channel-width 1600000 1600000
 us-channel 0 docsis-mode atdma
 us-channel 0 minislot-size 4
 us-channel 0 modulation-profile 221
 no us-channel 0 shutdown
 us-channel 1 channel-width 1600000 1600000
 us-channel 1 docsis-mode atdma

Router# show running-config | begin controller Upstream-Cable 8
Router#
Router#
```

Additional References

Related Documents

Related Topic	Document Title
CMTS commands	Cisco CMTS Cable Command Reference

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Line Card Redundancy

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

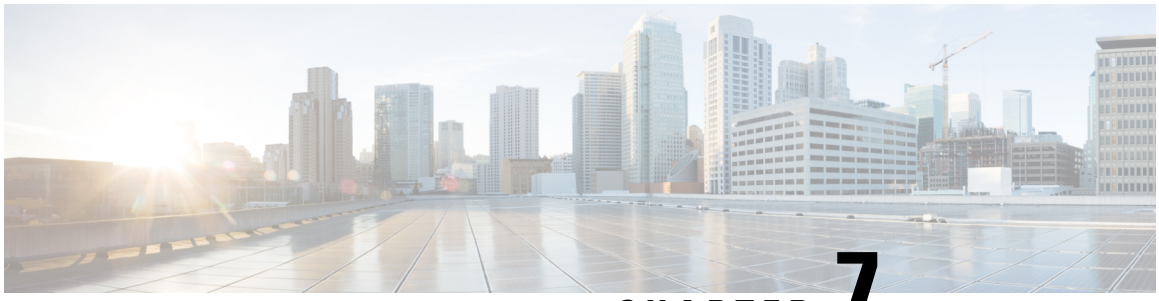
Table 11: Feature Information for Line Card Redundancy

Feature Name	Releases	Feature Information
Line Card Redundancy	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Router s.

PART III

Layer 2 and DOCSIS 3.0 Configuration

- Downstream Interface Configuration, on page 125
- Upstream Interface Configuration, on page 137
- DOCSIS Interface and Fiber Node Configuration, on page 145
- Service Group Based Configuration of the Cisco cBR Router, on page 165
- DOCSIS Load Balancing Groups, on page 173
- DOCSIS Load Balancing Movements, on page 201
- DOCSIS 3.0 Downstream Bonding, on page 239
- DOCSIS 2.0 A-TDMA Modulation Profiles , on page 261
- Downstream Resiliency Bonding Group , on page 279
- Downstream Channel ID Assignment, on page 295
- Upstream Channel Bonding, on page 303
- Spectrum Management and Advanced Spectrum Management, on page 335
- Upstream Scheduler Mode , on page 387
- Generic Routing Encapsulation, on page 393
- Transparent LAN Service over Cable , on page 417
- Downgrading Channel Bonding in Battery Backup Mode, on page 429
- Upstream Bonding Support for D-PON, on page 439
- Energy Management Mode, on page 447
- Cable Modem Steering, on page 459



CHAPTER 7

Downstream Interface Configuration

This document describes how to configure the downstream interfaces on the Cisco cBR Series Converged Broadband Router .

- Finding Feature Information, on page 125
- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 125
- Information About Downstream Interface Configuration , on page 126
- How to Configure Downstream Interfaces, on page 128
- Configuration Examples, on page 132
- Additional References, on page 135
- Feature Information for Downstream Interface Configuration on the Cisco cBR Router, on page 136

Finding Feature Information

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 12: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About Downstream Interface Configuration

Overview

- Each downstream port requires port level configuration and channel level configuration. Port level configuration is optimized with a frequency profile that defines ranges of frequencies available on the port. Channel level configuration is optimized with a QAM profile and channel range configuration block that auto-increments frequency and duplicates annex, modulation, and interleaver.
- Each channel requires a set of parameters: frequency, annex, modulation, interleaver, and DOCSIS channel id.
- Configuration is done in 4 major blocks of configuration:
  - QAM Profile—Example: “cable downstream qam-profile 1”
  - Frequency Profile—Example: “cable downstream freq-profile 2”
  - Port/Controller—Example: “controller Integrated-Cable 3/0/0”
  - RF Channel block—Example: “rf-chan 0 31”

Downstream RF Port and Channel Management

The downstream RF port and channel management feature is responsible for the configuration and management of the downstream RF ports and channels. Each downstream RF channel can be provisioned either as a DOCSIS or traditional MPEG video QAM channel.

QAM Profile

A QAM profile describes the common downstream channel modulator settings, referred to as physical layer parameters. This includes QAM constellation, symbol rate, interleaver-depth, spectrum-inversion, and annex. The QAM profile is described by *CCAP DownPhyParams* object. Default QAM profiles are supported and customized for DOCSIS or MPEG Video, which are described as *DocsisPhyDefault* and *VideoPhyDefault* objects, respectively.

A maximum of 32 QAM profiles can be defined. There are four system-defined QAM profiles (0 to 3), which cannot be deleted or modified. You can define profiles 4 to 31.

The system defined profiles are:

- Profile 0 - default-annex-b-64-qam
  - interleaver-depth: I32-J4
  - symbol rate: 5057 kilo-symbol/second
  - spectrum-inversion: off
- Profile 1 - default-annex-b-256-qam
  - interleaver-depth: I32-J4
  - symbol rate: 5361 kilo-symbol/second
  - spectrum-inversion: off
- Profile 2 - default-annex-a-64-qam
  - interleaver-depth: I12-J17
  - symbol rate: 6952 kilo-symbol/second
  - spectrum-inversion: off
- Profile 3 - default-annex-a-256-qam
  - interleaver-depth: I12-J17
  - symbol rate: 6952 kilo-symbol/second
  - spectrum-inversion: off

Frequency Profile

A frequency profile defines the ranges of frequencies available on a port. A maximum of 16 frequency profiles can be defined. There are four system-defined frequency profiles (0 to 3), which cannot be deleted or modified. You can define profiles 4 to 15.

The system defined profiles are:

- Profile 0 - annex-b-low, Frequency range (Hz): 90000000 - 863999999
- Profile 1 - annex-b-high, Frequency range (Hz): 234000000 - 1002999999
- Profile 2 - annex-a-low, Frequency range (Hz): 94000000 - 867999999
- Profile 3 - annex-a-high, Frequency range (Hz): 267000000 - 1002999999

The frequency ranges are defined using lanes and blocks:

- Four lanes per port, each lane can support 216 MHz range.
- Four blocks per lane, each block can support 54 MHz range.
- Lanes and blocks may have overlapping frequency ranges.

How to Configure Downstream Interfaces

This section contains the following:

Configuring the Cisco CMTS Manually Using Configuration Mode

Connect a console terminal to the console port on the I/O controller. When asked if you want to enter the initial dialog, answer **no** to go into the normal operating mode of the router. After a few seconds the user EXEC prompt (**Router>**) appears.

Configuring the QAM Profile on the Downstream Channels

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable downstream qam-profile** *Qam_Profile_ID* **Example:** `Router(config)# cable downstream qam-profile 3`	Defines or modifies a QAM profile.
Step 4	**annex** {**A** | **B** | **C**} **Example:**	Defines the profile MPEG framing format. The default is Annex B.

	Command or Action	Purpose
	`Router(config-qam-prof)# annex A`	
Step 5	**description** *LINE* **Example:** `Router(config-qam-prof)# description qam1`	Name or description for this profile.
Step 6	**interleaver-depth** {**I12-J17** | **I128-J1** | **I128-J2** | **I128-J3** | **I128-J4** | **I128-J5** | **I128-J6** | **I128-J7** | **I128-J8** | **I16-J8** | **I32-J4** | **I64-J2** | **I8-J16**} **Example:** `Router(config-qam-prof)# interleaver-depth I64-J2`	Defines the interleaver depth. The default is I32 J4 for DOCSIS.
Step 7	**modulation** {**256** | **64**} **Example:** `Router(config-qam-prof)# modulation 64`	Defines the modulation. The default is 256QAM.
Step 8	**spectrum-inversion** {**off** | **on**} **Example:** `Router(config-qam-prof)# spectrum-inversion on`	Enables or disables spectrum inversion. Default is off.
Step 9	**symbol-rate** *value* **Example:** `Router(config-qam-prof)# symbol-rate 5057`	Defines the symbol rate. Value is in kilo-symbol/sec.
Step 10	**exit** **Example:** `Router(config-qam-prof)# exit`	Exits from the QAM profile configuration mode.

Configuring the Frequency Profile on the Downstream Channels

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**cable downstream freq-profile** *DS_frequency_profile_ID* **Example:** `Router(config)# cable downstream freq-profile 4`	Defines or modifies a frequency profile.
Step 4	**lane** *lane_id* **start-freq** *start_freq_value* **Example:** `Router(config-freq-prof)# lane 1 start-freq 90000000`	Defines the frequency lanes.

Step 5	**block** *block_id* **start-freq** *bl_start_freq_value*
	Example:
	`Router(config-freq-prof-lane)#block 1 start-freq 90000000`
	`Router(config-freq-prof-lane)#`
		Configures the lane frequency blocks.

Step 6	**exit** **Example:** `Router(config-freq-prof-lane)# exit`	Exits from the frequency lane configuration mode.

Configuring the Controller on the Downstream Channels

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**controller integrated-cable** *slot/subslot/port* **Example:** `Router(config)#controller Integrated-Cable 3/0/0`	Enters the controller sub-mode.
Step 4	**base-channel-power** *value* **Example:** `Router(config-controller)#base-channel-power 26`	Sets the base channel power level. If not specified, the default value is calculated based on the number of carriers. Maximum limit is 34 dBmV DRFI. If you configure a value greater

	Command or Action	Purpose
		than the maximum specified by DRFI, the following message is displayed: Caution: RF Power above DRFI specification. May result in minor fidelity degradation.
Step 5	**freq-profile** *number* **Example:** `Router(config-controller)#freq-profile 0`	Specifies the frequency profile for the port.
Step 6	**max-carrier** *value* **Example:** `Router(config-controller)#max-carrier 1`	Specifies the maximum number of carriers.
Step 7	**mute** **Example:** `Router(config-controller)#mute`	Mutes the port. Use the **no** prefix to unmute the port. Default is "no mute".
Step 8	**rf-chan** *starting_Qam_ID ending_Qam_ID* **Example:** `Router(config-controller)#rf-chan 0 1`	Enters RF channel configuration sub-mode to configure an individual channel or a block of channels.
Step 9	**shutdown** **Example:** `Router(config-controller)#shutdown`	Changes the port administration state to down. Use the **no** prefix to change the port administration state to up.

Configuring the RF Channel on a Controller

The RF channel submode is entered from the channel controller configuration submode using the **rf-chan** command as described in the previous section. If an individual channel was specified in the **rf-chan** command, only that channel configuration is changed. If a block of channels was specified in the **rf-chan** command, the configuration change is applied to all channels in the block.

Procedure

	Command or Action	Purpose
Step 1	**docsis-channel-id** *dcid* **Example:** `Router(config-rf-chan)#docsis-channel-id 1`	Changes the channel DOCSIS channel identifier. In block mode, the value is assigned to the first channel and incremented for successive channels.
Step 2	**frequency** *value* **Example:** `Router(config-rf-chan)#frequency 93000000`	Configures the channel's center frequency in Hz. The available frequency range is determined from the port's frequency profile, if configured. If not configured, the available range will be

	Command or Action	Purpose
		the full port spectrum. In block mode, the frequency will be assigned to the first channel. Successive channels will get the next center frequency for the annex specified in the QAM profile (+6 Hz for Annex B, +8 Hz for Annex A).
Step 3	**mute** **Example:** `Router(config-rf-chan)#mute`	Mutes the RF channel. Enter the **no** prefix to unmute the channel. Default is "no mute".
Step 4	**power-adjust** *pwr_adj_range* **Example:** `Router(config-rf-chan)#power-adjust 8.0 - 0.0 dBmV`	Adjusts the RF channel's power.
Step 5	**qam-profile** *qam_profile_number* **Example:** `Router(config-rf-chan)#qam-profile 0`	Specifies the QAM profile for this channel.
Step 6	**rf-output** *value* **Example:** `Router(config-rf-chan)#rf-output normal`	Changes the RF output mode to test the channel.
Step 7	**shutdown** **Example:** `Router(config-rf-chan)#shutdown`	Changes the channel administration state to down. Use the **no** prefix to change the channel administration state to up. The default is "no shut".
Step 8	**type** *value* **Example:** `Router(config-rf-chan)#type video`	Configures the channel QAM type. The default is DOCSIS.

Configuration Examples

Downstream Interface Configuration Example

The example below shows the configuration of:

- QAM Profile—The system defined QAM profile for Annex B and 256 QAM.
- Frequency Profile—The system defined frequency profile annex-b-low.
- Controller and RF channel—Port 0 on slot 3/0 with frequency profile 0; 96 channels with QAM profile 1 and center frequencies starting at 93 MHz.

```
cable downstream qam-profile 1
  annex B
  modulation 256
  interleaver-depth I32-J4
  symbol-rate 5361
  spectrum-inversion off
  description default-annex-b-256-qam
cable downstream freq-profile 0
  lane 1 start-freq 90000000
    block 1 start-freq 90000000
    block 2 start-freq 138000000
    block 3 start-freq 186000000
    block 4 start-freq 234000000
  lane 2 start-freq 282000000
    block 1 start-freq 282000000
    block 2 start-freq 330000000
    block 3 start-freq 378000000
    block 4 start-freq 426000000
  lane 3 start-freq 474000000
    block 1 start-freq 474000000
    block 2 start-freq 522000000
    block 3 start-freq 570000000
    block 4 start-freq 618000000
  lane 4 start-freq 666000000
    block 1 start-freq 666000000
    block 2 start-freq 714000000
    block 3 start-freq 762000000
    block 4 start-freq 810000000
controller Integrated-Cable 3/0/0
 max-carrier 128
 base-channel-power 34
 freq-profile 0
 rf-chan 0 95
  type DOCSIS
  frequency 93000000
  rf-output NORMAL
  power-adjust 0
  docsis-channel-id 1
  qam-profile 1
```

Show Command Examples for Displaying the State

Use the following commands to display the state of any QAM profile, Frequency profile, downstream controller or channel.

QAM Profile Configuration Example

```
Router#show cable qam-profile 0
QAM Profile ID 0: default-annex-b-64-qam
        annex: B
        modulation: 64
        interleaver-depth: I32-J4
        symbol rate: 5057 kilo-symbol/second
        spectrum-inversion: off
Router#
```

Frequency Profile Configuration Example

```
Router#show cable freq-profile 0
Frequency Profile ID 0 annex-b-low:
        Lane 1 start-freq 90000000hz
                Block 1 start-freq 90000000hz
                Block 2 start-freq 138000000hz
                Block 3 start-freq 186000000hz
                Block 4 start-freq 234000000hz
        Lane 2 start-freq 282000000hz
                Block 1 start-freq 282000000hz
                Block 2 start-freq 330000000hz
                Block 3 start-freq 378000000hz
                Block 4 start-freq 426000000hz
        Lane 3 start-freq 474000000hz
                Block 1 start-freq 474000000hz
                Block 2 start-freq 522000000hz
                Block 3 start-freq 570000000hz
                Block 4 start-freq 618000000hz
        Lane 4 start-freq 666000000hz
                Block 1 start-freq 666000000hz
                Block 2 start-freq 714000000hz
                Block 3 start-freq 762000000hz
                Block 4 start-freq 810000000hz
Router#
```

Controller Configuration Example

```
Router#show controller Integrated-Cable 3/0/0 rf-port
  Admin:  UP  MaxCarrier: 128  BasePower: 34 dBmV  Mode: normal
  Rf Module 0: UP
  Frequency profile: 0
  Free freq block list has 1 blocks:
    666000000 - 863999999
  Rf Port Status: UP
Router#
```

RF Channel Configuration Example

```
Router#show controller integrated-Cable 3/0/0 rf-channel 0-3 95
Chan State Admin Frequency  Type   Annex Mod  srate Interleaver  dcid  power  output
 0     UP   UP   93000000   DOCSIS  B    256   5361  I32-J4       1    34     NORMAL
 1     UP   UP   99000000   DOCSIS  B    256   5361  I32-J4       2    34     NORMAL
 2     UP   UP   105000000  DOCSIS  B    256   5361  I32-J4       3    34     NORMAL
 3     UP   UP   111000000  DOCSIS  B    256   5361  I32-J4       4    34     NORMAL
 95    UP   UP   663000000  DOCSIS  B    256   5361  I32-J4       96   34     NORMAL

Router# show controller integrated-Cable 3/0/0 rf-channel 0 verbose
Chan State Admin Frequency  Type   Annex Mod  srate Interleaver  dcid  power  output
 0     UP   UP   93000000   DOCSIS  B    256   5361  I32-J4       1    34     NORMAL
 Qam profile: 1
 Spectrum Inversion: Off
 Frequency Lane: 1  Block: 1  index: 1
 Resource status:   OK
 License: granted <02:00:04 EDT Jan 2 2012>
 JIB channel number: 0
Chan EnqQ  Pipe  RAF SyncTmr Vid Mac            Video Primary DqQ  TM Mpts Sniff
   0     0     0   4       0   0 0000.0000.0000     0       0    0  0    0 NO
Grp  Prio P  Prate Phy0-ctl Phy1-ctl Enable Tun-Id  L2TPv3_Ses_id
   0    0  0     1        1        0   TRUE       0      0
Chan  Qos-Hi Qos-Lo  Med-Hi Med-Lo  Low-Hi Low-Lo
   0  32774  16384   32768  16384   65536  32768
```

```
Chan  Med Low TB-neg Qos_Exc  Med_Xof  Low_Xof       Qdrops  Pos Qlen(Hi-Med-lo) Fl
    0   0   0      0       0        0        0            0  Y     0     0     0  0
DSPHY Info:
  DSPHY Register Local Copy: QPRHI = c0000163, QPRLO = e30d0
  DSPHY Register Local Copy Vaddr = 80000290, qam2max_mapping = 80000000
  DSPHY Register Local Copy: SPR ID = 0, SPR Mapping= c200000a
  Last read from HW: Mon Jan  2 02:02:04 2012
   QPRHI = c0000163, QPRLO = e30d0, SPR = c200000a SPRMAPING c0000000 Q2Max 80000000
  Last time read spr rate info from HW: Mon Jan  2 13:21:41 2012
    SPR ID 0, rate value in kbps 0, overflow count 0, underflow count 0


Router#show controllers Integrated-Cable 3/0/0 counter rf-channel

Controller  RF    MPEG       MPEG       MPEG       Sync       MAP/UCD
            Chan  Packets    bps        Mbps       Packets    Packets
                  Tx                               Tx         Tx
3/0/0       0     6          0          0.000000   0          6
3/0/0       1     6          0          0.000000   0          6
3/0/0       2     6          0          0.000000   0          6
3/0/0       3     6          0          0.000000   0          6
3/0/0       4     6          0          0.000000   0          6
3/0/0       5     6          0          0.000000   0          6
3/0/0       6     6          0          0.000000   0          6
3/0/0       7     6          0          0.000000   0          6
3/0/0       8     5124124    1381035    1.332459   329444     6531411


Router# show cable licenses ds
-------------------------------------------
Entitlement:  Downstream License
Consumed count: 672
Consumed count reported to SmartAgent: 672
Forced-Shut count: 0
Enforced state: No Enforcement

Router#
```

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

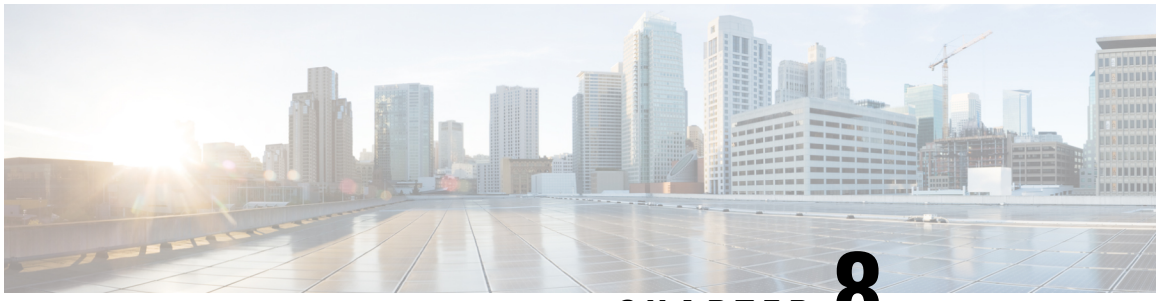
Feature Information for Downstream Interface Configuration on the Cisco cBR Router

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 13: Feature Information for Downstream Interface Configuration

Feature Name	Releases	Feature Information
Downstream Interface Configuration	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 8

Upstream Interface Configuration

This document describes how to configure the upstream interfaces on the Cisco cBR Series Converged Broadband Router .

- Finding Feature Information, on page 137
- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 137
- Information About Upstream Interface Configuration, on page 138
- How to Configure Upstream Interfaces, on page 139
- Configuration Examples, on page 142
- Additional References, on page 143
- Feature Information for Upstream Interface Configuration on the Cisco cBR Router, on page 143

Finding Feature Information

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 14: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About Upstream Interface Configuration

Upstream Channel Management

Upstream Channel Management (UCM) is responsible for the physical (PHY) layer configuration and resource management of upstream channels in the Cisco cBR Series Converged Broadband Router .

Upstream Controller

An upstream port represents a physical upstream RF connector on a cable line card, connected to one or more fiber nodes. An upstream RF port is a container of upstream RF channels, which imposes constraints on both topology and spectrum for the group of RF channels contained in the physical port. An upstream RF port also represents the RF front-end hardware component on a cable line card including the connector, variable gain adjustment (VGA), and A/D converter. This is directly connected to a set of upstream physical channel receivers. The number of upstream physical channels per port is thus constrained by the number of receivers accessible to the port.

Upstream Channel

An upstream RF channel represents DOCSIS physical layer operation on a single upstream center frequency with a particular channel width. It is contained by a single physical port on the CMTS line card hardware.

Upstream Resource Management

The upstream resource management (URM) feature is primarily responsible for the maintenance of the relationship between a physical upstream connector on the line card and the upstream RF channels received on that connector.

How to Configure Upstream Interfaces

This section contains the following:

Configuring the Cisco CMTS Manually Using Configuration Mode

Connect a console terminal to the console port on the I/O controller. When asked if you want to enter the initial dialog, answer **no** to go into the normal operating mode of the router. After a few seconds the user EXEC prompt (**Router>**) appears.

Configuring the Modulation Profile and Assigning to an Upstream Channel

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable modulation-profile** *profile mode_of_oper qam_profile* **Example:** `Router(config)# cable modulation-profile 23 tdma qam-16`	Creates a preconfigured modulation profile, where the burst parameters are set to their default values for each burst type.
Step 4	**Controller Upstream-Cable** *slot/subslot/port* **Example:** `Router(config)# Controller Upstream-Cable 7/0/0`	Enters the controller interface configuration mode.
Step 5	**us-channel** *n* **modulation-profile** *primary-profile-number* [*secondary-profile-number*] [*tertiary-profile-number*] **Example:**	Assigns up to three modulation profiles to an upstream port.

	Command or Action	Purpose
	`Router(config-if)#cable upstreamus-channel 0 modulation-profile 23`	
Step 6	**end** **Example:** `Router(config-controller)# end`	Exits controller configuration submode and returns to privileged EXEC mode.

Configuring the Upstream Channel with PHY Layer

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**controller upstream-cable** *slot/subslot/port* **Example:** `Router(config)# controller upstream-cable 1/0/0`	Specifies the controller interface line card and enters upstream controller config configuration submode.
Step 4	**us-channel** *rf-channel* **frequency** *freq-val* **Example:** `Router(config-controller)# us-channel 1 frequency 20000000`	Assigns frequency to an RF channel on a controller interface.
Step 5	**us-channel** *rf-channel* **docsis-mode** *mode* **Example:** `Router(config-controller)# us-channel 1 docsis-mode tdma`	Assigns DOCSIS mode to an RF channel on a controller interface.
Step 6	**us-channel** *rf-channel* **channel-width** *value* **Example:** `Router(config-controller)# us-channel 1 channel-width 3200000`	Assigns channel width in Hertz to an RF channel on a controller interface.
Step 7	**us-channel** *rf-channel* **modulation-profile** *profile* **Example:**	Assigns modulation profile to an RF channel on a controller interface.

	Command or Action	Purpose
	`Router(config-controller)# us-channel 1 modulation-profile 21`	
Step 8	**no us-channel** *rf-channel* **shutdown** **Example:** `Router(config-controller)# no us-channel 1 shutdown`	Enables the upstream channel.
Step 9	**end** **Example:** `Router(config-controller)# end`	Exits upstream controller configuration submode and returns to privileged EXEC mode.

Associating Upstream Channels with a MAC Domain and Configuring Upstream Bonding

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** *slot/subslot/cable-interface-index* **Example:** `Router(config)# interface cable 7/0/0`	Specifies the cable interface line card on a Cisco CMTS router.
Step 4	**downstream integrated-cable** *slot/subslot/port* **rf-channel** *rf-chan* [**upstream** *grouplist*] **Example:** `Router(config-if)# downstream integrated-cable 7/0/0 rf-channel 3 upstream 3`	Associates a set of upstream channels to the integrated downstream channels.
Step 5	**upstream** *md-us-chan-id* **upstream-cable** *slot/subslot/port* **us-channel** *rf-channel* **Example:** `Router(config-if)# upstream 0 upstream-cable 7/0/0 us-channel 0`	Associates a set of physical upstream channels with the Mac Domain.

	Command or Action	Purpose
Step 6	**cable upstream bonding-group** *id* **Example:** `Router(config-if)# cable upstream bonding-group 200`	Creates the upstream bonding group on the specified cable interface and enters upstream bonding configuration submode.
Step 7	**upstream** *number* **Example:** `Router(config-upstream-bonding)# upstream 1`	Adds an upstream channel to the upstream bonding group. A maximum of 16 upstream channels can be configured for each MAC Domain, which are divided into two groups: • Group 1: upstream channel 0-7 • Group 2: upstream channel 8-15 The **upstream bonding-group** should include all the upstream channels either from Group 1 or Group 2 only.
Step 8	**attributes** *value* **Example:** `Router(config-upstream-bonding)# attributes eeeeeeee`	Modifies the attribute value for the specified upstream bonding group.
Step 9	**end** **Example:** `Router(config-upstream-bonding)# end`	Exits upstream bonding configuration submode and returns to privileged EXEC mode.

Configuration Examples

Upstream Channel with PHY Layer Configuration Example

```
...
us-channel 0 frequency 20000000
us-channel 0 channel-width 3200000 3200000
us-channel 0 power-level 0
us-channel 0 docsis-mode tdma
us-channel 0 minislot-size 2
us-channel 0 modulation-profile 21
no us-channel 0 shutdown
...
```

Upstream Channels with a MAC Domain Configuration Example

```
...
interface Cable8/0/0
downstream Modular-Cable 8/0/0 rf-channel 0
upstream 0 Upstream-Cable 8/0/0 us-channel 0
upstream 1 Upstream-Cable 8/0/0 us-channel 1
```

```
cable mtc-mode
cable upstream bonding-group 1
  upstream 0
  upstream 1
  attributes 80000000

...
```

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

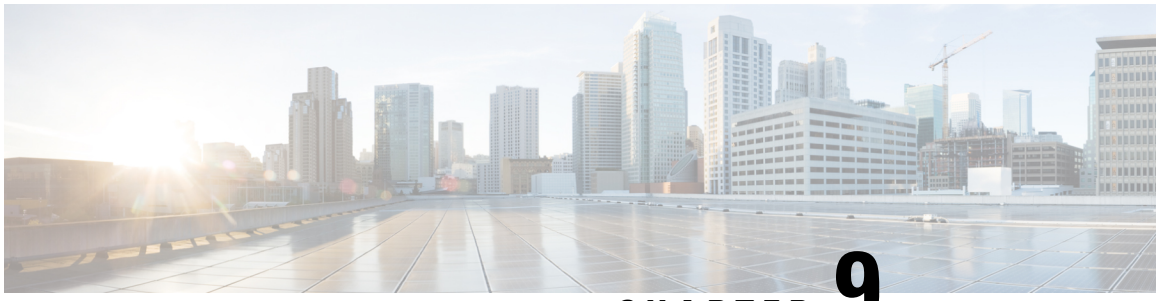
Feature Information for Upstream Interface Configuration on the Cisco cBR Router

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 15: Feature Information for Upstream Interface Configuration

Feature Name	Releases	Feature Information
Upstream Interface Configuration	Cisco IOS-XE 3.15.0S	This feature was integrated into the Cisco cBR Series Converged Broadband Router .



CHAPTER 9

DOCSIS Interface and Fiber Node Configuration

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 145
- Overview of DOCSIS Interfaces and Fiber Node Configurations, on page 146
- Configuring DOCSIS Interfaces and Fiber Nodes, on page 147
- Configuring MAC Domain Service Groups, on page 153
- Downstream Bonding Group Configuration, on page 156
- Upstream Bonding Group Configuration, on page 160
- Additional References, on page 163
- Feature Information for DOCSIS Interface and Fiber Node Configuration , on page 163

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 16: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Overview of DOCSIS Interfaces and Fiber Node Configurations

The Interface line card used in the Cisco cBR chassis is an integrated line card that has two downstream modules and one upstream module. The line cards support DOCSIS 3.0 features including downstream bonding groups, and upstream bonding groups.

Downstream Features

Physically, the DS (downstream) modules support eight physical connectors or ports. The DS modules support the following features:

- The DS modules support eight downstream integrated-cable controllers for these eight ports. Each downstream integrated-cable controller is associated with an RF port.
- Each downstream controller supports up to 128 downstream channels (0-127).
- Each downstream controller can be configured with 128 integrated-cable interfaces. Therefore, each line card has 1024 integrated-cable interfaces.
- Each integrated-cable interface has a static mapping to an integrated-cable controller RF channel. For example, Integrated-Cable interface 3/0/0:0 is mapped to RF Channel 0 on Integrated-Cable controller 3/0/0.
- 768 downstream DOCSIS channels may be configured on each line card.

- A total of 512 wideband-cable interfaces (downstream bonding groups) may be configured on each line card.
  - Each wideband-cable interface supports a maximum of 64 downstream channels.
  - 128 of the 512 wideband-cable interfaces (downstream bonding groups) may contain 33 or more channels.

Upstream Features

The Interface line card has one upstream module supporting 16 physical connectors or ports. The upstream features are as follows:

- The line card supports 16 upstream-cable controllers, each mapping to one upstream connector.
- 12 upstream channels can be configured per upstream controller.
- 12 upstream channels can be enabled per pair of upstream controllers.

For more details on the upstream features, see the *Downstream Upstream Guide*.

MAC Domains (Cable Interfaces)

1. 16 MAC domains (cable interfaces) may be configured per line card.
2. Maximum of 16 upstream channels can be configured in each MAC domain.
3. A maximum of 255 downstream channels may be added to a MAC domain.
4. Maximum number of primary capable downstream channels per MAC Domain is 32. Non Primary downstream channels are added automatically to the MAC domains when the fiber nodes are configured.

Fiber Nodes

512 fiber nodes may be configured for each Cisco cBR-8 chassis.

Configuring DOCSIS Interfaces and Fiber Nodes

Configuring Upstream Channels

Verifying the Controller Configuration

Use the **show controllers upstream-cable** command to verify the configuration of the upstream channels in the controllers. Use the modifier **| include upstream** to see the administrative and operational state of the controllers.

```
Router#show controllers upstream-Cable 1/0/0 | include upstream
Controller 1/0/0 upstream 0  AdminState:UP OpState: UP
Controller 1/0/0 upstream 1  AdminState:UP OpState: UP
Controller 1/0/0 upstream 2  AdminState:UP OpState: UP
Controller 1/0/0 upstream 3  AdminState:UP OpState: UP
Controller 1/0/0 upstream 4  AdminState:DOWN Opstate: DOWN (Reason: Default)
Controller 1/0/0 upstream 5  AdminState:DOWN Opstate: DOWN (Reason: Default)
Controller 1/0/0 upstream 6  AdminState:DOWN Opstate: DOWN (Reason: Default)
```

```
Controller 1/0/0 upstream 7  AdminState:DOWN Opstate: DOWN (Reason: Default)
Router#
```

Binding Upstream Channels to MAC Domain

By default, a MAC domain does not contain any upstream channels. This section describes the configurations necessary to bind one or more upstream channels to a MAC domain. Each upstream channel is bound to only one MAC domain. The MAC domain and the upstream channel must reside on the same line card (same slot). If required, the upstream channels within the same upstream controller can be bound to different MAC domains.

Before you begin

Restrictions

- A maximum of 8 upstream channels may be bound to one MAC domain.

Note A maximum of 16 upstream channels may be configured in each MAC domain.

- The MAC domain and channels must share the same slot. That is, a MAC Domain may include channels from any controller on the same slot.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** **Example:** `Router#interface cable 1/0/0`	Enters MAC Domain configuration mode. Values for slot are 0-3 and 6-9, for subslot is always 0, for MD Index is 0-15.
Step 4	**upstream upstream-Cable us-channel** **Example:** `Router(config-if)#upstream 4 upstream-Cable 1/0/0 us-channel 7`	Binds the specified upstream channel to the MAC Domain.
Step 5	**end** **Example:** `Router# end`	Returns to privileged EXEC mode.

What to do next

To verify MAC Domain configurations for upstream, use the **show cable mac-domain** command with **cgd-associations** keyword.

The MD US Binding table shows the upstream channel binding.

```
Router#show cable mac-domain c1/0/0 cgd-associations
Load for five secs: 1%/0%; one minute: 2%; five minutes: 2%
Time source is NTP, *13:36:26.209 PST Fri Jan 20 2012
CGD Host  Resource  DS Channels            Upstreams    (ALLUS)   Active DS
Ca1/0/0   1/0/0     8                      0-1          Yes        8

                    16                     0-1          Yes        16

                    24                     0-1          Yes        24

                    32-33                  0-1          Yes        32-33

                    40                     0-1          Yes        40


MD US binding:
Host    MD Controller US channel State
Ca1/0/0 U0 1/0/0      0          UP
Ca1/0/0 U1 1/0/0      1          UP
Ca1/0/0 U2 1/0/0      2          UP
Ca1/0/0 U3 1/0/0      3          UP
Ca1/0/0 U4 1/0/1      0          UP
Ca1/0/0 U5 1/0/1      1          UP
Ca1/0/0 U6 1/0/1      2          UP
Ca1/0/0 U7 1/0/1      3          UP

Router#
```

Configuring Primary Capable Downstream Channels

Verifying Downstream Configuration in Controller

Use the **show controller Integrated-Cable** command to verify the status of the downstream channels configured on an Integrated-cable controller.

```
Router# show controller Integrated-Cable 1/0/0 rf-channel 0-127
Chan State Admin Frequency Type   Annex Mod srate Interleaver dcid power output
0 UP UP    381000000       DOCSIS B     256 5361  I32-J4      1    32    NORMAL
1 UP UP    387000000       DOCSIS B     256 5361  I32-J4      2    34    NORMAL
2 UP UP    393000000       DOCSIS B     256 5361  I32-J4      3    34    NORMAL
3 UP UP    399000000       DOCSIS B     256 5361  I32-J4      4    34    NORMAL
```

Configuring Integrated-cable Interface

Configure an integrated-cable interface to prepare a downstream channel for inclusion within a MAC Domain as a primary-capable downstream channel. The interface configuration provides the following advantages:

- It enables the allocation of bandwidth to the downstream channel.
- It enables the control of the administrative state (shut/no shut) of the channel interface.

Each integrated-cable interface is mapped statically to an integrated-cable controller RF channel. For example, IC interface 1/0/0:0 is mapped to IC controller 1/0/0 RF channel 0. Similarly, IC interface 1/0/0:1 is mapped to IC controller 1/0/0 RF channel 1.

IC controllers are numbered 0-7 and RF Channels on each controller are numbered 0-127.

Before you begin

Determine the percentage of bandwidth to allocate to a channel. The bandwidth percentage configured is converted to a committed information rate (CIR) value for the interface. The value is used to admit non-bonded service flows on this channel. For more information, see the *Dynamic Bandwidth Sharing on the Cisco CMTS Router*.

Procedure

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **interface integrated-cable**

Example:

```
Router(config)# interface integrated-cable 1/0/0:0
```

Enter the integrated-cable interface configuration mode for specified integrated-cable interface.

Step 4 **cable rf-bandwidth-percent** *percentage-number*

Example:

```
Router(config-if)# cable rf-bandwidth-percent 30
```

Configures the bandwidth allocation to the specified integrated-cable interface.

Step 5 **end**

Example:

```
Router# end
```

Returns to privileged EXEC mode.

What to do next

The following conditions are used to determine if an IC (Integrated-Cable) interface is up in current software.

- The IC interface is associated to a MD (MAC Domain) interface.
- The MD interface, which the IC interface associated to, is in UP state.
- The IC interface is not configured shut down.
- The IC interface is configured with bandwidth.
- The associated downstream channel within the IC controller is operationally up.

Use the **show interface Integrated-Cable controller** command to verify the status of the specified integrated-cable interface. The State info table provides information to diagnose issues affecting the operational state of the interface.

```
Router# show interface Integrated-Cable 1/0/0:0 controller
Integrated-Cable1/0/0:0 is up, line protocol is up
…
----------------------------------------------
State info (DSNB if and its underlying states)
----------------------------------------------
DSNB IF state : UP
RF Chan state : UP
RF Chan frequency : 381000000
Bandwidth configured on DSNB IF : YES
Inject Header/HW flow creation status : DSNB_IF_SM_UP
MD state (9/0/0) : UP
*DSNB i/f Line State : UP
----------------------------------------------
```

Binding Primary Capable Downstream Channels to a MAC Domain

After a downstream channel has a properly configured Integrated-cable interface, it may be bound to a MAC Domain as a primary capable channel. The Channel Grouping Domain (CGD) configuration allows specified downstream channels to be bound to a MAC Domain as primary capable channels. Optionally, it also allows downstream channels to be associated with a subset of upstream channels within the MAC domain.

Before you begin

Restrictions

- The downstream channel and MAC domain must reside on the same line card (same slot)
- A maximum of 32 primary capable downstream channels may be bound to a single MAC domain

Procedure

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **interface cable**

Example:

```
Router#interface cable 1/0/0
```

Enters MAC domain configuration mode.

- *slot*—Specifies the chassis slot number of the interface line card. Valid values are 0-3 and 6-9
- *subslot*—Specifies the secondary slot number of the interface line card. Valid subslot is 0.
- *MD index*—Specifies the MAC Domain index number. Valid values are 0-15.

Step 4 **downstream Integrated-Cable** *slot/subslot/port* **rf-channels** *grouplist*

Example:

```
Router#downstream Integrated-Cable 1/0/0 rf-channels 1-6
```

Configures the downstream primary capable channels.

- *grouplist*—Specify the range of downstream rf-channels.

Step 5 **end**

Example:

```
Router# end
```

Returns to privileged EXEC mode.

What to do next

To verify the downstream primary capable channels, use the **show cable mac-domain** command with **cgd-associations** keyword.

```
Router#show cable mac-domain c1/0/0 cgd-associations
Load for five secs: 1%/0%; one minute: 2%; five minutes: 2%
Time source is NTP, *13:36:26.209 PST Fri Jan 20 2012
CGD Host  Resource  DS Channels           Upstreams    (ALLUS)   Active DS
Ca1/0/0   1/0/0     8                     0-1          Yes        8

                    16                    0-1          Yes        16

                    24                    0-1          Yes        24

                    32-33                 0-1          Yes        32-33

                    40                    0-1          Yes        40
```

```
MD US binding:
Host    MD Controller US channel State
Ca1/0/0 U0 1/0/0      0          UP
Ca1/0/0 U1 1/0/0      1          UP
Ca1/0/0 U2 1/0/0      2          UP
Ca1/0/0 U3 1/0/0      3          UP
Ca1/0/0 U4 1/0/1      0          UP
Ca1/0/0 U5 1/0/1      1          UP
Ca1/0/0 U6 1/0/1      2          UP
Ca1/0/0 U7 1/0/1      3          UP

Router#
```

Configuring MAC Domain Service Groups

Configuring the Fiber Nodes

A maximum of 512 fiber nodes may be configured per CMTS. A fiber node configured on the CMTS represents one or more matching physical fiber nodes in the HFC plant. The CMTS uses the fiber node configuration to identify the DOCSIS downstream service group (DS-SG) and DOCSIS upstream Service Group (US-SG) of the physical fiber nodes in the plant. The Service Group information is compared with MAC Domain channel configuration to automatically calculate the MAC Domain downstream and upstream service groups (MD-DS-SGs and MD-US-SGs respectively) within the MAC Domains.

The following is required to create a valid fiber node configuration.

- Each fiber node configuration must include at least one downstream controller and one upstream controller.
- Every MAC Domain and Wideband Interface including channels within the fiber node must use the same bundle interface.
- All downstream channels included within the fiber node must be assigned a unique frequency.
- All downstream channels associated with a particular MAC Domain must be assigned a unique DOCSIS channel ID.

If automatic DOCSIS channel ID allocation is preferred over manual DOCSIS channel ID configuration, the **cable downstream-channel-id automatic** command may be used to enable automatic DOCSIS channel ID allocation for the CMTS.

For details, see the *Cisco CMTS Cable Command Reference*.

Procedure

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **cable fiber-node** *id*

Example:

```
Router(config)#cable fiber-node 1
Router(config-fiber-node)#
```

Enters cable fiber-node configuration mode to configure a fiber node

- *id*— cable fiber node ID. The valid range is 1-512.

Step 4 **downstream Integrated-Cable** *slot/subslot/port*

Example:

```
Router(config-fiber-node)#downstream Integrated-Cable 1/0/0
```

Adds the DOCSIS downstream channels within the controller to the fiber node.

Step 5 **upstream upstream-Cable** *slot/subslot/port*

Example:

```
Router(config-fiber-node)#upstream upstream-Cable 1/0/0
```

Adds the upstream channels within the controller to the fiber node.

Step 6 **end**

Example:

```
Router# end
```

Returns to privileged EXEC mode.

What to do next

To verify the fiber-node configuration use the **show cable fiber-node** command.

```
Router# show cable fiber-node 1
------------------------------------------------------------------------------
--
Fiber-Node 1
Description: Feed Mac Domain: Cable1/0/0
Channel(s) : downstream Integrated-Cable 1/0/0: 0-3, 32-35, 64-67,
96-99
Channel ID(s): 1 2 3 4 33 34 35 36 65 66 67 68 97 98
99 100
Upstream-Cable 1/0/0
FN Config Status: Configured (status flags = 0x01)
MDD Status: Valid
Router#
```

The output shows the downstream channel IDs configured on a fiber node. It also shows the status of the upstream-cable configured on the fiber node. Further, it shows the status of MAC Domain Descriptor (MDD) messaging.

Verify MD-DS-SG Channel Membership

Once the fiber node is valid, the MD-DS-SGs within the associated MAC domains will be automatically populated with downstream channels. The MD-DS-SGs will include the active primary downstream channels within the MAC domain as well as non-primary downstream channels automatically associated with the MAC domain through the fiber node configuration. The non-primary channels must be properly configured within the controller (that are operationally up) to be included in MD-DS-SGs

Use the **show cable mac-domain** command with the **downstream-service-group** option to display the MD-DS-SG channel membership.

```
outer#show cable mac-domain c1/0/0 downstream-service-group
Cable MD-DS-SG RF
IF Id Resource Chan  Primary Chan
C1/0/0 5 1/0/0 0-3   0-3
               32-35 32-35
               64-67
               96-99
```

To verify that the primary downstream channels are transmitting MAC Management Messages (MMMs) use the **show controller Integrated-Cable counter rf-channel** command.

```
Router#show controller Integrated-Cable 1/0/0 counter rf-channel
Controller  RF    MPEG       MPEG       MPEG       Sync       MAP/UCD
            Chan  Packets    bps        Mbps       Packets    Packets
                  Tx                               Tx         Tx
1/0/0       0     51256      1504       0.001504   0          51256
1/0/0       1     51256      1504       0.001504   0          51256
1/0/0       2     51256      1504       0.001504   0          51256
1/0/0       3     51256      1504       0.001504   0          51256
1/0/0       4     51256      1504       0.001504   0          51256
1/0/0       5     51256      1504       0.001504   0          51256
1/0/0       6     51256      1504       0.001504   0          51256
1/0/0       7     51256      1504       0.001504   0          51256
1/0/0       8     47625214   1416567    1.367991   5124345    50884388
1/0/0       9     51256      1504       0.001504   0          51256
1/0/0       10    51256      1504       0.001504   0          51256
1/0/0       11    51256      1504       0.001504   0          51256
1/0/0       12    51256      1504       0.001504   0          51256
1/0/0       13    51256      1504       0.001504   0          51256
1/0/0       14    51256      1504       0.001504   0          51256
1/0/0       15    51256      1504       0.001504   0          51256
1/0/0       16    47627672   1414913    1.366337   5124342    50884344
1/0/0       17    51256      1504       0.001504   0          51256
1/0/0       18    51256      1504       0.001504   0          51256
1/0/0       19    51256      1504       0.001504   0          51256
1/0/0       20    51256      1504       0.001504   0          51256
1/0/0       21    51256      1504       0.001504   0          51256
1/0/0       22    51256      1504       0.001504   0          51256
1/0/0       23    51256      1504       0.001504   0          51256
1/0/0       24    47629532   1414862    1.366286   5124341    50884325
1/0/0       25    51256      1504       0.001504   0          51256
1/0/0       26    51256      1504       0.001504   0          51256
```

```
1/0/0       27    51256      1504       0.001504   0          51256
1/0/0       28    51253      1504       0.001504   0          51253
1/0/0       29    51253      1504       0.001504   0          51253
1/0/0       30    51253      1504       0.001504   0          51253
1/0/0       31    51253      1504       0.001504   0          51253
1/0/0       32    47642351   1412005    1.363429   5124337    50891994
1/0/0       33    47646042   1412757    1.364181   5124336    50892055
1/0/0       34    51251      1504       0.001504   0          51251
1/0/0       35    51251      1504       0.001504   0          51251
1/0/0       36    51251      1504       0.001504   0          51251
1/0/0       37    51251      1504       0.001504   0          51251
1/0/0       38    51251      1504       0.001504   0          51251
1/0/0       39    51251      1504       0.001504   0          51251
1/0/0       40    47634991   1409649    1.361073   5124329    50884177
1/0/0       41    51251      1504       0.001504   0          51251
```

Verify MD-US-SG Channel Membership

Use the **show cable mac-domain** command with the **upstream-service-group** option to display the MD-US-SG channel membership.

```
Router#show cable mac-domain c1/0/0 upstream-service-group
Cable MD 1/0/0
US-SG-ID : 5 US-Chan : U0,1,2,3
Primary-DS: 1/0/0:0 US-SG-ID: 5
MDD US-List : U0,1,2,3
MDD Ambiguity : U0,1,2,3
Primary-DS: 1/0/0:1 US-SG-ID: 5
MDD US-List : U0,1,2,3
MDD Ambiguity : U0,1,2,3
Primary-DS: 1/0/0:2 US-SG-ID: 5
MDD US-List : U0,1,2,3
MDD Ambiguity : U0,1,2,3
Primary-DS: 1/0/0:3 US-SG-ID: 5
MDD US-List : U0,1,2,3
MDD Ambiguity : U0,1,2,3
```

Downstream Bonding Group Configuration

Configuring Wideband-cable Interface (Downstream Bonding Grouping)

A Wideband-Cable interface forwards bonded traffic in the downstream direction. A set of downstream RF channels is configured under the Wideband interface. Each line card will support up to 512 Wideband interfaces.

Although there is no real relationship between these wideband interfaces and the 8 controllers (ports), there is a convention of dividing the wideband interfaces into groups per controller.

The 512 wideband interfaces are divided among the 8 controllers - 64 interfaces per controller.

You can create a wideband-cable interface to define a downstream bonding group. A downstream bonding group bonds together a set of downstream RF channels. It can only contain the RF channels from the same line card.

Associations between bonding groups and MAC Domains are automatically created. Associations occur when a bonding group channel set is found to be a subset of a MAC Domain Downstream Service Group (MD-DS-SG)

within a MAC Domain. The automatic associations will trigger the creation of an RCC that contains the bonding group's channel set within the MAC Domain.

Before you begin

Restrictions:

1. Included downstream channels must be a part of the same line card slot.
2. All the downstream channels must be from the integrated-cable controllers 0-3 or 4-7.

Procedure

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **interface wideband-Cable**

Example:

```
Router(config)#interface wideband-Cable 1/0/0:1
Router(config-if)#
```

Enter the wideband-cable interface configuration mode for specified wideband-cable interface.

Step 4 **cable bundle** *id*

Example:

```
Router(config-if)#cable bundle 1
```

Configures the cable bundle id for this wideband-cable interface. The configured cable bundle id must match the cable bundle id configured in associated MAC domains.

- *Bundle number*— cable bundle number. The valid range is 1-255.

Step 5 **cable rf-channels channel-list** *grouplist* **bandwidth-percent** *percentage-bandwidth*

Example:

```
Router(config-if)#cable rf-channel channel-list 1-3 bandwidth-percent 10
```

Configures the bandwidth allocation for specified channel-list and includes the channels in the downstream bonding group. Range for channel numbers are 0-127 (<first channel num-last channel num>).

- *grouplist*—Specify the range of downstream rf-channels.

Step 6 **cable rf-channels controller** *controller number* **channel-list** *grouplist* **bandwidth-percent** *percentage-bandwidth*

Example:

```
Router(config-if)#cable rf-channel controller 1 channel-list 1-3 bandwidth-percent 10
```

Configures the bandwidth allocation for specified channel-list on downstream controllers and includes the channels in the downstream bonding group. Range for channel numbers are 0-127.

- *controller number*—Downstream controller number. The valid numbers are 0-7.
- *grouplist*—Specify the range of downstream rf-channels.

Step 7 **end**

Example:

```
Router# end
```

Returns to privileged EXEC mode.

What to do next

Verify the Bonding Group Interfaces.

Verifying the Bonding Group Interfaces

Use **wideband-channel** option of the **show controllers integrated-Cable** command to display bonding group interfaces:

```
Router# show controllers integrated-cable 1/0/0 wideband-channel
Load for five secs: 2%/0%; one minute: 2%; five minutes: 2%
Time source is NTP, *17:45:51.964 PST Thu Jan 12 2012
WB                      BG    Primary
channel                 ID    BG
Wideband-Cable1/0/0:0   12289 Yes
Wideband-Cable1/0/0:1   12290 Yes
Wideband-Cable1/0/0:2   12291 Yes
Wideband-Cable1/0/0:3   12292 Yes
Wideband-Cable1/0/0:4   12293 Yes
Wideband-Cable1/0/0:5   12294 Yes
Wideband-Cable1/0/0:6   12295 Yes
Wideband-Cable1/0/0:7   12296 Yes
Wideband-Cable1/0/0:8   12297 Yes
Wideband-Cable1/0/0:9   12298 Yes
Wideband-Cable1/0/0:10  12299 Yes
Wideband-Cable1/0/0:11  12300 Yes
Wideband-Cable1/0/0:12  12301 Yes
Wideband-Cable1/0/0:13  12302 Yes
Wideband-Cable1/0/0:14  12303 Yes
Wideband-Cable1/0/0:15  12304 Yes
Wideband-Cable1/0/0:16  12305 Yes
```

```
Wideband-Cable1/0/0:17  12306 Yes
Wideband-Cable1/0/0:18  12307 Yes
Wideband-Cable1/0/0:19  12308 Yes
Wideband-Cable1/0/0:20  12309 Yes
Wideband-Cable1/0/0:21  12310 Yes
Wideband-Cable1/0/0:22  12311 Yes
Wideband-Cable1/0/0:23  12312 Yes
Wideband-Cable1/0/0:24  12313 Yes
Wideband-Cable1/0/0:25  12314 Yes
```

- To display the RF-channel mapping to wideband channels, use **mapping wb-channel** option.

```
Router# show controllers Integrated-Cable 1/0/0 mapping wb-channel 0
Ctrlr    WB       RF         WB %   WB Rem
1/0/0    0        1/0/0:0    40     1
                  1/0/0:1    40     1
```

- To display the downstream MAC Domain service groups, use the **dsbg-associations** option of the **show cable mac-domain** command.

```
Router# show cable mac-domain c1/0/0 dsbg-associations
Wi1/0/0:0       Wi1/0/0:1
```

Use the **show interface Wideband-Cable controller** command to verify the bonding group configurations. The State info table shows the downstream bonding group state information.

```
Router#show interface Wideband-Cable 1/0/0:0 controller
Wideband-Cable1/0/0:0 is up, line protocol is up
  Hardware is CMTS WB interface, address is c414.3c17.1dcb (bia c414.3c17.1dcb)
  MTU 1500 bytes, BW 150000 Kbit/sec, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation MCNS, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 112500 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 3 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out


BG controller details
Wi1/0/0:0 BGID: 12289
Member RFIDs:
  Config RFIDs: 12288-12291 Count: 4
```

```
  Active RFIDs: 12288-12291 Count: 4
Attribute mask: 0x80000000
-------------------------------------------
State info (DSBG if and its underlying states)
-------------------------------------------
DSBG IF state               : UP
DSBG Member RF chan states  : UP (4 out of 4 chans are UP)
DSBG HWID(FCID)             : 0x3800
*DSBG i/f Line State        : UP
-------------------------------------------
-----------------------------
DMP Resources
DMP handle     : 0x10000800
-----------------------------
DMP BG pool entry details
HW-id   BGid  BGSize  Enabled
-----------------------------
0     : 12289 4        1
-----------------------------
Bgid  BGecnt BGaddr Channels (1023 means invalid/Unused)
    0      0      0:   0    1    2    3 1023 1023 1023 1023
BG         Rate   Neg   Pos    LastTS  CurrCr  Pos
   0      25000 65535 65535         0       0   N

-----------------------------
RFID - JIB chan mapping for active RFIDs: [rfid:jib-chan-no]
[12288:0] [12289:1] [12290:2] [12291:3]

Router#
```

Upstream Bonding Group Configuration

Restrictions for Upstream Bonding Groups

- Upstream bonding groups are configured within the MAC Domain interface
- Upstream bonding groups consist of a set of upstream channels that are bonded together.
- A maximum of 16 upstream channels can be configured for each MAC Domain, which are divided into two groups:
  - Group 1: upstream channel 0-7
  - Group 2: upstream channel 8-15

  The **upstream bonding-group** should include all the upstream channels either from Group 1 or Group 2 only.

Configuring Upstream Bonding Groups

Before you begin

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** *slot/subslot/MD index* **Example:** `Router(config)# interface cable 1/0/0`	Enters MAC domain configuration mode. • *slot*—Specifies the chassis slot number of the interface line card. Valid values are 0-3 and 6-9 • *subslot*—Specifies the secondary slot number of the interface line card. Valid subslot is 0. • *MD index*—Specifies the MAC Domain index number. Valid values are 0-15.

Step 4	**cable upstream bonding-group**
	Example:
	`Router(config-if)#cable upstream bonding-group 7`
	`Router(config-upstream-bonding)#`
		Creates a static upstream bonding group on a MAC Domain.

Step 5	**upstream** **Example:** `Router(config-upstream-bonding)#upstream 7`	Add upstream channels to an upstream bonding group. A maximum of 16 upstream channels can be configured for each MAC Domain, which are divided into two groups: • Group 1: upstream channel 0-7 • Group 2: upstream channel 8-15 The **upstream bonding-group** should include all the upstream channels either from Group 1 or Group 2 only.
Step 6	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Router# **end**	

What to do next

Use the **show interface cable upstream bonding-group** command to display upstream bonding group information.

```
Router#show interface cable 1/0/0 upstream bonding-group
Load for five secs: 1%/0%; one minute: 2%; five minutes: 2%
Time source is NTP, *10:47:17.142 PST Thu Jan 12 2012

Cable1/0/0: Upstream Bonding Group 1
     0 packets input, 0 octets input
     Segments: 0 valid, 0 discarded, 0 lost
     Reserved Bandwidth Max : 0 bits/sec
     Reserved Bandwidth     : 0 bits/sec
     Available Bandwidth    : 46080000 bits/sec
     Total Service Flows On This Bonding Group: 0
Cable1/0/0: Upstream Bonding Group 2
     0 packets input, 0 octets input
     Segments: 0 valid, 0 discarded, 0 lost
     Reserved Bandwidth Max : 0 bits/sec
     Reserved Bandwidth     : 0 bits/sec
     Available Bandwidth    : 46080000 bits/sec
     Total Service Flows On This Bonding Group: 0
Cable1/0/0: Upstream Bonding Group 65536
     0 packets input, 0 octets input
     Segments: 0 valid, 0 discarded, 0 lost
     Reserved Bandwidth Max : 0 bits/sec
     Reserved Bandwidth     : 0 bits/sec
     Available Bandwidth    : 15360000 bits/sec
     Total Service Flows On This Bonding Group: 0
Cable1/0/0: Upstream Bonding Group 65537
     0 packets input, 0 octets input
     Segments: 0 valid, 0 discarded, 0 lost
     Reserved Bandwidth Max : 0 bits/sec
     Reserved Bandwidth     : 0 bits/sec
     Available Bandwidth    : 15360000 bits/sec
     Total Service Flows On This Bonding Group: 0

Router#
```

Verifying Upstream Bonding Groups

Use the **show cable upstream bonding-group** command.

```
Router#show interface cable 1/0/0 upstream bonding-group
Load for five secs: 1%/0%; one minute: 2%; five minutes: 2%
Time source is NTP, *10:47:17.142 PST Thu Jan 12 2012

Cable1/0/0: Upstream Bonding Group 1
     0 packets input, 0 octets input
     Segments: 0 valid, 0 discarded, 0 lost
     Reserved Bandwidth Max : 0 bits/sec
     Reserved Bandwidth     : 0 bits/sec
     Available Bandwidth    : 46080000 bits/sec
     Total Service Flows On This Bonding Group: 0
```

```
Cable1/0/0: Upstream Bonding Group 2
     0 packets input, 0 octets input
     Segments: 0 valid, 0 discarded, 0 lost
     Reserved Bandwidth Max : 0 bits/sec
     Reserved Bandwidth     : 0 bits/sec
     Available Bandwidth    : 46080000 bits/sec
     Total Service Flows On This Bonding Group: 0
Cable1/0/0: Upstream Bonding Group 65536
     0 packets input, 0 octets input
     Segments: 0 valid, 0 discarded, 0 lost
     Reserved Bandwidth Max : 0 bits/sec
     Reserved Bandwidth     : 0 bits/sec
     Available Bandwidth    : 15360000 bits/sec
     Total Service Flows On This Bonding Group: 0
Cable1/0/0: Upstream Bonding Group 65537
     0 packets input, 0 octets input
     Segments: 0 valid, 0 discarded, 0 lost
     Reserved Bandwidth Max : 0 bits/sec
     Reserved Bandwidth     : 0 bits/sec
     Available Bandwidth    : 15360000 bits/sec
     Total Service Flows On This Bonding Group: 0

Router#
```

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

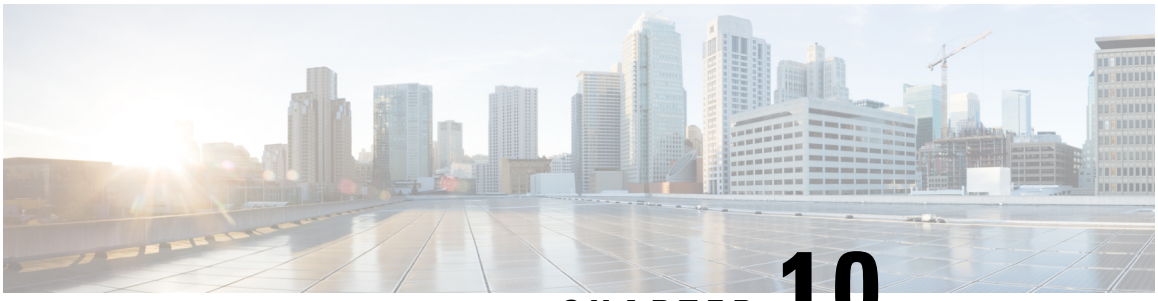
Feature Information for DOCSIS Interface and Fiber Node Configuration

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 17: Feature Information for DOCSIS Interface and Fiber Node Configuration

Feature Name	Releases	Feature Information
DOCSIS Interface and Fiber Node Configuration	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the cisco cBR Series Converged Broadband Routers.



CHAPTER 10

Service Group Based Configuration of the Cisco cBR Router

- Service Group Profile Based Configuration, on page 165
- Service Profile Configuration for 16x8 with One MAC Domain, on page 167
- Service Profile Configuration for 16x8 with Two MAC Domains, on page 169

Service Group Profile Based Configuration

The DOCSIS Interface and Fiber Node Configuration guide describes the interface and fiber node configurations that are required to operationalize the Cisco cBR router. The steps described to accomplish the tasks involved in such configuration are complex.

To simplify and speed up the process of configuring the physical and logical interfaces required to deploy the Cisco cBR router quickly, a service group (SG) profile based approach is adopted. This document describes the simplified SG profile approach.

This approach provides the following advantages:

- Improves and simplifies the deployment of Cisco cBR router.
- Improves and simplifies the configuration of Cisco cBR Router by eliminating duplicate configurations.
- Improves and simplifies troubleshooting of Cisco cBR router.
- Supports faster Converged Cable Access Platform (CCAP) provisioning by using common and quick replication across nodes and regions.

To configure the interfaces and quickly operationalize the Cisco cBR router, a set of common profiles are created and are created and configured into global service group profiles. These global service group profiles may be applied to fiber node interfaces along with a mapping of the service group interfaces to the physical interfaces.

Common profiles are profiles which contain configurations for common service group (SG) interfaces like MAC domain, wideband-cable, and primary downstream

The common profiles and the global SG profiles may be independent of the topology of the network in which they are applied. The SG interface to physical interface mapping defines the behavior of the profiles in the topology that the SG profiles have been applied to.

Limitations:

- A common profile cannot be deleted if it is associated with the fiber nodes. It can be modified by entering the profile configuration mode using the **cable profile** command.
- A service-group profile cannot be deleted or modified if it is associated with the fiber nodes.
- To associate a new global SG profile to a fiber node, dissociate the currently associated global SG profile from the fiber node.
- To configure for a unique topology, use the full configuration approach provided in DOCSIS Interface and Fiber Node Configuration guide.
- When the secondary line card is in active mode, the following limitations apply:
  - A MAC domain, Wideband-Cable interface, downstream channel, and SG profile cannot be modified when they are associated with the fiber nodes.
  - SG profile cannot be dissociated from the fiber nodes.
  - A MAC domain, Wideband-Cable interface, downstream channel, and SG profile can be created but cannot be associated to the fiber node.
  - To enable the SG operation simplification feature, auto-reset feature must be enabled using the **cable wideband auto-reset** command.

Broadly, the following steps define and deploy the common profiles and the SG profiles:

1. Define and configure the Common profiles: The SG interface profiles or the common profiles contain configuration parameters common to a group of similar interfaces. For example, a profile may contain specific configuration parameters shared by multiple Wideband-Cable interfaces, associated across multiple line cards. A common profile is configured at the global or chassis level. All interfaces associated with a profile will inherit the configuration in the profile. Any common profile may be associated with any global SG profile. Use the **cable profile** *profile-type profile-name* command to define the following common profiles:
   - MAC Domain (MD) profile
   - Primary Downstream channel (DS) profile
   - Wideband-cable Interface (WB) profile
   - Global Service Group (SG) profile
2. Complete the following configurations to the fiber node interface using the **cable fiber-node** command:
   - Define the downstream and upstream ports. Map the appropriate interface-cable using the **downstream interface-cable** command. Map the appropriate upstream-cable interface using the **upstream upstream-cable** command.
   - Use the **downstream sg-channel** command to map the logical downstream SG channels to the physical RF channels and the **upstream sg-channel** command to map the logical upstream SG channels to the physical upstream channels.
   - Use the **service-group profile** to associate the global service group profile to the fiber node.

See the use case scenarios for configurations and examples.

Service Profile Configuration for 16x8 with One MAC Domain

This section describes the service group based configurations for a 16x8 service group with one MAC domain.

Procedure

Step 1 **cable profile** *profile-type profile-name*

For MAC Domain profile, specify *profile-type* as **mac-domain**.

```
Router(config)#cable profile mac-domain MD1
Router(config-profile-md)#cable dynamic-secret mark
Router(config-profile-md)#cable shared-secret 0 cisco
Router(config-profile-md)#cable ip-init ipv4
Router(config-profile-md)#cable mtc-mode
Router(config-profile-md)#cable mrc-mode
Router(config-profile-md)#cable privacy mandatory
Router(config-profile-md)#cable privacy bpi-plus-policy
```

For Primary downstream profile, specify *profile-type* as **downstream**.

```
Router(config)#cable profile downstream DS1
Router(config-profile-ds)#cable rf-bandwidth-percent 20
Router(config-profile-ds)#cable attribute-mask 0x80000000
```

For wideband-cable interface profile, specify *profile-type* as **wideband-interface**.

```
Router(config)#cable profile wideband-interface BG1
Router(config-profile-wb)#cable downstream attribute-mask 0x80000000
```

For global service group profile, specify *profile-type* as service-group. In the service group profile, configure the cable bundle associated, mac-domain profile, and the wideband interface profile

```
Router(config)#cable profile service-group SG-16x8-1_1
Router(config-profile-sg)#cable bundle 71
Router(config-profile-sg)#mac-domain 0 profile md1
Router(config-profile-sg-md)#downstream sg-channel 0-7  profile ds1 upstream 0-3
Router(config-profile-sg-md)#upstream 0 sg-channel 0
Router(config-profile-sg-md)#upstream 1 sg-channel 1
Router(config-profile-sg-md)#upstream 2 sg-channel 2
Router(config-profile-sg-md)#upstream 3 sg-channel 3
Router(config-profile-sg-md)#upstream 4 sg-channel 4
Router(config-profile-sg-md)#upstream 5 sg-channel 5
Router(config-profile-sg-md)#upstream 6 sg-channel 6
Router(config-profile-sg-md)#upstream 7 sg-channel 7
Router(config-profile-sg-md)#us-bonding-group 1
Router(config-profile-sg-md-usbg)#upstream 0
Router(config-profile-sg-md-usbg)#upstream 1
Router(config-profile-sg-md-usbg)#upstream 2
Router(config-profile-sg-md-usbg)#upstream 3
Router(config-profile-sg-md-usbg)#upstream 4
Router(config-profile-sg-md-usbg)#upstream 5
Router(config-profile-sg-md-usbg)#upstream 6
```

```
Router(config-profile-sg-md-usbg)#upstream 7
Router(config-profile-sg-md-usbg)#attributes 8000000
Router(config-profile-sg-md-usbg)#exit
Router(config-profile-sg-md)#exit
Router(config-profile-sg)#

Router(config-profile-sg)#wideband-interface 1 profile BG1
Router(config-profile-sg-bg)#downstream sg-channel 0 15 rf-bandwidth-percent 10
Router(config-profile-sg-bg)#end
Router#
```

Step 2 **cable fiber-node**

Enter fiber-node configuration mode. Configure the following in fiber-node configuration mode:

- downstream port
- upstream port
- downstream sg-channels to rf-channels mapping
- upstream sg-channels to us-channels mapping
- global service group mapping

```
Router(config)#cable fiber-node 1
Router(config-fiber-node)#downstream integrated-cable 3/0/0
Router(config-fiber-node)#upstream upstream-cable 3/0/0
Router(config-fiber-node)#downstream sg-channel 0 15 integrated-cable 3/0/0 rf-channel 0
15
Router(config-fiber-node)#upstream sg-channel 0 7 upstream-cable 3/0/0 us-channel 0 7
Router(config-fiber-node)#service-group profile SG-16X8-1_1
```

What to do next

Use the **show cable fiber-node** [*id*] **mapping** and the **show cable fiber-node** [*id*] **derived** commands to check the configuration of the interfaces.

```
Router#show cable fiber-node 1 mapping
Fiber-node 1:
Upstream:
Sg chan       Us-chan        Op state
0             3/0/0 0        Up
1             3/0/0 1        Up
2             3/0/0 2        Up
3             3/0/0 3        Up
4             3/0/0 4        Up
5             3/0/0 5        Up
6             3/0/0 6        Up
7             3/0/0 7        Up
Downstream:
Sg chan       Ds-rf-chan     Op state
0             3/0/0:0        Up
1             3/0/0:1        Up
2             3/0/0:2        Up
3             3/0/0:3        Up
```

```
4             3/0/0:4        Up
5             3/0/0:5        Up
6             3/0/0:6        Up
7             3/0/0:7        Up
8             3/0/0:8        Up
9             3/0/0:9        Up
10            3/0/0:10        Up
11            3/0/0:11        Up
12            3/0/0:12        Up
13            3/0/0:13        Up
14            3/0/0:14        Up
15            3/0/0:15        Up
```

```
Router#show cable fiber-node 1 derived
Fiber-node 1:
              interface                   Assoc successed
mac-domain 0   Cable3/0/0                  Y
Wideband 1     Wideband-Cable3/0/0:0       Y
Router#
```

Use the **show cable mac-domain fiber-node** command to verify the interface associations.

```
Router#show cable mac-domain fiber-node 1 md 0 downstream-service-group
Cable    MD-DS-SG              RF
IF       Id        Resource    Chan     Primary Chan
C3/0/0
Router#
```

```
Router#show cable mac-domain fiber-node 1 md 0 upstream-service-group
Cable MD 3/0/0
Router#
```

Service Profile Configuration for 16x8 with Two MAC Domains

This section describes the service group based configurations for a 16x8 service group with two MAC domains, split downstream and overlaid upstream channels.

Procedure

Step 1 **cable profile** *profile-type profile-name*

For MAC Domain profile, specify *profile-type* as **mac-domain**.

```
Router(config)#cable profile mac-domain MD1
Router(config-profile-md)#cable dynamic-secret mark
Router(config-profile-md)#cable shared-secret 0 cisco
Router(config-profile-md)#cable ip-init ipv4
Router(config-profile-md)#cable mtc-mode
Router(config-profile-md)#cable mrc-mode
Router(config-profile-md)#cable privacy mandatory
Router(config-profile-md)#cable privacy bpi-plus-policy
```

For Primary downstream profile, specify *profile-type* as **downstream**.

```
Router(config)#cable profile downstream DS1
Router(config-profile-ds)#cable rf-bandwidth-percent 20
Router(config-profile-ds)#cable attribute-mask 0x80000000
```

For wideband-cable interface profile, specify *profile-type* as **wideband-interface**.

```
Router(config)#cable profile wideband-interface BG1
Router(config-profile-wb)#cable downstream attribute-mask  0x80000000
```

For global service group profile, specify *profile-type* as service-group. In the service group profile, configure the cable bundle associated, mac-domain profile, and the wideband interface profile

```
Router(config)#cable profile service-group SG-16x4-1_2
Router(config-profile-sg)#cable bundle 71
Router(config-profile-sg)#mac-domain 0 profile md1
Router(config-profile-sg-md)#downstream sg-channel 0-15  profile ds1 upstream 0-3
Router(config-profile-sg-md)#upstream 0 sg-channel 0
Router(config-profile-sg-md)#upstream 1 sg-channel 1
Router(config-profile-sg-md)#upstream 2 sg-channel 2
Router(config-profile-sg-md)#upstream 3 sg-channel 3
Router(config-profile-sg-md)#us-bonding-group 1
Router(config-profile-sg-md-usbg)#upstream 0
Router(config-profile-sg-md-usbg)#upstream 1
Router(config-profile-sg-md-usbg)#upstream 2
Router(config-profile-sg-md-usbg)#upstream 3
Router(config-profile-sg-md-usbg)#exit
Router(config-profile-sg-md)#exit
Router(config-profile-sg)#

Router(config-profile-sg)#wideband-interface 1 profile BG1
Router(config-profile-sg-bg)#downstream sg-channel 0 7 rf-bandwidth-percent 10
Router(config-profile-sg-bg)#exit
Router(config-profile-sg)#wideband-interface 2 profile BG1
Router(config-profile-sg-bg)#downstream sg-channel 8 15 rf-bandwidth-percent 10
Router#
```

Step 2 cable fiber-node

Enter fiber-node configuration mode. Configure the following in fiber-node configuration mode:

- downstream port
- upstream port
- downstream sg-channels to rf-channels mapping
- upstream sg-channels to us-channels mapping
- global service group mapping

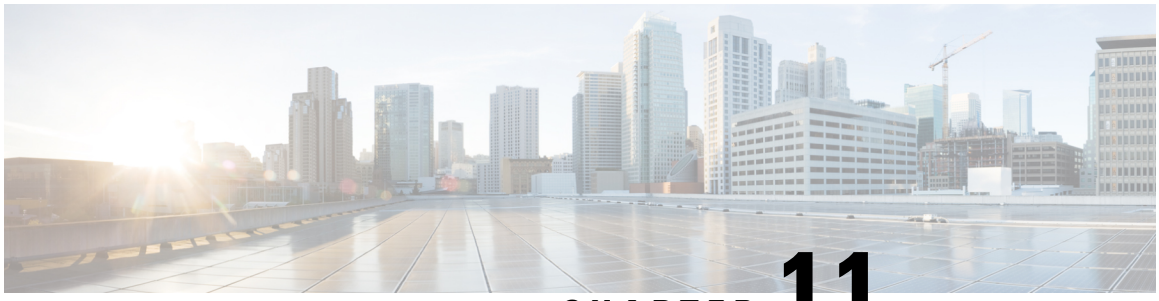
```
Router(config)#cable fiber-node 1
Router(config-fiber-node)#downstream integrated-cable 3/0/0
Router(config-fiber-node)#upstream upstream-cable 3/0/0
Router(config-fiber-node)#downstream sg-channel 0 7 integrated-cable 3/0/0 rf-channel 0 15
Router(config-fiber-node)#upstream sg-channel 0 3 upstream-cable 3/0/1 us-channel 0 3
```

```
Router(config-fiber-node)#service-group profile SG-16X4-1_2
Router(config-fiber-node)#exit
Router(config)#
```

```
Router(config)#cable fiber-node 2
Router(config-fiber-node)#downstream integrated-cable 3/0/0
Router(config-fiber-node)#upstream upstream-cable 3/0/1
Router(config-fiber-node)#downstream sg-channel 0 7 integrated-cable 3/0/0 rf-channel 8 15
Router(config-fiber-node)#downstream sg-channel 8 15 integrated-cable 3/0/0 rf-channel 0 7
Router(config-fiber-node)#upstream sg-channel 0 3 upstream-cable 3/0/1 us-channel 0 3
Router(config-fiber-node)#service-group profile SG-16X4-1_2
Router(config-fiber-node)#exit
Router(config)#
```

What to do next

Use the **show cable fiber-node** [*id*] **mapping** and the **show cable fiber-node** [*id*] **derived** commands to check the configuration of the interfaces.



CHAPTER 11

DOCSIS Load Balancing Groups

First Published: April 11, 2015

Support for the restricted load balancing group (RLBG)/general load balancing group (GLBG) is based on DOCSIS 3.0 specifications.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 173
- Prerequisites for DOCSIS Load Balancing Groups, on page 174
- Restrictions for DOCSIS Load Balancing Groups, on page 174
- Information About DOCSIS Load Balancing Groups, on page 175
- How to Configure DOCSIS Load Balancing Groups, on page 182
- Configuration Examples for DOCSIS Load Balancing Groups, on page 193
- Verifying DOCSIS Load Balancing Groups, on page 194
- Additional References, on page 199
- Feature Information for DOCSIS Load Balancing Groups , on page 199

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 18: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for DOCSIS Load Balancing Groups

DOCSIS Load Balancing Groups including Restricted/General Load Balancing groups with Downstream Dynamic Load Balancing feature has the following prerequisites:

- A RLBG and a DOCSIS 2.0 GLBG should have a load balancing group (LBG) ID.
- A LBG should have a default policy ID.
- During registration, a cable modem (CM) that has been assigned to a LBG must also be assigned a policy ID and priority, through Simple Network Management Protocol (SNMP), the cable modem configuration file, or Cisco Cable Modem Termination System (CMTS) configuration.
- The cable modem must send service type identifier (STID), service class name, and DOCSIS version and capability type/length/value (TLV) settings to the Cisco CMTS for registration if the fields are used by general tagging.

Restrictions for DOCSIS Load Balancing Groups

The DOCSIS Load Balancing Groups (LBG) including RLBG/GLBG Support with DLB Support feature has the following restrictions:

- A maximum of 256 DOCSIS policies and 256 rules per chassis are supported.
- Cross-line card (LC) configuration or moving of cable modems is not supported.

- When deployed with channel restriction features, if the target upstream channel attribute masks are against that of the cable modem, then the cable modem on the higher load upstream will not be load balanced, as the current load balancing moves cable modems only to the target upstream. However, cable modems that do not have an attribute mask can still be load balanced. You should consider the following while deploying the load balancing groups: the target upstream will always be the upstream that has the lowest load. If some other upstreams have the same load, the upstream with the lowest index will be chosen as the target upstream.
- We recommend all LBGs that share channels must use the same LB method.

The DOCSIS LBG with RLBG/GLBG Support and DLB Support feature have the following cross functional restrictions:

- Cable modems operating in the multiple transmit channel (MTC) mode do not register for a RLBG assignment, even if their configuration file contains relevant TLVs, such as STID and LBG ID. However, cable modems operating in the multiple receive channel (MRC) can register for a RLBG assignment.
- The Cisco CMTS can parse a specific TLV encoded in cable modem configuration file, and prohibit any DCC operation on the cable modems.
- DOCSIS MAC domain downstream service group (MD-DS-SG) channels in MDD messages are incorrect when a combination of channels from multiple line card types are placed in the same fiber node.

In a complex fiber node setup, with channels from more than one line card, or downstream channels of one MAC domain in more than one fiber node, some modems may not come w-online (wideband online). If a MAC domain has more than one MD-DS-SG, the MDD will contain more than one MD-DS-SG and cause the modem to perform downstream ambiguity resolution. When the modem analyzes the downstream channels from the other line card, it will not see MDD packets and disqualify the channel and the MD-DS-SG. The modem then sends a requested MD-DS-SG of 0 to the CMTS implying it will not participate in a bonding group.

Use the **show cable mac-domain downstream-service-group** command to see the channels in the same MD-DS-SG.

The DOCSIS LBG with RLBG/GLBG Support and DLB Support feature have the following scaling limitations:

- The total number of RLBGs and DOCSIS 2.0 GLBGs cannot exceed 256.
- The total number of tags in a Cisco CMTS cannot exceed 256.
- The total number of DOCSIS 3.0 GLBGs is bounded by free memory.
- A cable modem reset occurs if a CM moves from one cable interface to another because DCC init-tech 0 resets a cable modem during a LB move. A cable modem also resets if the two cable interfaces have been configured with a mismatched **cable ip-init** command.

Information About DOCSIS Load Balancing Groups

The DOCSIS 2.0 “Autonomous Load Balancing” specification is CM-centric, allowing a channel (US or DS) to be part of multiple RLBGs. Therefore, with the DOCSIS 2.0 specifications, you can decide on which channel the CM can be load balanced.

To configure the Restricted/General Load Balancing and Narrowband Dynamic Bandwidth Sharing with Downstream Dynamic Load Balancing feature, you should understand the following concepts:

Service-Based Load Balancing

Using the DOCSIS 3.0 modem-based load balancing specifications, you can manage the LB activity on a per-modem basis as follows:

1. Modem to RLBG association through STID
2. Modem to RLBG association through LBG ID
3. Per-modem LB policy assignment
4. Per-modem LB priority assignment
5. Per-modem channel restriction

Implementing the DOCSIS 3.0 modem-based LB specifications enables the Cisco CMTS to provide an advanced service-based LB. The service-based LB can be used to alleviate the burden for the modem-based provisioning and provide the operator an ability to selectively control LB activity based on modem service type. For example, for LB purposes modems can be classified based on:

- Device type
- DOCSIS version
- Service class

The results of the classification can then be used to selectively control the modem LB activity by mapping the modem to the following settings:

- LBG
- Policy

With the service-based LB enabled, existing service-based cable modem segregation features and channel restriction become special cases and can be handled within the same LB framework.

Functionality

The Cisco CMTS functions in the following ways for general tagging and service-based LB:

- The Cisco CMTS can classify some modems with user-defined modem classifiers using the STID, service class name, DOCSIS version and capability TLVs and MAC Organization Unique Identifier (OUI).
- Each modem classifier has a unique tag. The Cisco CMTS allows each modem to carry one tag. When multiple tags match one cable modem, the tag that has the least index gets applied on the cable modems.
- The Cisco CMTS classifies a CM and assigns a tag, and if a RLBG with that tag is configured, the CM gets assigned to that RLBG.
- The Cisco CMTS can match multiple tags to a RLBG and a DOCSIS policy.
- On the Cisco CMTS, a user can configure whether the general tagging overrides the RLBG or DOCSIS policy assignment using TLVs in the CM configuration file and SNMP when a conflict occurs.
- When doing autonomous LB, the Cisco CMTS ensures that the target channels are available to a specific CM with regard to admission control, the SF attribute masks, and CM attribute masks.
- The user can configure the number of times that a DCC fails a CM before the CM is removed from dynamic LB on the Cisco CMTS.
- The user can configure DCC initialization techniques or whether to use Upstream Channel Change (UCC) for a LBG or for a particular source and target pair on the Cisco CMTS. However, DCC is not issued to cable modems provisioned in DOCSIS 1.0 mode. By default, the UCC for a LBG is not configured and therefore, all channel changes are done through DCC.
- The Cisco CMTS supports LB on at least one logical channel on a physical US channel that has multiple logical US channels.

- As per the DOCSIS 3.0 specifications, a lower load balancing priority indicates a higher likelihood that a CM will be moved due to load balancing operations.
- You can create a policy to set the lower bandwidth for CMs. the LBG can only move cable modems with throughput that is above the threshold.

Compatibility

Both downstream and upstream autonomous load balancing is supported for single channel cable modems.

RLBG/GLBG Assignment

The user can configure one or more service type IDs for each RLBG. The user can also configure the Cisco CMTS, using CLI or SNMP, to restrict a particular cable modem to a certain STID and RLBG ID. However, if such a configuration is made, both the STID and RLBG ID in the configuration file are ignored by the Cisco CMTS.

When the STID is configured by CLI or SNMP or the STID is present in the cable modem configuration file, the Cisco CMTS selects an upstream and downstream channel, which offers the signaled service type, from a RLBG, if such channels exist. However, if an upstream and downstream channel do not exist that provide the signaled service type the Cisco CMTS assigns an upstream and downstream channel that does not offer the signaled service type.

When the LBG ID is configured by CLI or SNMP or the LBG ID is present in the cable modem configuration file, the Cisco CMTS examines the available choices for upstream and downstream channels and, if they include a channel pair associated with the signaled LBG, the Cisco CMTS assigns the cable modem to the signaled LBG. If these conditions are not met, the Cisco CMTS disregards the LBG ID.

If there are multiple upstream and downstream channels available that meet the requirements of the STID, if present, and the LBG ID, if present, the Cisco CMTS selects an upstream and/or downstream channel that meet the cable modem required and forbidden attribute masks requested in the configuration file. If upstream and downstream channels are not available that meet these criteria, the Cisco CMTS can disregard the cable modem attribute masks and select an alternative upstream and/or downstream channel.

In determining a target channel pair for a cable modem during registration time, the Cisco CMTS tries to find the target channel pair that can actually reach the cable modem by checking the current channel pair, the MD-DS-SG-ID (Media Access Control Domain Downstream Service Group Identifier) of cable modem (CM-DS-SG-ID) and the MD-US-SG-ID (Media Access Control Domain Upstream Service Group Identifier) of cable modem (CM-US-SG-ID), if present, and fiber node (FN) configurations. If the target channel pair is available to the cable modem and is different from the current channel pair, the Cisco CMTS is required to move the CM by means of DCC technique 0 or downstream frequency override (DFO).

When the Cisco CMTS identifies multiple candidate RLBGs for a CM, but cannot determine which fiber node configuration the cable modem is actually wired to, or cannot determine if the wired RLBG is unusable (when interfaces in the load balance group are disabled or in an administratively down state), the Cisco CMTS assigns the cable modem to the RLBG with the lowest group index. This assignment causes the Cisco CMTS to attempt to move the cable modem to interfaces it is not physically connected to, resulting in service outages for the CM.

The Cisco CMTS enforces fiber node checking during RLBG assignment.

The Cisco CMTS follows the following RLBG assignment rules:

- If there is no fiber node configuration, there is no change in the candidate RLBG list. However, if the fiber node is configured, the fiber node must be configured correctly to reflect the real fiber node connection.

- If the cable modem is inside a fiber node, only those RLBGs that are inside that fiber node are selected.
- If the cable modem is not inside any fiber node, that is, the fiber node configuration does not cover all the channels, only those RLBGs that are not inside any fiber node are selected.
- If an RLBG spans across multiple fiber nodes, it is not considered to be inside any fiber node.
- If no candidate RLBG is found, cable modems are assigned to the GLBG, if the GLBG exists.

Channel Assignment

For cable modems operating in MRC mode, the registration request message can have multiple TLVs to influence the selection of upstream and downstream channels that the Cisco CMTS assigns. To avoid conflicts between the multiple TLVs, the Cisco CMTS follows the precedence order defined below:

1. TLV 56—Channel Assignment
2. TLV 43.11—Service Type Identifier
3. TLV 43.3—Load Balancing Group ID
4. TLVs 24/25.31-33—Service Flow Attribute Masks
5. TLV 43.9—CM Attribute Masks

The Cisco CMTS must follow this TLV precedence order for cable modems not operating in MRC mode:

1. TLV 43.11—Service Type Identifier
2. TLV 43.3—Load Balancing Group ID
3. TLV 43.9—CM Attribute Masks
4. TLVs 24/25.31-33—Service Flow Attribute Masks

Note When a target for the new receive channel configuration (RCC) and Transmit channel configuration (TCC) is selected, ensure that the service level for cable modems is not decreased. Target total RCCs and TCCs must not be less than the source total RCCs and TCCs so that cable modems can keep their service level unchanged. This may cause some unbalanced results when high capacity cable modems come online, later releases..

The Cisco CMTS also considers the DOCSIS 3.0 cable modem capabilities defined in the registration request message and assigns the maximum number of channels that the CM requests.

The tables below define the load balancing matrix for RLBG and GLBG assignment:

Table 19: RLBG Assignment for DOCSIS Cable Modems

Operational Mode	MAC Version				
	DOCSIS 3.0 CM	**DOCSIS 2.x CM**	**DOCSIS 2.0 CM**	**DOCSIS 1.1 CM**	**DOCSIS 1.0 CM**
Non-MRC mode (online)	Assigned	Assigned	Assigned	Assigned	Assigned
MRC mode only (w-online)	Assigned	Assigned	Assigned	NA	NA

Operational Mode	MAC Version				
MRC/MTC mode (UB-online)	Assigned	NA	NA	NA	NA
	DOCSIS 3.0 cable modems are assigned to the DOCSIS 3.0 RLBG	NA	NA	NA	NA

Table 20: GLBG Assignment for DOCSIS Cable Modems

Operational Mode	MAC Version				
	DOCSIS 3.0 CM	**DOCSIS 2.x CM**	**DOCSIS 2.0 CM**	**DOCSIS 1.1 CM**	**DOCSIS 1.0 CM**
Non-MRC mode (online)	Assigned to the DOCSIS 2.0 GLBG without MD-DS-SG-ID/MD-US-SG-ID				
	Assigned to the DOCSIS 3.0 GLBG with MD-DS-SG-ID/MD-US-SG-ID		NA	NA	NA
MRC mode only (w-online)	Assigned to the DOCSIS 2.0 GLBG without MD-DS-SG-ID/MD-US-SG-ID				
	Assigned to the DOCSIS 3.0 GLBG with MD-DS-SG-ID/MD-US-SG-ID		NA	NA	NA
MRC/MTC mode (UB-online)	Assigned	NA	NA	NA	NA
	DOCSIS 3.0 cable modems are assigned to the DOCSIS 3.0 GLBG	NA	NA	NA	NA

The tables below give a snapshot view of the load balancing methods and the operations used to "move" bonded and non-bonded CMs.

Table 21: Load Balancing Method to Move Bonded and Non-bonded cable modems

Modem Mode	Dynamic Service Charge (Initialization Technique)	
	Within MAC Domain	**Across MAC Domains**
DOCSIS 3.0 cable modems in MTC mode	NA	DCC initialization technique 0
DOCSIS 3.0/DOCSIS 2.x cable modems in MRC-only mode	DCC initialization technique 0 **Note** CM with primary DS outside RLBG moves inside RLBG with DOCSIS 2.0 LB.	DCC initialization technique 0
DOCSIS 3.0 cable modems in MRC-only mode	DCC **Note** CM outside RLBG moves inside RLBG with DOCSIS 2.0 LB.	DCC initialization technique 0

Modem Mode	Dynamic Service Charge (Initialization Technique)	
DOCSIS 2.x cable modems in MRC-only mode	DCC/UCC **Note** CM outside RLBG moves inside RLBG with DOCSIS 2.0 LB.	DCC initialization technique 0
DOCSIS 2.0 /DOCSIS 1.1 cable modems in NB mode	DCC **Note** CM outside RLBG moves inside RLBG with DOCSIS 2.0 LB.	DCC initialization technique 0
	UCC **Note** CM outside RLBG moves inside RLBG with DOCSIS 2.0 LB.	UCC
DOCSIS 1.0 in NB mode	Force reinitialize CM **Note** CM outside RLBG moves inside RLBG with DOCSIS 2.0 LB.	Force reinitialize CM
	UCC **Note** CM outside RLBG moves inside RLBG with DOCSIS 2.0 LB.	UCC

Table 22: Using DCC/DBC to Load Balance Bonded and Non-bonded Cable Modems

Channel	CM in MRC, non-MTC Mode	DOCSIS 1.1/DOCSIS 2.0 cable modems with Single US/DS	DOCSIS 1.0 cable modems with Single US/DS
Upstream (US)	DCC	DCC	UCC
Downstream (DS)	NA (within the same MAC domain)	DCC (within the same MAC domain).	Force reinitialize CM
	DCC with initialization technique 0 when moving cable modems across MAC domains.	DCC with initialization technique 0 when moving cable modems across MAC domains.	Force reinitialize CM

Error Handling of Channel Assignment

This restriction is modified. As long as the interface state of the channels is not "administratively down", all channels are available for LBG assignment. For other load balancing operations, such as moving modems using DCC, UCC, or DBC, the interface state of the channels should be in "initial", "up", "suspicious", or "testing" states.

The following conditions apply when an LBG is disabled:

- cable modems that match all load balancing criteria can be assigned to an LBG.
- cable modem moves for load balancing are disabled, but cable modem moves from outside of the LBG to inside of the LBG are allowed.

Upstream Load Balancing for DOCSIS 3.0 Cable Modems in Single Upstream Mode

The upstream load balancing functionality enables the Cisco CMTS router to effectively handle upstream traffic for wideband and narrowband cable modems that are in single upstream mode. Single upstream mode (Mx1) means that the modems cannot send upstream traffic on multiple upstream channels. In the event of traffic overload on a single upstream channel of a wideband or narrowband cable modem, the Cisco CMTS router automatically moves the cable modem to another upstream channel in the same load balancing group.

Note A cable modem operating in single upstream mode is assigned to a load balancing group based on the primary channel of the modem. A cable modem in single upstream mode can support multiple receive channel (MRC) mode or narrowband mode. However, a cable modem in single upstream mode cannot support multiple transmit channel mode (MTC).

Auto-generate DOCSIS 2.0 GLBG

Cisco CMTS does not automatically implement DOCSIS 2.0 GLBG. DOCSIS 2.0 GLBG is configured manually after a new fiber node - MAC domain (FN-MD) pair is added.

This enhancement to automatically generate DOCSIS 2.0 GLBG after adding a new FN-MD pair and resolving a new combination of MAC domain, cable modem, and service group (MD-CM-SG). This enhancement is implemented through a new command **cable load-balance d20 GLBG auto-generate**. The command has options to renew and update DOCSIS 2.0 GLBGs for a fiber node configuration.

Independent Upstream/Downstream Throughput Rules

Currently, during upstream or downstream load balancing, to move modems in load balancing operations, Cisco CMTS applies the DOCSIS policy throughput rules to both upstream and downstream throughput to upstream or downstream load balancing operations. In other words, for downstream load balancing, both upstream and downstream sets of rules are applied and similarly for upstream load balancing both set of rules are applied. This prevents movement of modems with low upstream or high downstream throughput and high upstream or low downstream throughput.

Upstream or downstream throughput rules are checked independently to corresponding upstream or downstream load balancing operations. During upstream load balancing, only upstream throughput rules are checked, and during downstream load balancing, only downstream throughput rules are checked.

The following important points are implemented for independent upstream/downstream throughput rules:

- If a load balancing operation involves a change only in the downstream channel of a cable modem without any change to the upstream channel, then only the downstream lower boundary rules are checked.
- If a load balancing operation involves a change only in the upstream channel of a cable modem without any change to the downstream channel, then only the upstream lower boundary rules are checked.
- If a load balancing operation involves a change in both the upstream and downstream channels of a cable modem, then the modem rule check must pass all the rules for that (upstream or downstream) load balancing.
- If the load balancing policy configured is **pure-ds-load**, then only the downstream rules are checked.

- If the load balancing policy configured is **us-across-ds** or both **us-across-ds** and **pure-ds-load**, then two types of target interfaces occur as follows:
  - Local interface—where the cable modem shares the upstream with the source. Only downstream load balancing operation occurs.
  - Remote interface—where the the cable modem does not share the upstream with the source. The upstream/downstream load balancing is triggered by upstream load.

If the load balancing policy configured is neither us-across-ds nor pure-ds-load, then the load balancing is done based on Mac domain load.

How to Configure DOCSIS Load Balancing Groups

The Restricted/General Load Balancing and Narrowband Dynamic Bandwidth Sharing with Downstream Dynamic Load Balancing feature can be configured as follows:

- A user can configure a DOCSIS 2.0 general load balancing group (GLBG) on the Cisco CMTS according to DOCSIS specification. The Cisco CMTS creates a DOCSIS 3.0 GLBG for each Media Access Control Domain Cable Modem Service Group (MD-CM-SG) automatically and checks whether the GLBG contains both upstream and downstream channels.
- A cable modem that is not provisioned to any RLBG and cannot resolve its MD-CM-SG gets assigned to a DOCSIS 2.0 GLBG. However, if the cable modem resolves its MD-CM-SG, it gets assigned to a DOCSIS 3.0 GLBG.
- A user can configure RLBGs and any upstream or downstream channel into multiple RLBGs on the Cisco CMTS. The Cisco CMTS checks whether a RLBG contains both upstream and downstream channels. A RLBG can cross multiple MDs.
- A backward compatibility with existing Cisco LB schemes is maintained. The users can switch between the old and new DOCSIS 3.0 compliant LB schemes.

Note When the Cisco IOS system is upgraded, if the docsis-policy configuration of the DOCSIS load balancing groups, is missing in the output of the **show running-config** command, apply the docsis-policy to the DOCSIS load balancing groups using the **docsis-policy** *policy-id* command again.

The following sections describe how to create and configure DOCSIS load balancing groups to enable DOCSIS load balancing on the Cisco CMTS:

Configuring DOCSIS 3.0 and 2.0 RLBG and DOCSIS 2.0 GLBG

This section describes how to create and configure a DOCSIS load balancing group. There is a separate configuration mode for a DOCSIS load balancing group that is different from the legacy load balancing group.

Note UGS/PCMM policy and threshold do not apply on DOCSIS 3.0 LB.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable load-balance docsis-enable** **Example:** `Router(config)# cable load-balance docsis-enable`	Enables DOCSIS load balancing on the Cisco CMTS.
Step 4	**cable load-balance docsis-group** *docsis-group-id* **Example:** `Router(config)# cable load-balance docsis-group 1`	Creates a DOCSIS load balance group on the Cisco CMTS, with the following parameter: The router enters DOCSIS load balancing group configuration mode.
Step 5	**init-tech-list** *tech-list* [**ucc**] **Example:** `Router(config-lb-group)# init-tech-list 1 ucc`	Sets the DCC initialization techniques that the Cisco CMTS can use to load balance cable modems.
Step 6	**downstream** {**Cable** {*slot/subslot/port* | *slot/port*} | **Integrated-Cable** {*slot/subslot/bay* | *slot/port*} {**rf-channel** *group list*}|{| *slot/port*} {**rf-channel** *group list*}} **Example:** `Router(config-lb-group)# downstream integrated-Cable 5/0/0 rf-channel 2`	Sets the downstream RF channels.
Step 7	**upstream Cable** {*slot/subslot/port* | *slot/port*} *upstream-list* **Example:**	Sets upstream channels with the following parameters:

	Command or Action	Purpose
	`Router(config-lb-group)# upstream Cable 1/0 2`	
Step 8	**docsis-policy** *policy-id* **Example:** `Router(config-lb-group)# docsis-policy 0`	Assigns a policy to a group with the parameter that becomes the default policy assigned to the CM, if the CM does not choose a different policy.
Step 9	**restricted** **Example:** `Router(config-lb-group)# restricted`	Selects the restricted group type. By default, the general group type is selected.
Step 10	**init-tech-ovr Cable** {*slot/subslot/port* | *slot/port*} *upstream* **Cable** {*slot/subslot/port* } | *slot/port upstream* **init-tech-list** *0-4* [**ucc**] **Example:** `Router(config-lb-group)# init-tech-ovr Cable 8/1/0 0 Cable 8/1/1 1 init-tech-list 1 ucc`	Sets DCC initialization techniques that overrides the physical upstream channel pair. The **init-tech-ovr** command can also be used to determine whether the UCC can be used for modems during dynamic upstream load balancing. The following parameters override the physical upstream channel pair: **Note** The **init-tech-list** keyword accepts an upstream that is not added into the load balancing group. The upstream channel pair is invalid until the upstream is added. When the load balancing group is removed, all upstream channel pairs are also removed.
Step 11	**service-type-id** *string* **Example:** `Router(config-lb-group)# service-type-id commercial`	Adds a service type ID, with the following parameter, that is compared against the cable modem provisioned service type ID, to determine an appropriate restricted load balancing group (RLBG):
Step 12	**tag** *tag name* **Example:** `Router(config-lb-group)# tag t1`	Adds a tag to the RLBG.

	Command or Action	Purpose
Step 13	**interval** *<1-1000>* **Example:** `Router(config-lb-group)# interval 60`	Sets the time interval, the Cisco CMTS waits before checking the load on an interface.
Step 14	**method** {**modems** | **service-flows** | **utilization**} {**us-method** {**modems** | **service-flows** | **utilization**}} **Example:** `Router(config-lb-group)# method modems us-method modems`	Selects the method the Cisco CMTS use to determine the load.

Step 15	**policy** {**pcmm** | **ugs** | **us-across-ds** | **pure-ds-load**}
	Example:
	`Router(config-lb-group)# policy us-across-ds`
	`Router(config-lb-group)# policy ugs`
	`Router(config-lb-group)# policy pure-ds-load`
		Selects the modems based on the type of service flow that are balanced.

Step 16	**threshold** {**load** {**minimum** *<1-100>* | *<1-100>*}| **pcmm** *<1-100>* | **stability** *<0-100>* | **ugs** *<1-100>*}
	Example:
	`Router(config-lb-group)# threshold load minimum 10`
	`Router(config-lb-group)# threshold pcmm 70`
	`Router(config-lb-group)# threshold load 10`
	`Router(config-lb-group)# threshold stability 50`
	`Router(config-lb-group)# threshold ugs 70`
		Selects the percentage of use beyond which load balancing occurs.

Step 17	**exit** **Example:** `Router# exit`	Exits DOCSIS LBG configuration.

Configuring DOCSIS 3.0 GLBG

The following sections describe how to configure a DOCSIS 3.0 GLBG and also how to configure default values of DOCSIS 3.0 certification for the DOCSIS 3.0 general group:

Note If a Cable interface on the line card is in "no shut down" state, the associated DOCSIS 3.0 GLBGs are restored in the running-configuration.

Configuring a DOCSIS 3.0 General Load Balancing Group

This section describes how to configure a DOCSIS 3.0 general load balancing group.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable load-balance docsis-enable** **Example:** `Router(config)# cable load-balance docsis-enable`	Enables DOCSIS load balancing on the Cisco CMTS.
Step 4	**cable load-balance docsis-group FN** *fn-id* **MD** *cable* {*slot/subslot/port* | *slot/port*} **Example:** `Router(config)# cable load-balance docsis-group FN 1 MD c5/0/0`	Enters the DOCSIS load balancing group configuration mode.
Step 5	**init-tech-list** *tech-list* [**ucc**] **Example:** `Router(config-lb-group)# init-tech-list 1 ucc`	Sets the DCC initialization technique list, with the following parameters.
Step 6	**disable** **Example:** `Router(config-lb-group)# disable`	Disables the load balance group.
Step 7	**docsis-policy** *policy-id* **Example:** `Router(config-lb-group)# docsis-policy 0`	Sets the load balance group policy.
Step 8	**interval** *1-1000* **Example:**	Sets the interface polling interval.

	Command or Action	Purpose
	`Router(config-lb-group)# interval 10`	
Step 9	**method** {**modems** | **service-flows** | **utilization**} {**us-method** {**modems** | **service-flows** | **utilization**}} **Example:** `Router(config-lb-group)# method modems us-method modems`	Sets the load balancing type or method.
Step 10	**policy** {**pcmm** | **ugs** | **us-across-ds** | **pure-ds-load**} **Example:** `Router(config-lb-group)# policy us-across-ds`	Sets load balancing policy.
Step 11	**threshold** {**load** {**minimum** *1-100* | *1-100*} | **pcmm** *1-100* | **stability** *0-100* | **ugs** *1-100*} **Example:** `Router(config-lb-group)# threshold pcmm 70`	Sets the load balancing threshold in percentage.
Step 12	**exit** **Example:** `Router# exit`	Exits the DOCSIS load balancing group configuration mode.

Configuring Default Values of DOCSIS 3.0 Load Balancing Group

This section describes how to configure default values of DOCSIS 3.0 certification for a DOCSIS 3.0 general group on the Cisco CMTS. A DOCSIS 3.0 general group is automatically created for each MD-CM-SG derived from the fiber node (FN) configuration, and the group parameters are set as default values.

Note The configured default values of DOCSIS 3.0 certification are applicable to the new automatically created DOCSIS 3.0 GLBGs and do not affect the existing DOCSIS 3.0 GLBGs. When a DOCSIS 3.0 GLBG is removed and recreated, its group parameters do not change.

Note The default settings for interface polling interval, load balancing method, policy for modems selection, and threshold usage in percent, can be configured for DOCSIS 3.0 general group. For more information, see the Cisco CMTS Cable Command Reference.

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable load-balance d30-ggrp-default disable** **Example:** `Router(config)# cable load-balance d30-ggrp-default disable`	Disables the default values of the DOCSIS 3.0 general load balance group (GLBG).
Step 4	**cable load-balance d30-ggrp-default init-tech-list** *tech-list* **Example:** `Router(config)# cable load-balance d30-ggrp-default init-tech-list 1`	Sets the default DOCSIS 3.0 GLBGs DCC and dynamic bonding change (DBC) initialization techniques.
Step 5	**cable load-balance d30-ggrp-default docsis-policy** *0-0xffffffff* **Example:** `Router(config)# cable load-balance d30-ggrp-default docsis-policy 2`	Sets the default DOCSIS 3.0 GLBGs policy ID.
Step 6	**exit** **Example:** `Router# exit`	Exits the global configuration mode.

Configuring Cable Modems to RLBG or a Service Type ID

This section shows how to configure a list of cable modems that are statically provisioned at the Cisco CMTS to a RLBG or a service type ID.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**cable load-balance restrict modem** *index mac-addr* [*mac-mask*] {**docsis-group** *docsis-group-id* | **service-type-id** *string*} **Example:** `Router(config)# cable load-balance restrict modem 1 001a.c30c.7eee FFFF.FFFF.0000 docsis-group 100`	Assigns a modem or a group of modems with a common MAC mask to a load balancing group or a service type ID.
Step 4	**exit** **Example:** `Router# exit`	Exits the global configuration mode.

Configuring Rules and Policies

This section shows how to create and configure rules and DOCSIS policies to restrict the movement of modems during load balancing. Rules determine whether a modem can be moved and during which time periods. The time periods are measured in seconds with the start time being an offset from midnight measured in seconds. Rules are created individually and can be combined into policies. The user is able to create DOCSIS policies that consist of one or more rules. When more than one rule is part of a DOCSIS policy, all rules apply. Each group has a default DOCSIS policy.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable load-balance rule** *rule-id* **Example:** `Router(config)# cable load-balance rule 1`	Creates a rule to prevent the modem from being moved.
Step 4	**cable load-balance rule** rule-id {**enabled** | **disabled** | {**disable-period dis-start** *0-86400*	Configures the rule.

	Command or Action	Purpose
	dis-period *<0-86400>***} | disable-throughput-lowerbound ds** | **us** *thrupt in kbps* | **vdoc-enabled}** **Example:** `Router(config)# cable load-balance rule 1 disable-period dis-start 40 dis-period 50`	**Note** Static multicast groups should be configured on the appropriate bundle interface as well as on the correct forwarding interfaces to enable this rule. This feature will not be supported on load balancing groups which are derived from fiber node configuration and with multicast encryption.
Step 5	**cable load-balance docsis-policy** *policy-id* **rule** *rule-id* **Example:** `Router(config)# cable load-balance docsis-policy 2 rule 1`	Associates a particular rule with the DOCSIS policy with the following parameters:
Step 6	**exit** **Example:** `Router# exit`	Exits the global configuration mode.

Troubleshooting Tips

Problem When you disable load balancing and enable it for the next day using the **cable load-balance rule** *rule-id* **disable-period dis-start** *start-time* **dis-period** *disable-period* command, the load balancing is enabled at 12.00 am instead of the configured *disable-period*.

Possible Cause Load balancing rule cannot be disabled and enabled on the next day (that is, after 24 hours) using a single load balancing rule.

Solution Configure separate load balancing rules for disabling load balancing and enabling it on the next day. Configure the rule to disable load balancing using the **cable load-balance rule** *rule-id* **disable-period dis-start** *start-time* **dis-period 0** command. Configure the rule to enable load balancing using the **cable load-balance rule** *rule-id* **disable-period dis-start 0 dis-period** *disable-period* command to enable it for the next day.

Configuring Load Balancing Parameter for a Cable Modem Movement Failure

This section describes how to configure the number of times a CM can fail before the CM is removed from the dynamic load balancing group.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable load-balance modem max-failures** *0-100* **Example:** `Router(config)# cable load-balance modem max-failures 10`	Configures the number of times a CM can fail before the CM is removed from the dynamic load balancing group.
Step 4	**exit** **Example:** `Router# exit`	Exits the global configuration mode.

Creating and Configuring TLV type Tag

The tags for TLV type matching rule are created and configured in this section.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable tag** *1-1000* **Example:** `Router(config)# cable tag 1`	Creates a tag. Enters the cmts-tag configuration mode.
Step 4	**name** *tag name* **Example:** `Router(cmts-tag)# name CSCO`	Specifies the name of the tag.
Step 5	[**exclude**] **service-type-id** *service-type-id* **Example:** `Router(cmts-tag)# service-type-id HSD`	Configures the specified service type ID for the tag.
Step 6	[**exclude**]**service-class** *service-class-name* **Example:**	Configures the specified service class name for the tag.

	Command or Action	Purpose
	`Router(cmts-tag)# service-class work`	
Step 7	[**exclude**] **docsis-version** *docsis version* **Example:** `Router(cmts-tag)# docsis-version docsis20`	Configures the specified DOCSIS version of the cable modem for the tag.
Step 8	**[exclude] oui** *oui of CM* **Example:** `Router(cmts-tag)# oui 00.1a.c3`	Configures the specified OUI of the cable modem for the tag.
Step 9	[**exclude**] **tlv** *type value* **Example:** `Router(cmts-tag)# tlv mrcs 4`	Configures the specified TLV type for the tag.
Step 10	**override** **Example:** `Router(cmts-tag)# override`	Overrides the TLV or SNMP during load balancing an RLBG.
Step 11	**exit** **Example:** `Router(cmts-tag)# exit`	Exits the cmts-tag configuration mode.
Step 12	**cable load-balance docsis-group** *docsis-group-id* **Example:** `Router(config)# cable load-balance docsis-group 1`	Creates a DOCSIS load balancing group on the Cisco CMTS. If the DOCSIS load balancing group is already present, the router enters the specified DOCSIS load balancing group configuration mode.
Step 13	**tag** *tag name* **Example:** `Router(config-lb-group)# tag CSCO`	Adds a tag to the load balancing group.
Step 14	**exit** **Example:** `Router(config-lb-group)# exit`	Exits the DOCSIS load balancing group configuration mode.
Step 15	**cable load-balance docsis-policy** *policy-id* **tag** *tag name* [**override**] **Example:** `Router(config)# cable load-balance docsis-policy 2 tag CSCO`	Creates a DOCSIS policy and associates a new rule or an existing rule with the policy.
Step 16	**exit** **Example:** `Router# exit`	Exits the global configuration mode.

Configuration Examples for DOCSIS Load Balancing Groups

This section describes a sample configuration example for configuring DOCSIS Load Balancing Groups including Restricted/General Load Balancing and downstream dynamic load balancing:

Example: Configuring a Tag

The following example shows how you can configure the tag to exclude a DOCSIS version, a MAC address, a service class name or a service type ID:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# cable tag 1
Router(cmts-tag)# exclude ?
  docsis-version   set the match rule for docsis version
  oui              set the match rule for oui
  service-class    set the match rule for service class name
  service-type-id  set the match rule for service type id
Router(cmts-tag)# exclude docsis-version ?
  docsis10  Match docsis 1.0 modems
  docsis11  Match docsis 1.1 modems
  docsis20  Match docsis 2.0 modems
  docsis30  Match docsis 3.0 modems
Router(cmts-tag)# exclude docsis-version docsis10
Router(cmts-tag)# exclude oui ?
  WORD  OUI of the vendor in the format xx.xx.xx or xx:xx:xx
Router(cmts-tag)# exclude oui 00.1a.c3
Router(cmts-tag)# exclude service-class ?
  WORD  Service class name
Router(cmts-tag)# exclude service-class work
Router(cmts-tag)# exclude service-type-id ?
  WORD  Service Type ID
Router(cmts-tag)# exclude service-type-id commercial
```

Example: Disabling Load Balancing

Use the following commands to disable DOCSIS 3.0 GLBG:

```
Router(config)# cable load-balance docsis-group FN 1 MD cable 6/0/0
Router(config-lb-group)# disable
Router(config-lb-group)#
```

Use the following commands to disable DOCSIS 3.0 RLBG:

```
Router(config)# cable load-balance docsis-group 1
Router(config-lb-group)# disable
Router(config-lb-group)#
```

Verifying DOCSIS Load Balancing Groups

This section describes how to use certain show commands to verify the configuration of the Restricted/General Load Balancing and Narrowband Dynamic Bandwidth Sharing with Downstream Dynamic Load Balancing feature.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**show cable load-balance docsis-group** {*docsis-group-id* | **FN** *fn-id* **MD cable** {*slot/subslot/port* | *slot/port*}} [**all** | **load** | **pending** | **statistics** | **target** | **modem-list** | **primary-load**]
	Example:
	`Router# show cable load-balance docsis-group 1`
	`Router# show cable load-balance docsis-group fn 1 MD c8/1/4`
		Displays real-time configurational, statistical, and operational information of the load balancing operations on the router.

Step 3	**show cable fiber-node** *fiber-node-id* [**spectrum**] **Example:** `Router# show cable fiber-node 3`	Displays information about a fiber node.
Step 4	**show cable load-balance** [**group** *n*] | [**all** | **load** | **pending** | **statistics** | **target** | **fiber-node-validation**] **Example:** `Router# show cable load-balance group 1`	Displays real-time statistical and operational information for load balancing operations. If given without any options, this command displays information for the load balancing groups and each cable interface's current load and load balancing status.
Step 5	**show cable modem** [*ip-address* | *mac-address* | **cable** *slot/port* [**upstream** *port*] | **name** *fqdn*] [*verbose*] **Example:** `Router# show cable modem 40.3.160.15 verbose`	Displays information for the registered and unregistered CMs.

Examples

Use the **show cable load-balance docsis-group** command to see the DOCSIS group status and to see the list of modems in the group, use the **show cable fiber-node** command to see the information

on fiber nodes, use the **show cable load-balance** command to see information on LBG and DOCSIS channels, and use the **show cable modem** command to see the information on all the CMs.

The following examples show the output of the **show cable load-balance docsis-group** command:

```
Router# show cable load-balance docsis-group 2
DOCSIS LB Enabled: Yes
DOCSIS     Group Status Interval DCC mask  Policy     Method Threshold
Group      Index                /UCC                DS/US  M/E/U/P/S
2          82    RE 10       0xF8(0)/N 0          s/s    1/1/70/70/50
Router# show cable load-balance docsis-group 1 modem-list
US                    Group Index Mac Address      Priority
Mo1/0/0:0/U0          81          (1)
                                  0000.ca45.9898   0
Mo1/0/0:0/U1          81          (0)
Mo1/0/0:0/U2          81          (2)
                                  0013.711c.0820   0
                                  0016.924f.8300   0
```

The output of the **show cable load-balance docsis-group** command is modified to include an additional field MUPFXLR to display more status information on the modems in the DOCSIS groups. For more information, see the Cisco IOS CMTS Cable Command Reference.

The following example shows the modified output of the **show cable load-balance docsis-group** command:

```
Router#show cable load docsis-group fn 1 md c6/0/0 modem-list
Load for five secs: 1%/0%; one minute: 2%; five minutes: 1%
Time source is NTP, 13:39:31.300 PDT Thu Mar 28 2013
Codes: M - Multicast, U - UGS, P - PCMM, F - Max-Failures, X - eXcluded
       L - L2vpn, R - RSVP
Primary DS     Grp Idx MAC Address    RCC-ID Bad Rfid Priority MUPFXLR
In6/0/0:0/UB   40448   (6)
                       e448.c70c.98af 1              2        -------
                       e448.c70c.9b76 1              2        -------
                       e448.c70c.9c15 1              2        -------
                       e448.c70c.9a92 1              2        -------
                       e448.c70c.99e4 1              2        -------
                       e448.c70c.9a35 1              2        -------
In6/0/0:0/U0   40448   (0)
In6/0/0:0/U1   40448   (1)
                       e448.c70c.9915                2        -------
In6/0/0:0/U2   40448   (0)
In6/0/0:0/U3   40448   (0)
In6/0/0:1/UB   40448   (5)
                       e448.c70c.9abc 1              2        -------
                       e448.c70c.993f 1              2        -------
                       e448.c70c.9927 1              2        -------
                       e448.c70c.9b82 1              2        -------
                       4458.2945.2cb8 1              2        -------
In6/0/0:1/U0   40448   (0)
In6/0/0:1/U1   40448   (0)
In6/0/0:1/U2   40448   (0)
In6/0/0:1/U3   40448   (0)
In6/0/0:2/UB   40448   (5)
                       e448.c70c.9759 1              2        -------
                       e448.c70c.9a0e 1              2        -------
                       e448.c70c.992d 1              2        -------
                       e448.c70c.9a38 1              2        -------
```

```
                     0025.2ed9.9984 1               2        -----L-
In6/0/0:2/U0   40448  (0)
In6/0/0:2/U1   40448  (0)
In6/0/0:2/U2   40448  (0)
In6/0/0:2/U3   40448  (0)
In6/0/0:3/UB   40448  (5)
                     e448.c70c.9c00 1               2        -------
                     e448.c70c.99a5 1               2        -------
                     e448.c70c.9a5f 1               2        -------
                     e448.c70c.9a3b 1               2        -------
                     e448.c70c.96b1 1               2        -------
In6/0/0:3/U0   40448  (0)
In6/0/0:3/U1   40448  (0)
In6/0/0:3/U2   40448  (0)
In6/0/0:3/U3   40448  (0)
```

The following example shows the output of the **show cable fiber-node** command:

```
Router# show cable fiber-node
Fiber-Node         Config Status
Fiber-Node 1
       Modular-Cable 1/0/0:  0-1
    FN Config Status: Configured (status flags = 0x01)
    MDD Status: Valid
```

The following examples show the output of the **show cable load-balance** command:

```
Router#show cable load-balance
Group  Interval  Method        DCC Init   Threshold
                               Technique  Minimum  Static  Enforce  Ugs   PCMM
1      10        service-flows 1          1        2%      2%       ---   ---
2      10        modems        0          5        10%     ---      ---   ---

DOCSIS LB Enabled: No
Router# show cable load-balance load
Interface            State      Group  Utilization  Reserved Modems Flows Weight
                                Index
Cable5/0/3 (459 MHz) up         1      0%(0%/0%)       0%    7      7     37
Cable5/0/3/U0        up         1      0%              0%    2      2     1.2
Cable5/0/3/U1        up         1      0%              0%    2      2     1.2
Cable5/0/3/U2        up         1      0%              0%    2      2     1.2
Cable5/0/3/U3        up         1      0%              0%    1      1     1.2
Cable5/0/4 (465 MHz) up         1      0%(0%/0%)       0%    7      7     37
Cable5/0/4/U0        up         1      0%              0%    1      1     1.2
Cable5/0/4/U1        up         1      0%              0%    2      2     1.2
Cable5/0/4/U2        up         1      0%              0%    2      2     1.2
Cable5/0/4/U3        up         1      0%              0%    2      2     1.2
Mo1/0/0:0 (555 MHz)  down       1      0%(0%/0%)       0%    0      0     0
Router# show cable load-balance fiber-node-validation
DOCSIS LBG ID    Match    Channel    Fiber-node list
1                match    Ca5/0/0/U0 {1}
                          Ca5/0/0/U1 {1}
                          Ca5/0/0/U2 {1}
                          Ca5/0/0/U3 {1}
                          Mo1/0/0:0  {1}
                          Mo1/0/0:1  {1}
2                mismatch Ca5/0/0/U0 {1}
                          Ca5/0/0/U1 {1}
```

```
                                Ca5/0/0/U2 {1}
                                Ca5/0/0/U3 {1}
                                Ca5/0/0    {}
```

The following example shows the output of the **show cable modem** command:

```
Router# show cable modem 40.3.160.19 verbose
LB group ID assigned(index)        : 1(81)
LB group ID in config file(index)  : N/A(N/A)
LB policy ID  : 0
LB policy ID in config file        : 0
LB priority                        : 0
Tag                                :
```

DOCSIS 3.0 GLBG is generated dynamically by the fiber node configuration, if a valid fiber node is configured.

For example, if the fiber node configuration is:

```
cable fiber-node 2
 downstream Modular-Cable 1/0/0 rf-channel 0-3
 downstream Cable7/0/0
 upstream Cable 7/0 connector 0-3
!
```

The GLBG generated by this fiber node is similar to:

```
Router# show cable load-balance docsis-group fn 2 md cable 7/0/0
DOCSIS 3.0 General LB
MD       FN  Group S Intv DCC mask  Policy     Mtd MD-CM-SG Threshold
             Index        /UCC                 D/U          M/E/U/P/S
Ca7/0/0  2   48129 E 30   0xF8(0)/N 0          m/m 0x3C0101 5/10/70/70/50
```

```
Router# show cable load-balance docsis-group fn 2 md cable 7/0/0 all
DOCSIS 3.0 General LB
MD       FN  Group S Intv DCC mask  Policy     Mtd MD-CM-SG Threshold
             Index        /UCC                 D/U          M/E/U/P/S
Ca7/0/0  2   48129 E 30   0xF8(0)/N 0          m/m 0x3C0101 5/10/70/70/50
Current load:
DOCSIS load-balancing load
Interface            State   Group  Utilization Rsvd NBCM  WB/UB Flows Weight
                             Index                    Total Total
Cable7/0/0 (333 MHz) up      48129  0%(0%/0%)   0%   2     8     7     37
Cable7/0/0/U0        up      48129  0%          0%   22    7     29    7.6
Cable7/0/0/U1        up      48129  0%          0%   21    8     28    7.6
Cable7/0/0/U2        up      48129  0%          0%   21    8     28    7.6
Cable7/0/0/U3        up      48129  0%          0%   20    10    30    7.6
Mo1/0/0:0 (501 MHz)  up      48129  0%(0%/0%)   0%   2     63    2     36
Mo1/0/0:0/U0         up      48129  0%          0%   22    7     29    7.6
Mo1/0/0:0/U1         up      48129  0%          0%   21    8     28    7.6
Mo1/0/0:0/U2         up      48129  0%          0%   21    8     28    7.6
Mo1/0/0:0/U3         up      48129  0%          0%   20    10    30    7.6
Mo1/0/0:1 (507 MHz)  up      48129  0%(0%/0%)   0%   1     58    1     36
Mo1/0/0:1/U0         up      48129  0%          0%   22    7     29    7.6
```

```
Mo1/0/0:1/U1          up      48129  0%           0%   21    8     28    7.6
Mo1/0/0:1/U2          up      48129  0%           0%   21    8     28    7.6
Mo1/0/0:1/U3          up      48129  0%           0%   20    10    30    7.6
Mo1/0/0:2 (513 MHz)   up      48129  0%(0%/0%)    0%   2     59    2     36
Mo1/0/0:2/U0          up      48129  0%           0%   22    7     29    7.6
Mo1/0/0:2/U1          up      48129  0%           0%   21    8     28    7.6
Mo1/0/0:2/U2          up      48129  0%           0%   21    8     28    7.6
Mo1/0/0:2/U3          up      48129  0%           0%   20    10    30    7.6
Mo1/0/0:3 (519 MHz)   up      48129  0%(0%/0%)    0%   1     61    1     36
Mo1/0/0:3/U0          up      48129  0%           0%   22    7     29    7.6
Mo1/0/0:3/U1          up      48129  0%           0%   21    8     28    7.6
Mo1/0/0:3/U2          up      48129  0%           0%   21    8     28    7.6
Mo1/0/0:3/U3          up      48129  0%           0%   20    10    30    7.6
Target assignments:
Interface             State      Group   Target
                                 Index
Cable7/0/0 (333 MHz)  up         48129
Cable7/0/0/U0         up         48129
Cable7/0/0/U1         up         48129
Cable7/0/0/U2         up         48129
Cable7/0/0/U3         up         48129
Mo1/0/0:0 (501 MHz)   up         48129   Mo1/0/0:1 (507 MHz)
Mo1/0/0:0/U0          up         48129
Mo1/0/0:0/U1          up         48129
Mo1/0/0:0/U2          up         48129
Mo1/0/0:0/U3          up         48129
Mo1/0/0:1 (507 MHz)   up         48129
Mo1/0/0:1/U0          up         48129
Mo1/0/0:1/U1          up         48129
Mo1/0/0:1/U2          up         48129
Mo1/0/0:1/U3          up         48129
Mo1/0/0:2 (513 MHz)   up         48129
Mo1/0/0:2/U0          up         48129
Mo1/0/0:2/U1          up         48129
Mo1/0/0:2/U2          up         48129
Mo1/0/0:2/U3          up         48129
Mo1/0/0:3 (519 MHz)   up         48129
Mo1/0/0:3/U0          up         48129
Mo1/0/0:3/U1          up         48129
Mo1/0/0:3/U2          up         48129
Mo1/0/0:3/U3          up         48129
Statistics:
Target interface       State      Transfers
                                  Complete  Pending   Retries   Failures
Cable7/0/0 (333 MHz)   up         8         0         0         0
Cable7/0/0/U0          up         30        0         0         0
Cable7/0/0/U1          up         83        0         0         0
Cable7/0/0/U2          up         48        0         0         0
Cable7/0/0/U3          up         34        0         0         0
Mo1/0/0:0 (501 MHz)    up         19        0         0         0
Mo1/0/0:0/U0           up         33        0         0         0
Mo1/0/0:0/U1           up         46        0         0         0
Mo1/0/0:0/U2           up         22        0         0         0
Mo1/0/0:0/U3           up         22        0         0         0
Mo1/0/0:1 (507 MHz)    up         22        0         0         0
Mo1/0/0:1/U0           up         9         0         0         0
Mo1/0/0:1/U1           up         19        0         0         0
Mo1/0/0:1/U2           up         15        0         0         0
Mo1/0/0:1/U3           up         21        0         0         0
Mo1/0/0:2 (513 MHz)    up         21        0         0         0
Mo1/0/0:2/U0           up         4         0         0         0
Mo1/0/0:2/U1           up         3         0         0         0
Mo1/0/0:2/U2           up         6         0         0         0
Mo1/0/0:2/U3           up         7         0         0         0
```

```
Mo1/0/0:3 (519 MHz)     up          9          0          0          0
Mo1/0/0:3/U0            up          1          0          0          0
Mo1/0/0:3/U1            up          2          0          0          0
Mo1/0/0:3/U2            up          4          0          0          0
Mo1/0/0:3/U3            up          4          0          0          0
Pending:
Modem          Grp Idx Primary RF/RCC              MD/TCS         Action Active Retries
                       Src           Target        Src    Target          Time
```

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

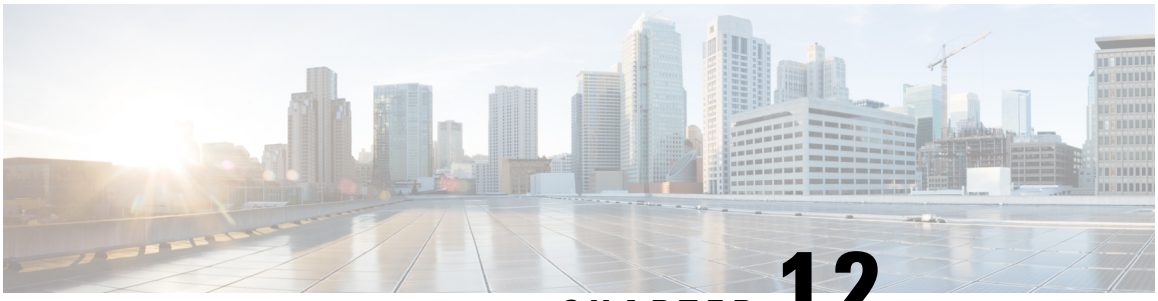
Feature Information for DOCSIS Load Balancing Groups

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 23: Feature Information for DOCSIS Load Balancing Groups

Feature Name	Releases	Feature Information
DOCSIS Load Balancing Groups	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the cisco cBR Series Converged Broadband Routers.



CHAPTER 12

DOCSIS Load Balancing Movements

Cisco CMTS supports static load balancing for MTC/MRC modems and dynamic load balancing for non-MTC and/or non-MRC modems. Support for configuration of load balancing groups (LBGs) that entail multiple interfaces, multiple load balancing policies, and the option to configure multiple additional load balancing parameters are also included.

The load balancing policies can be configured on the Cisco CMTS, indexed by an ID, to limit the movement of cable modems within a Load Balancing Group (LBG). The cable modem will forward TLV43.1in its registration request (REG-REQ) message, which is then parsed and stored in the Cisco CMTS. A policy defines whether and when cable modems can be moved within their load balancing groups.

During dynamic load balancing, the specified policy of the cable modem is checked to determine whether the cable modem is allowed to move.

Load balancing supports Dynamic Channel Change (DCC). DCC in DOCSIS 1.1, dynamically changes cable modem upstream or downstream channels without forcing a cable modem to go offline, and without re-registration after the change.

Load balancing distributes downstream load balancing with upstream channel loads in the same upstream load balancing group. This improves upon the prior load balancing limitation, in which load balancing was implemented on the basis of the entire downstream channel load.

Load balancing uses rules and policies to decide on moving the cable modems within their LB groups. These policies are created on the Cisco CMTS and chosen on a per-CM basis using type-length-value (TL V) portion (43.1,Policy ID) of REG-REQ. These policies prohibit a modem from being moved or restricted.

A policy contains a set of rules. When the policy is defined by multiple rules, all rules apply in combinations. A rule can be defined as "enabled" , "disabled" , or "disabled during time period. " Each rule can be used by more than one policy.

DOCSIS 3.0 static modem count-based load balancing uses the dynamic bonding change (DBC) to modify the following parameters of DOCSIS 3.0 cable modem with multiple transmit channel (MTC) mode or multiple receive channel(MRC) mode without primary channel change:

- Transmit channel set (TCS)
- Receive channel set (RCS)
- Downstream IDs (DSID) or DSID-associated attributes
- Security association for encrypting downstream traffic

These parameters and additional load balancing schemes are supported on the Cisco CMTS, and described in this document. This document describes all implementations of load balancing on the Cisco CMTS, dependent upon the Cisco IOS release installed and the desired parameters.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 202
- Prerequisites, on page 203
- Restrictions, on page 204
- Information on the Load Balancing on the Cisco CMTS, on page 208
- How to Configure Load Balancing, on page 224
- How to Configure Dynamic Channel Change for Load Balancing, on page 228
- Configuration Examples for Load Balancing, on page 234
- Additional References, on page 237
- Feature Information for DOCSIS Load Balancing Movements , on page 237

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 24: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites

Prerequisites for Load Balancing

The Load Balancing feature has the following prerequisites:

- Load balancing can be done only on upstreams and downstreams that share physical connectivity with the same group of cable modems.

Prerequisites for Dynamic Channel Change for Load Balancing

- DCC can be done only to a cable modem that is physically connected to both source and target upstream or downstream channels, or both.
- Upstreams and downstream channels that share the same physical connectivity must have different center frequencies separated by channel width.
- The difference between the physical layer parameters on the source and target DCC channels must be within the threshold required by the desired DCC initialization technique.

- DOCSIS 1.1 must be enabled for a modem to behave properly for the DCC operation. Note that not all DOCSIS 1.1 certified modems are DCC-capable, as the CableLabs DCC ATP tests need enhancement for complete coverage.

Prerequisites for Dynamic Bonding Change for DOCSIS 3.0 Static Modem Count-Based Load Balancing

- Initialization techniques 1 to 4, when used, require the Cisco CMTS to include the upstream channel descriptor (UCD) TLV (TLV46.5) in the DBC-REQ message.
- Bandwidth must be sufficient on the target bonding group to support DBC. This is determined by the admission control APIs.
- Fiber nodes must be configured before configuring DOCSIS 3.0 static modem count-based load balancing.

Restrictions

The following sections describe the restrictions applicable for the Load Balancing, Dynamic Channel Change, and Dynamic Bonding Change feature:

Restrictions for Load Balancing

The Load Balancing feature has the following restrictions:

- Load balancing can be done only on a per-line card basis—all interfaces in a load balancing group must be provided by the same line card.
- All downstreams and upstreams in a load balancing group must share physical connectivity to the same group of cable modems. All downstreams or all upstreams that have the same RF physical connectivity must be members of the same load balancing group.
- You can create a maximum of 256 load balancing groups on each line card.
- If an upstream port is operational, using the **no shutdown** command, and is not being used and not connected, load balancing attempts to use the port even though there are no cable modems registered on that port. When the upstream port is up, it is put into INIT state and load balancing includes this port as a potential target. However, if the load balancing sees multiple failures moving to this upstream, it is set to DISABLE state and the port is avoided later on in load balancing processes.
- The load balancing algorithms assume a relatively even distribution of usage among modems. In the situation where one cable modem creates the bulk of the load on an interface, the load balancing thresholds should be configured for a value above the load created by that single modem.
- You cannot select particular cable modems to be automatically moved for load balancing, although you can exclude cable modems from load balancing operations altogether on the basis of their MAC address or organization unique identifier (OUI). (You can use the **test cable load-balance** command to manually move a particular cable modem among upstreams, but this is done typically to test the configuration of the load balancing groups.)

- If you have configured upstream shared spectrum groups while doing downstream load balancing, the downstream in each MAC domain must not use overlapping upstream groups. For example, the downstream in one MAC domain could use an upstream spectrum band of 10 to 30 MHz, while the downstream in a second MAC domain could use an upstream spectrum band of 30 to 42 MHz. Each MAC domain has its own upstream shared spectrum group, allowing the load balancing group to contain the downstreams for both MAC domains.
- All upstream ports coming from the same splitter must be using different center frequencies that are separated by the channel width. For example, if the upstreams are using a channel width of 3.2 MHz, the center frequencies for all upstreams must be separated by at least 3.2 MHz.
- You can use four initialization techniques for Dynamic Channel Change (DCC).
- If you have configured load balancing, the provisioning system must not assign specific upstream channels or downstream frequencies to individual cable modems in their DOCSIS configuration files. Any cable modems requiring specific upstream channels or downstream frequencies must be excluded from load balancing operations (using the **cable load-balance exclude** command).
- Do not use the utilization method of load balancing on cable interfaces that have a small number of cable modems and where a single modem is responsible for the majority of the interface load. In this condition, the Cisco CMTS could end up continually moving cable modems from one interface to another in an endless attempt to load balance the interfaces. To avoid this, configure the utilization threshold to a value that is higher than what can be caused by any single cable modem.
- When deployed with channel restriction features, if the target upstream channel attribute masks are against that of the cable modem, then the cable modem on the higher load upstream will not be load balanced, as the current load balancing moves cable modems only to the target upstream. However, cable modems that do not have an attribute mask can still be load balanced. You should consider the following while deploying the load balancing groups: the target upstream will always be the upstream that has the lowest load. If some other upstreams have the same load, the upstream with the lowest index will be chosen as the target upstream.
- A TLV in a cable modem configuration file restricts dynamic load balancing on per modem basis.
- If you remove the last rule of a DOCSIS policy, the policy itself will be removed.
- The Cisco CMTS load balancing feature moves a cable modem based on the load of the channels in a load balancing group, without checking if the cable modem supports the extended frequency range (5Mhz-85Mhz). This may result in moving a cable modem that supports standard frequency range (5Mhz-65Mhz) to a channel that has extended frequency configured. To overcome such scenarios, operators should not mix upstreams that have standard and extended frequencies configured into the same load balancing group, unless all modems in the group support extended frequency range.

Restrictions for Dynamic Channel Change for Load Balancing

- The source and target upstreams and downstreams must share physical connectivity with the modem desired for a DCC transaction.
- Independent downstream change is not supported, and cross-MAC domain upstream changes must occur with the associated downstream changes.
- The source and target downstream interfaces must belong to the same virtual bundle and the same load balancing group if DCC is used for load balancing.

- For DCC initialization techniques 1 to 4, all the configuration variables of the cable modem must remain constant with the exception of the configuration variables that are explicitly changed by the Dynamic Channel Change request (DCC-REQ) messages encoding.
- DCC initialization techniques 2 to 4 must not be used if the propagation delay differences between the old and new channels exceeds the ranging accuracy requirement defined in DOCSIS, for example, ±0.25 usec plus ± symbol time.

  For example, for a symbol rate of 1.28 Msps, the timing offset difference between the source and target upstream channel is ± floor[(0.250 us + 0.5*0.781us)/(1/10.24)] = ± 6.
- The attenuation or frequency response differences between the old and new upstream channels causes the received power at the Cisco CMTS to change by more than 6 dB.
- DCC initialization technique 3 must not be used if the conditions for using technique 2 are not met.
- DCC initialization technique 4 must not be used if the conditions for using technique 2 cannot be met.
- Micro-reflections on the new upstream channel result in an unacceptable BER (greater than 1e-8) with pre-equalization coefficients set to the initial setting.
- DCC is used only for dynamic downstream load balancing on DOCSIS 1.1 and later CMs. Upstream Channel Change (UCC) is always used for dynamic upstream load balancing on DOCSIS 1.x CMs. For DOCSIS 2.x CMs, UCC is used when the *ucc* option is configured. For DOCSIS 3.x CMs, DCC is used irrespective of whether the *ucc* option is configured or not.
- Prolonged interruption of the multicast traffic is expected if the cable modem moved by DCC is the first one in a dynamic multicast group on the target interface. The downstream multicast service flow cannot be reestablished until the Cisco CMTS receives an Internet Group Management Protocol (IGMP) join message from the customer premises equipment (CPE) as the result of the Cisco CMTS IGMP query, where the IGMP query interval is set to one minute. This is an IGMPv2 limitation.
- Multiple statically-assigned IP addresses to a CPE can be pinged. However, this works only if all the security features, such as verification of IP addresses for cable modems and CPE devices on the upstream, and other security mechanism are disabled.
- The TCS and RCS assigned to the DOCSIS 3.0 cable modems are restricted by the upstream and downstream bonding groups configured by the Cisco CMTS.
- Load balancing and DCC are not supported for CMs that are enabled for Layer 2 VPN (L2VPN) support.
- When a DCC occurs, the cable modem US and DS counters are reset. The US and DS counters include counters such as data and throughput seen in the **show cable modem** (**mac-address**) **verbose** command output and packets and bytes seen in the **show cable modem (mac-address) counters** command output.

DCC Restrictions with N+1 Redundancy and Inter-Card Load Balancing

- Inter-card load balancing is not supported with cable interface line cards using N+1 redundancy. Refer to general DCC restrictions for additional information.
- Dynamic load balancing should not be used together with N+1 redundancy. Cable modems with outstanding DCC transactions go offline after a switchover event.

Note When cable modems go offline during a switchover event, the load balancing feature activates. Cable modems move in relation to the switchover event. When the cable modems return online, load balancing may need to initiate again.

To facilitate load balancing during a switchover, you can increase the dynamic load balance threshold, if a certain percentage of cable modems that reset during switchover is configured in the system. An alternate method is to use static load balancing with N+1 redundancy. For more information, see the Types of Load Balancing Operations.

Restrictions for DOCSIS 3.0 Static Modem Count-Based Load Balancing

- Static modem count-based load balancing is supported on MTC and MRC-only cable modems. Single-channel, narrowband cable modems will continue to be supported with dynamic load balancing. MRC-only modems are supported by dynamic load balancing on upstream channels.

Note DOCSIS 3.0 static modem count-based load balancing is not supported on:

  - Multiple line cards.
  - Load balancing groups and downstream channels shared across multiple line cards.

- DOCSIS 3.0 static modem count-based load balancing does not support service flow method of load balancing.

Restrictions for Dynamic Bonding Change for DOCSIS 3.0 Static Modem Count-Based Load Balancing

- The Cisco CMTS can use only DBC messaging to move modems within a MAC domain and applies only to cable modems operating in MTC mode or MRC-only mode without a primary downstream change.
- The Cisco CMTS moves the MRC-only cable modems with a primary channel change using DCC with initialization technique 0.
- The Cisco CMTS moves cable modems across MAC domains using only DCC with initialization technique 0.
- The Cisco CMTS must ensure minimum interruption to existing QoS services while considering an initialization technique that is suitable for the cable plant conditions.

Note DOCSIS 3.0 Static Load Balancing uses Initialization Technique 1 to move cable modems for DBC movement.

- Initialization Technique 1—(Broadcast initial ranging) may result in a lengthy interruption of service, which is mitigated by the reservation of QoS resources on the new channel(s). The service interruption can be further reduced if the Cisco CMTS supplies the UCD TLV in the DBC request in addition to providing more frequent initial ranging opportunities on the new channel.
- Initialization Technique 2—(Unicast ranging) offers the possibility of only a slight interruption of service. To use this technique, the Cisco CMTS must include the UCD TLV in the DBC message if the upstream channel is changing.
- Initialization Technique 3—(Broadcast or unicast ranging) offers the possibility of only a slight interruption of service. Use this technique when there is uncertainty when the CM may execute the DBC command and thus a chance that it might miss station maintenance slots. However, the Cisco CMTS should not use this technique if the conditions for using techniques 1 and 2 are not completely satisfied.
- Initialization Technique 4—(Use the new channel directly) results in the least interruption of service.

- For a DOCSIS 3.0 cable modem that in a DOCSIS 3.0 static load balancing group, the multicast join will be dropped before REG-HOLD time elapses.

Restrictions for MRC-Only Cable Modems

- MRC-only cable modems use single channel non-bonded upstreams (similar to narrowband (NB) modems) and multi-channel bonding groups on the downstream.

Note The following restrictions apply only to DOCSIS 2.0 and DOCSIS 3.0 cable modems in MRC-only mode.

- cable modems are moved across upstream channels using DCC.
- cable modems are moved to different downstream channels through DBC, if there is a change in the upstream channel and downstream channel bonding group, but not in the primary downstream channel and the upstream channel change is ignored.

  However, if there is a change in the primary downstream channel also, DCC with init tech 0 is used to balance the cable modems.
- MRC-only modems are treated similar to cable modems operating in MTC mode, to move modems across downstream channels. For change in upstream channel, MRC-only cable modems are treated similar to single-channel NB cable modems.

Information on the Load Balancing on the Cisco CMTS

This section describes the operation, concepts, and benefits of the Load Balancing on the Cisco CMTS feature:

Feature Overview

The Load Balancing on the Cisco CMTS feature allows service providers to optimally use both downstream and upstream bandwidth, enabling the deployment of new, high-speed services such as voice and video

services. This feature also can help reduce network congestion due to the uneven distribution of cable modems across the cable network and due to different usage patterns of individual customers.

By default, the Cisco CMTS platforms use a form of load balancing that attempts to equally distribute the cable modems to different upstreams when the cable modems register

This feature has been enhanced to make use of DOCSIS policies and rules to limit the movement of cable modems within a Load Balancing Group. A policy defines whether and when cable modems can be moved within their load balancing groups.

A policy consists of a set of rules. Each rule can be defined as “enabled”, “disabled”, or “disabled during time period.” Multiple policies can share a single rule. However, if you remove the last rule of a policy, that will also remove the policy.

Each rule can be used in any number of policies. When it is defined by multiple rules, all rules apply in combinations. Each rule helps to prohibit load balancing using a particular cable modem and to prohibit load balancing using a particular cable modem during certain times of the day.

Following are the general guidelines for the rules and policies:

- The policy or rule is recognized by a 32-bit ID.
- Each cable modem can have one policy only.
- Each rule can be associated to one or more policies.
- Each policy is described by at least one rule, otherwise it cannot be created.
- The zero Policy ID is reserved by Cisco CMTS indicating “Do nothing to LB prohibition.”
- If the policy ID specified by the cable modem configuration file is not configured on Cisco CMTS, no LB prohibition is applied to that CM. However, after the policy with the matched ID is configured, LB prohibition takes effect immediately.

Methods to Determine When Interfaces Are Balanced

In addition to selecting how interfaces should be balanced (using the static, passive, or dynamic types of load balancing), you can also select one of the following methods that the Cisco CMTS should use to determine when interfaces are balanced:

- Modems Method—Uses the number of active cable modems on an interface.
- Utilization Method—Uses the current percentage of utilization of an interface.

See the following sections for more information about each method.

Modems Method

The modem method of load balancing uses the number of active cable modems on an interface to determine the current load. This is a form of distribution-based load balancing, in which the absolute numbers of modems are used to determine whether interfaces are load balanced.

This method does not take into account the amount of traffic flowing through the cable modems, but the system does take into account the relative bandwidth of the channels being used, so that channels with higher bandwidths are allocated higher numbers of cable modems. This means that when interfaces are using different channel widths or modulation profiles, the system can assign different numbers of cable modems to the interfaces to achieve a balanced load. For example:

- Channel widths— If two upstreams are being load balanced, and one upstream is configured with a channel width of 1.6 MHz and the other upstream is configured for a channel width of 3.2 MHz, the Cisco CMTS allocates twice as many cable modems to the second upstream because its channel width is twice as large as the first upstream channel width.
- Modulation profiles— If one downstream is configured for 64-QAM and the other downstream is configured for 256-QAM, the Cisco CMTS allocates a proportionately larger number of cable modems to the second downstream so as to achieve a balanced load.

When both the channel width and modulation profile are set differently on two interfaces, the system calculates a "weight" value to use as a guide to determine the relative bandwidths of the interfaces.

Tip In a system with balanced loads, the interfaces will contain the same number of cable modems only when the interfaces are configured with the same channel width and modulation parameters.

Utilization Method

Note Narrowband cable modems, multiple downstream modems and upstreams of MRC-only cable modems participate in the utilization method.

The utilization method uses an interface's current percentage of utilization to determine the current load. This method uses the amount of traffic being sent over an interface, in the form of the percentage of total bandwidth being used. The system takes into account the relative throughput and bandwidth (as determined by the modulation profiles and channel widths) of each interface when evaluating the load on those interfaces.

For example, if two upstreams are being load balanced using the utilization method, and the first upstream has twice the bandwidth of the second upstream, the two upstreams are considered balanced when they reach the same percentage of utilization. The first upstream is carrying more traffic than the second upstream because it has a larger capacity for traffic, but the percentage of utilization will be the same.

Wideband Interface Average Utilization and Throughput

The average utilization and average throughput between Wideband interfaces of the same size can be calculated by:

Average Utilization (WB) = $\sum^{n}_{i-1}$rfch - util(rf_i)/n

- n represents the size of the Wideband interface
- $\sum^{n}_{i-1}$rfch - util(rf_i) represents the sum of rfch - util of QAM channels in Wideband interface.

Average Throughput (WB) = AverageThroughput(WB)in last 30s/$\sum^{n}_{i=1}$BW(rf_i)

- Average Throughput (WB) represents the KB recorded in the last 30 seconds
- $\sum^{n}_{i=1}$BW(rf_i) represents the total bandwidth of Wideband interface.

Note Use the **show cable load-balance load wideband** command to view the average utilization and average throughput between Wideband interfaces.

Load Balancing Parameters

You can determine which cable interfaces should participate in load balancing operations. You can also choose which of the following methods should be used to determine the current load on a cable interface, and therefore determine whether cable modems should be moved:

- Number of active cable modems
- Channel bandwidth utilization

You can also specify the threshold values that the Cisco CMTS should use to determine how to assign new cable modems to upstreams and downstreams for both types of load balancing. You can also configure whether cable modems with active Voice-over-IP (VoIP) calls should be moved, and if so, what thresholds should be used. You can also exclude certain cable modems from one or all of the different forms of load balancing.

Configurable Minimum Threshold under Utilization Method

The utilization method does not move cable modems for load balancing until the utilization of at least one of the interfaces reaches minimum threshold. This is done to avoid the unnecessary moving of cable modems due to temporary spikes in an interface's utilization rate.

Minimum utilization threshold can be configured under Utilization Method. The minimum utilization threshold may be configured in a range of 10 to 90 percent. As a result the cable modems will be moved only when the configured minimum utilization threshold is reached on an interface.

To configure the minimum threshold under the Utilization method, use the **cable load-balance method-utilization min-threshold** command in global configuration mode. For more information, refer to **cable load-balance method-utilization min-threshold** command reference.

Single Channel Load Balancing

Error Handling of Channel Assignment

As long as the interface state of the channels is not "administratively down", all channels are available for LBG assignment. For other load balancing operations, such as moving modems using DCC, or UCC, the interface state of the channels should be in "initial", "up", "suspicious", or "testing" states.

Downstream Load Balancing Distribution with Upstream Load Balancing

Downstream load balancing provides equalized load balancing with upstream group members. This enhancement synchronizes the “pending” statistic between different cable interface line cards in the load balancing group. The result is an alternative downstream load balancing scheme that makes use of per-upstream loads rather than total downstream loads.

This enhancement performs downstream load balancing that accounts for upstream channel loads in the same upstream load balancing group, rather than on the basis of the entire downstream channel load. Prior Cisco IOS releases may not have distributed cable modems evenly over individual upstream channels, nor in a way that accounted for downstream and upstream together.

The load balancing enhancement applies when downstream load balancing occurs on a headend system with separate upstream load balancing segments; the upstream segments are spread over multiple downstream segments.

The configuration and operation of making downstream load balancing decisions is enabled as follows:

- The target downstream segment is in the same downstream load balancing group as the source downstream segment.
- The upstream load balancing group can be set for the corresponding channel on which a cable modem is balanced.
- The Cisco CMTS automatically locates the upstream segment for a load balancing group and processes the upstream group status on the source interface that has the lowest load.
- The target downstream segment must have an upstream channel set in the upstream load balancing group.
- The highest target upstream segment must carry less load than any other potential target—the highest upstream segment on other interfaces.

For example, several upstream segments can be configured across multiple downstream segments as follows:

```
       U0      U1      U2      U3     Downstream
3/0    LB10    LB11    LB12    LB13      LB1
4/0    LB10    LB11    LB12    LB13      LB1
5/0    LB10    LB11    LB12    LB13      LB1
6/0    LB10    LB11    LB12    LB13      LB1
```

In this example, a cable modem that comes online on the interface cable 5/0 Upstream 2 could potentially come online on the following interfaces:

- cable 3/0 upstream 2
- cable 4/0 upstream 2
- cable 6/0 upstream 2

The enhancement enables the following advantages and behaviors:

- This enhancement adds support for synchronizing the "pending" statistic between different cable interface line cards and the network processing engine (NPE) so that a better decision can be made about where cable modems should be moved. This function can be used as a normal downstream load balancing implementation, if desired.
- This enhancement adds the **us-groups-across-ds** keyword to **cable load-balance group** command for configuring downstream load balancing groups with upstream resources.

Note You can use the **no cable load-balance docsis20-enable** command to disable DOCSIS 2.0 dynamic downstream and upstream load balance.

Upstream Load Balancing for DOCSIS 3.0 Cable Modems in Single Upstream Mode

The upstream load balancing functionality enables the Cisco CMTS router to effectively handle upstream traffic for wideband and narrowband cable modems that are in single upstream mode. Single upstream mode (Mx1) means that the modems cannot send upstream traffic on multiple upstream channels. In the event of traffic overload on a single upstream channel of a wideband or narrowband cable modem, the Cisco CMTS router automatically moves the cable modem to another upstream channel in the same load balancing group.

Note A cable modem operating in single upstream mode is assigned to a load balancing group based on the primary channel of the modem. A cable modem in single upstream mode can support multiple receive channel (MRC) mode or narrowband mode. However, a cable modem in single upstream mode cannot support multiple transmit channel mode (MTC).

Interaction with Spectrum Management

Cisco cable interface line cards support a number of features to maximize channel bandwidth and to minimize the impact of ingress noise on cable modem traffic. These features have the following impacts upon load balancing operations:

- Frequency hopping—Frequency hopping does not affect the load balancing algorithm, because it does not change either the bandwidth of a channel nor the number of cable modems on an interface.
- Dynamic modulation changes—The dynamic modulation feature affects the load balancing algorithm because it typically switches an interface from a higher-bandwidth modulation profile to a lower-bandwidth modulation profile in response to noise conditions on the interface.

  For example, if an upstream is configured for 16-QAM, sufficient noise levels could switch the upstream to a QPSK modulation profile. Depending on the load balancing configuration, this could then result in the movement of cable modems to other channels. Similarly, when the noise conditions improve, and the modulation is returned to the original, higher-bandwidth profile, the cable modems could be moved again to rebalance the upstream channels.
- Channel width changes—Cisco cable interface line cards support automatic changes to the channel width in response to noise conditions. Because changing the channel width affects the throughput of a channel, this also affects the load balancing algorithm.

  For example, if noise makes the current channel width unusable, the Cisco cable interface line card reduces the channel width until it finds a usable channel width. Because this reduces the available bandwidth on the channel, the load balancing algorithm moves cable modems to rebalance the upstreams.

  In addition, the Cisco cable interface line card does not automatically restore the original channel width when noise conditions improve. Instead, the card changes the channel width only when it performs a subsequent frequency hop, either in response to additional noise conditions or when an operator performs a manual frequency hop. When the hop occurs, the card then searches for the largest possible channel width, and this could result in another movement of cable modems to rebalance the channels.

Using Dynamic Channel Change

DCC in DOCSIS 1.1 dynamically changes cable modem upstream or downstream channels without forcing a cable modem to go offline, and without re-registration after the change. DCC supports five different initialization methods (0-4).

- Load balancing techniques allow for moving cable modems with DCC by using configurable initialization techniques.
- DCC allows line card channel changes across separate downstream channels in the same cable interface line card, with the DCC initialization techniques ranging from 0 to 4.
- DCC transfers cable modem state information from the originating downstream channel to the target downstream channel, and maintains synchronization of the cable modem information between the cable interface line card and the Network Processing Engine (NPE) or Route Processor (RP).

- Applications that are sensitive to delay , such as PacketCable (PC) and PacketCable Multi Media (PCMM), may use DCC initialization technique4 to retain services while the cable modem is performing DCC.
- If the channel is in mixed or ATDMA-only mode, the primary Service Identifier (SID) must be switched to ATDMA-only mode.

Note You can use the **no cable load-balance docsis20-enable** command to disable DOCSIS 2.0 dynamic downstream and upstream load balance.

Multiple Channel Load Balancing

Algorithm for Bonded Channel Cable Modem Load Balancing

During registration of the cable modem, the modem count-based method uses the number of active cable modems on the allowed RCS to determine the current load on each channel. After the modem is assigned an RCS, the Cisco CMTS does not move the cable modem even when traffic conditions change.

When a cable modem sends a registration request, modem count-based method of load balancing ranks the allowed receive channel sets (RCS) based on their modem count and assigns the set with the lowest number of CMs, to the ranging cable modem.

DOCSIS 3.0 Static Modem Count-Based Load Balancing

The static modem count-based load balancing supports the following:

- DOCSIS General and Restricted load balancing group assignment to include DOCSIS 3.0 cable modems in MTC and MRC-only modes.

  Note DOCSIS 3.0 static modem count-based load balancing is not supported:

  - Across multiple line cards.
  - For load balancing groups and downstream channels shared across multiple line cards. However, autonomous load balancing-based CM steering and load balancing group assignment is supported across multiple line cards

- Use of DCC and DBC in load balancing.
- Use of DBC for MRC-only modems during downstream move.
- Use of DCC with init tech 0 if the primary downstream channel is changed for MRC-only CMs.
- Use of DBC for cable modems in MTC mode for all upstream and downstream modem move.
- Separate counters for NB and wideband (WB)/upstream bonding (UB) CMs. For more information, see the **show cable load-balance docsis-group** command in the Cisco IOS CMTS Cable Command Reference.

- Aggregate logical channels to physical channels for load balancing. Physical channel load is calculated by using average weights among all logical channels.
- Non-primary downstream channels load where utilization of SPA QAM is considered

Note When the CM counts across different WB interfaces are within predefined threshold levels, the load is always considered as balanced; no more CM move is initiated by the LB system. No service flow count, whether primary or secondary, is taken into consideration during this LB process.

Note The attributes considered for the forward interface for the service flow (SF) are attribute mask and available bandwidth, and *not* the number of service flows on each channel. If a channel is within the new RCS, then irrespective of the type of narrowband SF, (whether primary or secondary, or static or dynamic) the SF continues to use its current channel.

Note The US Phy Mode counters (scdma, atdma, and tdma) remain 0 for the UB interfaces.

DOCSIS 3.0 static modem count-based load balancing is based on legacy load balancing and supports any type of channel combination (upstream and downstream)—MxN, with 1x1 combination being the subset.

DOCSIS 3.0 static modem count-based load balancing controls dynamic changes to the set of downstream and upstream channels used by a registered CM. It supports the following:

- Multiple channel load balancing operation.
- Load balancing operation based on policies and priorities.
- Load balancing with multicast. DOCSIS 3.0 static modem count-based load balancing does not move any CM with active video sessions.

DOCSIS 3.0 static modem count-based load balancing supports the modem count-based load balancing in a hybrid deployment of DOCSIS 1.x, 2.0 and 3.0 cable modems.

Static modem count-based load balancing is supported only for DOCSIS 3.0 CMs. Single-channel, narrowband cable modems will continue to be supported with dynamic load balancing. MRC-only cable modems are supported by dynamic load balancing on upstream channels.

With DOCSIS 3.0 static modem count-based load balancing, when load balancing related configuration within the LBG is changed as follows, the cable modems are forced to re-register:

- Partial shut or no shut interfaces under the LBG domain
- MRC or MTC mode in cable modems is turned on or turned off
- Change in fiber node for GLBG
- Change in wideband configuration for downstream group
- Change in the upstream bonding group

Use the following commands to force cable modems to re-register:

- **clear cable modem delete**
- **clear cable load state**
- **clear cable load counters**

Primary Channel Load Display for Target RCS

This feature enables the bonded modems to be moved at the time of registration such that the primary channels are distributed evenly among the primary-capable channels apart from the load being balanced on the target DS channels. Modem method ranks the RCS based on their primary loads and assigns the set with the lowest primary load to the ranging cable modem.

An optional keyword **primary-load** has been added to the **show cable load-balance docsis-group** command to display the primary load of an RCS. For more information, see the Cisco CMTS Command Reference.

Although the modem count-based method distributes the cable modems fairly as they register, the following conditions may cause a system imbalance:

- A channel or groups of channels fail because of a planned (administrative shutdown) or unplanned event.
- While some cable modems may continue to operate in partial mode, some may re-register because of the failure and are reassigned to the channels that are active.
- When the failed channels become operational again, the cable modems do not re-register and the system is unbalanced.

In this case, the modem count-based method sends an SNMP trap to alert the operator, and the operator can choose to manually intervene to re-balance the cable modems by resetting the MAC domain to force all cable modems to re-register.

Note For cable modems in MRC and MTC modes, the modem count based load balancing method considers the number of active modems and service flows on the primary channels in the RCS and TCS of the cable modem.

Note Use **no cable load-balance docsis30-enable static** command to disable this feature.

Dynamic Load Balancing for DOCSIS 3.0 Cable Modems

The existing Load Balancing (LB) feature is enhanced to cope with the increase in the number of downstream and upstream channels by Multi-Service Operators (MSO) and wider deployment of 16-channel, 24-channel and multiple downstream channel Cable Modems (CMs). This enhancement allows the customer to better utilize their available bandwidth. The enhancements made to the existing LB feature include:

- Utilization based Dynamic downstream LB for DOCSIS 3.0
- Support for DOCSIS 3.0 LB statistics
- Enable or Disable DOCSIS 3.0 LB feature
- Distribute the CM on all the primary channels for the target interface when performing DOCSIS 3.0 LB dynamic movement. This feature is only used on DOCSIS 3.0 LB dynamic movement. It is disabled by

default. Use **method utilization primary-distributed** command in the DOCSIS load balancing group mode to enable this feature. To disable this feature, use **no** form of this command or **method utilization us-method** command.

Note Use **cable load-balance docsis-enable** command to enable this feature. In addition, use **cable load-balance docsis30-enable** and **cable load-balance docsis30-enable dynamic downstream** command to enable dynamic and utilization based dynamic downstream LB for DOCSIS 3.0 Cable Modems.

Multiple Channel Load Balancing Operation

CMs load balance in MRC and MTC modes. The following rules apply while load balancing CMs operating in these modes:

- For CMs operating in MRC and MTC modes, DBC is used to move CMs across downstreams by changing the RCS of the CM within same MAC domain.

  CMs operating in MRC-only mode can be moved across upstreams only through a DCC request. However, the Cisco CMTS uses DCC with initialization technique 0 (reinitializing the MAC domain) when changing the downstream channel of a CM operating in MRC mode.

- During CM registration, the Cisco CMTS may send a multipart registration response (REG-RSP-MP) message to include a TCC TLV encoding to the CM. This CM is marked as TCC-capable.

  For CMs operating in MRC, non-MTC, non-TCC-capable mode, load balancing uses:

  - DBC to change RCS of the CM
  - DCC to change upstream channel of the CM

- For CMs operating in narrowband mode, DCC is used to move CMs within and across MAC domains.

The tables below provide a snapshot view of the load balancing methods and the operations used to move bonded and non-bonded CMs:

Table 25: Load Balancing Method to Move Bonded and Non-bonded CMs

Modem Mode	Load Balancing Method	Load Balancing Counters	Channels	Dynamic Service Charge (Initialization Technique)	
				Within MAC Domain	Across MAC Domains

Modem Mode	Load Balancing Method	Load Balancing Counters	Channels	Dynamic Service Charge (Initialization Technique)	
DOCSIS 3.0 CM in MTC mode	DOCSIS 3.0 static modem count-based load balancing (MCBLB) DOCSIS 3.0 dynamic load balancing	WB/UB	DS	DBC **Note** When DOCSIS 3.0 LB is enabled, and the MTC CM is outside RLBG, CM is moved inside RLBG.	DCC init tech 0
	DOCSIS 3.0 static modem count-based load balancing (MCBLB)	WB/UB	US	DBC **Note** When DOCSIS 3.0 LB is enabled, and the MTC CM is outside RLBG, CM is moved inside RLBG.	DCC init tech 0

Modem Mode	Load Balancing Method	Load Balancing Counters	Channels	Dynamic Service Charge (Initialization Technique)	
DOCSIS 3.0/D2.x CMs in MRC-only mode	DOCSIS 3.0 static MCBLB DOCSIS 3.0 dynamic load balancing	WB/UB	No change to the primary DS channel	DBC **Note** When DOCSIS 3.0 LB is enabled and CM with all DSs is outside RLBG, CM is moved inside RLBG.	DCC init tech 0
			Change to the primary DS channel	DCC init tech 0 **Note** CM with primary DS outside RLBG moves inside RLBG with DOCSIS 2.0 LB.	DCC init tech 0
DOCSIS 3.0 CMs in MRC-only mode	DOCSIS 2.0 static and dynamic MCBLB, dynamic utilization	NB	US	DCC **Note** CM outside RLBG moves inside RLBG with DOCSIS 2.0 LB.	DCC init tech 0

Modem Mode	Load Balancing Method	Load Balancing Counters	Channels	Dynamic Service Charge (Initialization Technique)	
D2.x CMs in MRC-only mode	DOCSIS 2.0 static and dynamic MCBLB, dynamic utilization	NB	US	DCC/UCC **Note** CM outside RLBG moves inside RLBG with DOCSIS 2.0 LB.	DCC init tech 0
DOCSIS 2.0 /DOCSIS 1.1 CMs in NB mode	DOCSIS 2.0 dynamic MCBLB, dynamic utilization	NB	DS	DCC **Note** CM outside RLBG moves inside RLBG with DOCSIS 2.0 LB.	DCC init tech 0
			US	UCC **Note** CM outside RLBG moves inside RLBG with DOCSIS 2.0 LB.	UCC

Modem Mode	Load Balancing Method	Load Balancing Counters	Channels	Dynamic Service Charge (Initialization Technique)	
DOCSIS 1.0 in NB mode	DOCSIS 2.0 dynamic MCBLB, dynamic utilization	NB	DS	Force reinitialize CM **Note** CM outside RLBG moves inside RLBG with DOCSIS 2.0 LB.	Force reinitialize CM
			US	UCC **Note** CM outside RLBG moves inside RLBG with DOCSIS 2.0 LB.	UCC

Table 26: Using DCC/DBC to Load Balance Bonded and Non-bonded Cable Modems

Channel	CM in MRC, MTC Mode	CM in MRC, non-MTC Mode	DOCSIS 1.1/2.0 CMs with Single US/DS	DOCSIS 1.0 CMs with Single US/DS
Upstream (US)	DBC	DCC	DCC	UCC
Downstream (DS)	DBC (within the same MAC domain)	DBC (within the same MAC domain)	DCC (within the same MAC domain)	Force reinitialize CM
	DCC with initialization technique 0 when moving CMs across MAC domains	DCC with initialization technique 0 when moving CMs across MAC domains	DCC with initialization technique 0 when moving CMs across MAC domains	Force reinitialize CM

Using DBC for DOCSIS 3.0 Load Balancing Movement

As part of the DOCSIS 3.0 specifications, at any time after registration, the Cisco CMTS uses the DBC command to change any of the following parameters in a DOCSIS 3.0 CM:

- Receive channel set
- Transmit channel set

- DSID(s) or DSID associated attributes
- Security association(s) for encrypting downstream traffic
- Service Flow Cluster Assignments

Note Only RCS and TCS are used by the DOCSIS 3.0 load balancing.

Use the **show cable load-balance docsis-group** command to display the current, real-time statistics for load balancing operations. For more information, see the Cisco IOS CMTS Cable Command Reference.

Using DBC to Change the Receive Channel Set

The Cisco CMTS can add, delete, or change the channels in the RCS of a cable modem by including a RCC in the DBC-REQ.

If an RCS change affects the primary downstream channel of the cable modem, the cable modem is required to re-register on its upstream channels.

If channels are deleted from the RCS, the Cisco CMTS may stop sending traffic on the downstream channel to be removed, which may cause loss of traffic. The Cisco CMTS minimizes packet loss by duplicating traffic on the new and old RCS until it receives a DBC-RSP from the cable modem.

Note For cable modems in MRC-only mode, a downstream channel move is initiated by a DBC message. However, DCC initialization technique 0 is used if there is a change in the primary downstream channel.

Using DBC to Change the Transmit Channel Set

The Cisco CMTS can add, delete, or replace one or multiple channels in the TCS in a single DBC message. Whenever the TCS of the cable modem changes, the CMTS appropriately modifies the service identifiers (SIDs) associated with the affected service flows.

A change in the TCS is accompanied by a valid initialization technique.

Using DBC to Change the Downstream ID

Using DBC, the Cisco CMTS can change the following attributes of a downstream ID (DSID):

- Re-sequencing encodings:
  - Downstream re-sequencing channel list—The CMTS can add, delete, and replace channels in the DS re-sequencing channel list.
  - DSID re-sequencing wait time—The CMTS can indicate a change in skew due to network or configuration changes through DSID re-sequencing wait time.
- re-sequencing Warning Threshold
- CM-STATUS Hold-Off Timer for Out-of-range Events
- Multicast Encoding—The CMTS can initiate a DBC transaction to either add, deleted, or change attributes of an existing multicast DSID:

- Client MAC Address
- Multicast cable modem interface Mask
- Group MAC Address

Using DBC to Change the Security Association for Encrypting Downstream Traffic

- The CMTS can initiate a DBC transaction to add or delete Security Associations (SA) used to encrypt downstream traffic.
- The CMTS cannot send a DBC request to a cable modem that is not in the "Authorized" State.
- The CMTS can send a DBC request with an SA that uses a cryptographic suite unsupported by the cable modem. However, if the cable modem receives a DBC request with an SA that it is not capable of using, the cable modem rejects the DBC request.

Using DBC to Change the Service Flow SID Cluster Assignments

The Cisco CMTS uses the Service Flow SID Cluster Assignments TLV in the DBC request to assign new channels to a service flow, remove channels from a service flow, or replace one channel with another for a service flow.

Note Multiple actions can occur within a single DBC message.

Benefits of Load Balancing

The Load Balancing feature on the Cisco CMTS provides the following benefits to cable service providers and their partners and customers:

- Provides a method that service providers can use for efficient bandwidth utilization, especially when using multiple upstream channels per fiber node.
- Allows service providers to expand their networks in an efficient manner, avoiding the cost of having to install additional fiber optic equipment and further segmenting the physical plant.
- Load balancing on downstream channels enables efficient bandwidth usage when using multiple downstream channels per fiber node to enable Video over IP and other services that require high-bandwidth real-time streams.
- Load balancing of upstream and downstream channels does not require any change to the provisioning servers or to any DOCSIS configuration files.
- Load balancing of upstream and downstream channels does not require any administrator or user intervention (such as manually resetting cable interfaces or manually rebooting cable modems).
- Allows service providers to equally balance their downstreams as cable modems register, so that cable modems do not all attempt to register on the same downstream, resulting in many cable modems failing to register and having to search for a new downstream.
- Cable modems can be moved among downstream and upstream channels without having to change any network parameters in manual fashion, such as IP address.

- Allows service providers to stay ahead of customers' bandwidth demands by dynamically responding to current load-usage conditions.
- Allows service providers to optimize the load balancing parameters for critical services, such as Voice over IP (VoIP).

Exclude Cable Modems from Load Balancing Groups

Load Balancing Process

The load balancing process has two phases.

- Assignment phase.

  When a modem is coming online in the assignment phase, the modem is moved to the load balance group by assigning it a load balancing group (LBG) ID. The assignment phase occurs only when a modem is coming online.

- Balancing phase.

  In the balancing phase, a modem is re-assigned to an LBG to balance the load.

Excluding Cable Modems from Load Balancing

There are four options that are used to exclude cable modems from an LBG:

- The **assignment** option:

  The **assignment** option is used to exclude a modem during the assignment phase. The modem is not assigned an LBG and LBG ID is not displayed in the output of the **show cable modem verbose** command. The **assignment** option cannot be used when a modem is already online.

- The **static** option:

  The **static** option is used to exclude a modem during the Balancing phase. The modem is assigned to an LBG with an LBG ID. The **static** option is used to exclude a modem during static load balancing.

- The **enforce** option:

  The **enforce** option is similar to the **static** option, except that the **enforce** option is used to exclude a modem during dynamic load balancing.

When a cable modem is excluded from load balancing using the **assignment** option, the cable modem is not available for load balancing using the **static** or the **enforce** options.

- The **strict** option:

  The **strict** option excludes a modem in both the phases of load balancing. When a modem is online already, the **strict** option applies the **static** and the **enforce** options. It applies the **assignment** option only when the modem comes online again.

How to Configure Load Balancing

To configure load balancing groups, and to enable load balancing, refer to the configurations in the *DOCSIS Load Balancing Groups* document. Each task is marked as required or optional, as appropriate.

Enabling Single Channel Load Balancing

To configure Single Channel Load Balancing, see the *DOCSIS Load Balancing Groups guide*.

Configuring Dynamic Bonding Change for DOCSIS 3.0 Static Load Balancing

Use the **cable load-balance docsis30-enabled** command in the global configuration mode, to enable DOCSIS 3.0 Static Load Balancing.

Note DOCSIS 3.0 Static Load Balancing always uses Modem Count Method for load balancing.

Before you begin

Configure Load Balancing Groups. For more details, see the *DOCSIS Load Balancing Groups guide*.

Excluding Cable Modems from a Load Balancing Group

This configuration is optional. This section describes how to exclude a particular cable modem, or all cable modems from a particular vendor, from participating in static or dynamic load balancing operations, and optionally marking the modems for passive load balancing. This task is optional, because, by default, cable modems on an interface participate in whatever load balancing operations have been configured.

Note This step might be required for some cable modems that are not DOCSIS-compliant. Such cable modems can go offline for long periods of time when load balancing is attempted using DOCSIS MAC messages. If this is the case, use the **cable load-balance exclude** command to exclude such cable modems from load balancing operations until the modem can be upgraded to DOCSIS-compliant software.

Tip You must exclude cable modems that require specific upstream channels or downstream frequencies. Load balancing cannot be done when cable modems are assigned specific channels or frequencies in their DOCSIS configuration files.

Support for Excluding Old Devices

Load balancing for old cable devices like Set Top Boxes (STBs) which do not support load balancing, will fail. In the output for **show cable load-balance group** command, these devices will show as 'suspicious' and then as 'disabled'. This will disrupt normal operations of other modems in the load balancing group. To exclude these STBs, a **cable load-balance exclude** command is configured to exclude each STB.

Note You can configure the **cable load-balance exclude** command once to exclude all the STBs, that do not support load balancing, instead of configuring the command several times with matched MAC addresses. You can also move cable modems that were moved to a load balancing group in assignment phase.

The **cable load-balance exclude** modem command is modified to include the *mask* argument as an optional argument. The MAC address of a cable modem that belongs to the range specified by the MAC address mask, will be excluded by matching the "1" bit in mask. While configuring a new range rule using the *mask* argument, an existent rule with the same range is overwritten.

The **cable load-balance exclude** modem command is modified to include the **assignment** option. This option allows you to exclude a cable modem that was moved into a load balancing group in assignment phase.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable load-balance exclude** {**modem** *mac-address* [*mac-mask*] | **oui** *oui-value*} [**assignment** | **enforce** | **static** | **strict**] **Example:** `Router(config)# cable load-balance exclude oui 00:00:0c`	Specifies that one or more cable modems should be excluded from load balancing operations. By default, the cable modems are excluded from dynamic and static load balancing, but they continue to participate in passive load balancing. Use the following options to exclude the cable modems from others combinations of load balancing:
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Distributing Downstream Load Balancing with Upstream Load Balancing

Two commands are used to configure or display the configuration and status of distributed load balancing on the Cisco CMTS:

- **cable load-balance group** *ds-lb-group-id* **policy** {**pcmm** | **ugs** | **us-groups-across-ds**}
- **show cable load all**

The optional configuration of making downstream load balancing decisions is enabled as follows:

- The target downstream segment is in the same downstream load balancing group as the source downstream segment. This feature finds the target frequency and interface based on the upstream loads within the same upstream group as the source.
- The upstream load balancing group can be set for the corresponding channel on which a cable modem is balanced on the downstream channels.
- The Cisco CMTS automatically locates the upstream segment for a load balancing group and processes the upstream group status on the source interface that has the lowest load.
- The target downstream segment must have an upstream channel set in the upstream load balancing group.
- The highest target upstream segment must carry less load than any other potential target—the highest upstream segment on other interfaces.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable load-balance group** *ds-lb-group-id* **policy** {**pcmm** | **ugs** | **us-groups-across-ds**} **Example:** `Router(config)# cable load-balance group 1 policy us-groups-across-ds`	Sets the type of service flow policy for use with Load Balancing. This command synchronizes the pending statistic between different cable interface line cards in the load balancing group. The result is an alternative downstream load balancing scheme that makes use of per-upstream loads rather than total downstream loads when making load balancing decisions.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 5	**show cable load all** **Example:** `Router# show cable load all`	Displays load balancing statistics and status of load balancing configurations on the Cisco CMTS, to include distributed upstream-to-downstream load balancing when configured.

How to Configure Dynamic Channel Change for Load Balancing

DCC in DOCSIS 1.1 dynamically changes cable modem upstream or downstream channels without forcing a cable modem to go offline, and without reregistration after the change. DCC supports five different initialization methods (0-4), instead of one, as in earlier DOCSIS support.

Dynamic Channel Change (DCC) and DCC for Load Balancing on the Cisco CMTS supports the following:

- Load balancing techniques allow for moving cable modems with DCC by using configurable initialization techniques.
- DCC allows line card channel changes across separate downstream channels in the same cable interface line card, with the DCC initialization techniques ranging from 0 to 4.
- DCC transfers cable modem state information from the originating downstream channel to the target downstream channel, and maintains synchronization of the cable modem information between the cable interface line card and the Network Processing Engine (NPE) or Route Processor (RP).
- Applications that are sensitive to delay, such as PacketCable (PC) and PacketCable MultiMedia (PCMM), may use DCC initialization technique 4 to retain services while the cable modem is performing DCC.
- If the channel is in mixed or ATDMA-only mode, the primary Service Identifier (SID) must be switched to ATDMA-only mode.

Configuring Dynamic Channel Change for Load Balancing

To configure the DCC feature for load balancing, use the following steps. Values indicated are sample values that may differ from your own.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable load-balance docsis-enable** **Example:** `Router(config)# cable load-balance docsis-enable`	Enables DOCSIS load balancing on the Cisco CMTS.
Step 4	**cable load-balance docsis-group** *docsis-group-id*	Creates a DOCSIS load balance group on the Cisco CMTS, with the following parameter:

	Command or Action	Purpose
	Example: `Router(config)# cable load-balance docsis-group 1`	The router enters DOCSIS load balancing group configuration mode.
Step 5	**init-tech-list** *tech-list* [**ucc**] **Example:** `Router(config-lb-group)# init-tech-list 1 ucc`	Sets the DCC initialization techniques that the Cisco CMTS can use to load balance cable modems.

Step 6	**policy** {**pcmm** | **ugs** | **us-across-ds** | **pure-ds-load**}
	Example:
	`Router(config-lb-group)# policy us-across-ds`
	`Router(config-lb-group)# policy ugs`
	`Router(config-lb-group)# policy pure-ds-load`
		Selects the modems based on the type of service flow that are balanced.

Step 7	**threshold** {**load** {**minimum** *<1-100>* | *<1-100>*}| **pcmm** *<1-100>* | **stability** *<0-100>* | **ugs** *<1-100>*}
	Example:
	`Router(config-lb-group)# threshold load minimum 10`
	`Router(config-lb-group)# threshold pcmm 70`
	`Router(config-lb-group)# threshold load 10`
	`Router(config-lb-group)# threshold stability 50`
	`Router(config-lb-group)# threshold ugs 70`
		Selects the percentage of use beyond which load balancing occurs.

Step 8	**end** **Example:** `Router# end`	Returns to privileged EXEC mode.

What to do next

To test and verify DCC for load balancing, use the following two commands:

- **test cable dcc**
- **show controllers cable**

These commands are described in the *Cisco CMTS Cable Command Reference* .

Verifying Load Balancing Operations

This section describes how to use certain test and show commands to verify the configuration and operation of the Load Balancing feature or Dynamic Channel Change feature on the Cisco CMTS.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**show cable load-balance** [**group** *n*] [**all** | **load** | **pending** | **statistics** | **target**] **Example:** `Router# show cable load-balance group 1`	Displays real-time statistical and operational information for load balancing operations. If given without any options, this command displays information for the load balancing groups and each cable interface's current load and load balancing status. You can also specify the following options:
Step 3	**test cable dcc** [*mac-addr* | *ip-addr* | *cable-if-src sid*] *cable-if-target uschan* {*ranging-tech* } **Example:** `Router# test cable dcc 0000.394e.4e59`	Tests Dynamic Channel Change (DCC) by moving a target cable modem, as specified by MAC address, IP address, or the primary service ID (SID) value. Applies to a cable modem on the source interface to an upstream channel on a target downstream interface using the initialization technique specified.

Example

This example shows the result of load balancing operations.

```
Router#show cable load all
DOCSIS 2.0 LB Enabled: Yes DOCSIS 3.0 LB Enabled: No
DOCSIS     Status Interval DCC mask  Policy     Method Threshold
Group                      /UCC                DS/US  M/E/U/P/S
1          RE     30       0xF8(0)/N 0          m/m    5/10/70/70/50
12345      GE     30       0xF8(0)/N 0          m/m    5/10/70/70/50
12346      RE     30       0xF8(0)/N 0          m/m    5/10/70/70/50
12347      RE     30       0xF8(0)/N 0          m/m    5/10/70/70/50
12348      RE     30       0xF8(0)/N 0          m/m    5/10/70/70/50
12349      RE     30       0xF8(0)/N 0          m/m    5/10/70/70/50

DOCSIS 3.0 General LB
MD       FN  Group ID   S Intv DCC mask  Policy     Mtd MD-CM-SG  Threshold
                                /UCC                D/U           M/E/U/P/S
Ca8/0/0  1   2147631104 E 30   0x30(2)/N 0          m/m 0x1200301 5/10/70/70/50
Ca8/0/1  3   2147631618 E 30   0x30(2)/N 0          m/m 0x1210301 5/10/70/70/50
Ca8/0/2  5   2147632132 E 30   0x30(2)/N 0          m/m 0x1220401 5/10/70/70/50
Ca8/0/2  6   2147632133 E 30   0x30(2)/N 0          m/m 0x1220402 5/10/70/70/50
Ca8/0/3  7   2147632646 E 30   0x30(2)/N 0          m/m 0x1230501 5/10/70/70/50
Ca8/0/3  8   2147632647 E 30   0x30(2)/N 0          m/m 0x1230502 5/10/70/70/50
Ca8/0/8  2   2147635201 E 30   0x30(2)/N 0          m/m 0x1280201 5/10/70/70/50
Ca8/0/9  4   2147635715 E 30   0x30(2)/N 0          m/m 0x1290201 5/10/70/70/50
```

```
Current load:

DOCSIS load-balancing load
Interface           State   Group      Utilization     Rsvd NBCM  WB/UB Weight
                                                           Total Total
In8/0/0:0(411 MHz)  initial 1          0%(0%/0%)       0%   0     11    37
In8/0/0:0(411 MHz)  initial 2147631104 0%(0%/0%)       0%   0     11    37
Us8/0/0:0           initial 1          0%              0%   0     31    30.7
Us8/0/0:0           initial 2147631104 0%              0%   0     31    30.7
Us8/0/0:1           initial 1          0%              0%   0     31    30.7
Us8/0/0:1           initial 2147631104 0%              0%   0     31    30.7
Us8/0/0:2           initial 2147631104 0%              0%   0     31    30.7
Us8/0/0:2           initial 1          0%              0%   0     31    30.7
Us8/0/0:3           initial 2147631104 0%              0%   0     31    30.7
Us8/0/0:3           initial 1          0%              0%   0     31    30.7
In8/0/0:4(435 MHz)  up      2147635201 0%(0%/0%)       0%   48    11    37
Us8/0/1:0           up      2147635201 0%              0%   15    0     30.7
Us8/0/1:1           up      2147635201 0%              0%   11    0     30.7
Us8/0/1:2           up      2147635201 0%              0%   11    0     30.7
Us8/0/1:3           up      2147635201 0%              0%   11    0     30.7
In8/0/0:8(459 MHz)  initial 1          0%(0%/0%)       0%   0     9     37
In8/0/0:8(459 MHz)  initial 2147631104 0%(0%/0%)       0%   0     9     37
Us8/0/0:0           initial 1          0%              0%   0     31    30.7
Us8/0/0:0           initial 2147631104 0%              0%   0     31    30.7
Us8/0/0:1           initial 1          0%              0%   0     31    30.7
Us8/0/0:1           initial 2147631104 0%              0%   0     31    30.7
Us8/0/0:2           initial 2147631104 0%              0%   0     31    30.7
Us8/0/0:2           initial 1          0%              0%   0     31    30.7
Us8/0/0:3           initial 2147631104 0%              0%   0     31    30.7
Us8/0/0:3           initial 1          0%              0%   0     31    30.7
In8/0/0:12(483 MHz) down    2147635201 0%(0%/0%)       0%   0           0
In8/0/0:16(507 MHz) initial 2147631104 0%(0%/0%)       0%   0     11    37
In8/0/0:16(507 MHz) initial 1          0%(0%/0%)       0%   0     11    37
Us8/0/0:0           initial 2147631104 0%              0%   0     31    30.7
Us8/0/0:0           initial 1          0%              0%   0     31    30.7
Us8/0/0:1           initial 2147631104 0%              0%   0     31    30.7
Us8/0/0:1           initial 1          0%              0%   0     31    30.7
Us8/0/0:2           initial 2147631104 0%              0%   0     31    30.7
Us8/0/0:2           initial 1          0%              0%   0     31    30.7
Us8/0/0:3           initial 2147631104 0%              0%   0     31    30.7
Us8/0/0:3           initial 1          0%              0%   0     31    30.7
In8/0/0:20(531 MHz) down    2147635201 0%(0%/0%)       0%   0           0
In8/0/1:0(555 MHz)  initial 2147631618 0%(0%/0%)       0%   0     12    37
Us8/0/2:0           initial 2147631618 0%              0%   0     19    30.7
Us8/0/2:1           initial 2147631618 0%              0%   0     19    30.7
Us8/0/2:2           initial 2147631618 0%              0%   0     19    30.7
```

Troubleshooting Tips

Problem Packets are dropped when a cable modem moves from one channel to another.

Possible Cause When the **test cable dcc** command is used to move a cable modem from one channel to another with DCC initialization technique 3:

- If the pre-equalization coefficient is enabled, the cable modem moves and packet drop occurs for 5 seconds.
- If the pre-equalization coefficient is disabled, the cable modem moves and packet drop occurs for less than 1 second.

Possible Cause When the **test cable dcc** command is used to move a cable modem from one channel to another with DCC initialization technique 4:

- If the pre-equalization coefficient is enabled, the cable modem moves and packet drop occurs for less than 1 second.
- If the pre-equalization coefficient is disabled, the cable modem moves without any packet drop.

Solution No action is required.

Examples

Use the **show cable load-balance target** command to display the interfaces being used for load balancing, use the **test cable load-balance** command to test whether a cable modem can move between interfaces, and use the **show cable load-balance statistics** command to display the results of the test.

The following example shows how to test whether a specific cable modem responds to both a UCC request and to an upstream channel override to move from one upstream to another in its load balancing group:

```
Router# show cable load-balance target

Target assignments:
Interface            State      Group  Target
Cable1/0/0 (669 MHz) up         1
Cable1/0/0/U0        up         1      Cable1/0/0/U1 [enforce]
Cable1/0/0/U1        up         1

Router# show cable load-balance statistics

Statistics:

Target interface     State      Transfers
                                Complete  Pending   Retries   Failures
Cable1/0/0 (669 MHz) up         15        0         1         0
Cable1/0/0/U0        up         33        0         1         0
Cable1/0/0/U1        up         22        0         2         0

Router# test cable load-balance 0000.394e.4e59

Sending UCC request: Cable1/0/0/U0 --> U1
Waiting for test completion ........
Test results:
    UCC Response: 0.0s
    Initial Ranging: 8.5s
    Ranging Complete: failed.
    Modem replied to DOCSIS ping.
Test summary:
    UCC Response: success rate 100% min 0.0s max 0.0s avg 0.0s
    Initial Ranging: success rate 100% min 8.5s max 8.5s avg 8.5s
Testing US Channel Override: Cable1/0/0/U1 --> U0
Waiting for test completion ...........
Test results:
    Initial Ranging: 8.5s
    Ranging Complete: failed.
    Modem replied to DOCSIS ping.
Test summary:
    UCC Response: success rate 100% min 0.0s max 0.0s avg 0.0s
    Initial Ranging: success rate 100% min 8.5s max 8.5s avg 8.5s

Router# show cable load-balance statistics

Statistics:
```

```
Target interface     State      Transfers
                                Complete  Pending   Retries   Failures
Cable1/0/0 (669 MHz) up         15        0         1         0
Cable1/0/0/U0        up         34        0         1         0
Cable1/0/0/U1        up         23        0         2         0
```

The following example shows how to test whether a specific modem responds to a UCC request to move from one upstream to another in its load balancing group:

```
Router# show cable load-balance statistics

Statistics:

Target interface     State      Transfers
                                Complete  Pending   Retries   Failures
Cable1/0/0 (669 MHz) up         15        0         1         0
Cable1/0/0/U0        up         34        0         1         0
Cable1/0/0/U1        up         23        0         2         0

Router# test cable load-balance 0007.0e01.4129 ucc 1

Sending UCC request: Cable1/0/0/U0 --> U1
Waiting for test completion ........
Test results:
    UCC Response: 0.0s
    Initial Ranging: 10.3s
    Ranging Complete: 11.2s
    Modem replied to DOCSIS ping.
Test summary:
    UCC Response: success rate 100% min 0.0s max 0.0s avg 0.0s
    Initial Ranging: success rate 100% min 10.3s max 10.3s avg 10.3s
    Ranging Complete: success rate 100% min 11.2s max 11.2s avg 11.2s

Router# show cable load-balance statistics

Statistics:

Target interface     State      Transfers
                                Complete  Pending   Retries   Failures
Cable1/0/0 (669 MHz) up         15        0         1         0
Cable1/0/0/U0        up         35        0         1         0
Cable1/0/0/U1        up         24        0         2         0
```

The following example shows information when moving a cable modem to a different upstream channel using DCC initialization technique 1. This example moves the cable modem 0012.17ea.f563 from interface c7/1/0 upstream 1 to interface c7/1/1 upstream 0 using DCC initialization technique 1:

```
Router# show cable modem

MAC Address    IP Address      I/F       MAC         Prim RxPwr  Timing  Num BPI
                                         State       Sid  (dB)   Offset  CPE Enb
 State       Sid  (dB)   Offset  CPE Enb
0012.17ea.f563 12.0.0.2        C7/1/0/U1 online      4    0.00  2449     0   N

Router# test cable dcc 0012.17ea.f563 c7/1/1 0 1

Router# show cable modem

MAC Address    IP Address      I/F       MAC         Prim RxPwr  Timing  Num BPI
                                         State       Sid  (dB)   Offset  CPE Enb
0012.17ea.f563 12.0.0.2        C7/1/1/U0 online      3    0.00  2451     0   N
```

Configuration Examples for Load Balancing

This section provides the following configuration examples:

Example: Configuring Dynamic Channel Change for Load Balancing

The following examples illustrate the working of dynamic load balancing working process in DOCSIS 3.0 cable modems.

Verify configuration:

```
Router#  show cable load-balance docsis-group 1
DOCSIS LB Enabled: Yes
DOCSIS 2.0 LB Enabled: No
DOCSIS 3.0 LB Enabled: Yes
DOCSIS 3.0 Static LB Enabled: No
DOCSIS 3.0 Dynamic Downstream LB Enabled: Yes
DOCSIS Status Interval DCC mask Policy Method Threshold
Group /UCC DS/US M/E/U/P/S
1 RE 60 0x38(2)/N 0 u/u 1/10/70/70/50
```

Verify channel current load:

```
Router#  show cable load-balance docsis-group 1 load wideband
DOCSIS load-balancing wide band load
Interface Size Group Throughput(Kbps)/bw(Mbps) Avg-Util
Wi9/0/0:1 8 1 93324/300 36%
Wi9/0/0:2 8 1 37329/300 39%
Wi9/0/0:3 8 1 74659/300 31%
Wi9/0/0:4 8 1 0/300 13%
Wi9/0/0:5 8 1 9332/300 2%
```

Verify channel overload and target:

```
Router#  show cable load-balance docsis-group 1 target wideband
Interface Bg-Id State Group Target
Wi9/0/0:1 28674 up 1 Wi9/0/0:5 ...
Wi9/0/0:2 28675 up 1 Wi9/0/0:5 ...
Wi9/0/0:3 28676 up 1 Wi9/0/0:5 ...
Wi9/0/0:4 28677 up 1 Wi9/0/0:5
Wi9/0/0:5 28678 up 1
```

Verify channel modem-list:

```
Router#  show cable load-balance docsis-group 1 modem-list wideband
Codes: M - Multicast, U - UGS, P - PCMM, F - Max-Failures, X - eXcluded
L - L2vpn, R - RSVP
Primary WB MAC Address Primary DS RCC-ID Priority MUPFXLR State
Wi9/0/0:1 (10)
c8fb.26a6.c02c In9/0/0:4 1 0 ------- LB_CM_READY
c8fb.26a6.c62c In9/0/0:4 1 0 ------- LB_CM_READY
c8fb.26a6.c706 In9/0/0:4 1 0 ------- LB_CM_READY
c8fb.26a6.c0dc In9/0/0:4 1 0 ------- LB_CM_READY
c8fb.26a6.c53a In9/0/0:4 1 0 ------- LB_CM_READY
```

Verify QAM channel utilization:

```
Router#  show cable load-balance docsis-group 1 rfch-util
Interface Pstate Pending-In Pending-Out Throughput(Kbps) Util
In9/0/0:4 up No No 6517 17
In9/0/0:5 NA No No 6574 17
In9/0/0:6 NA No No 6520 17
```

```
In9/0/0:7 NA No No 6738 17
In9/0/0:8 up No No 8624 22
In9/0/0:9 NA No No 8482 22
In9/0/0:10 NA No No 8353 22
```

Verify channel statistic movement:

```
Router#  show cable load-balance docsis-group 1 statistics wideband
Target interface State Transfers
Complete Pending Total Failures Disabled
Wi9/0/0:1 up 0 0 0 0 0
Wi9/0/0:2 up 0 0 0 0 0
Wi9/0/0:3 up 3 0 3 0 0
Wi9/0/0:4 up 0 0 0 0 0
Wi9/0/0:5 up 9 0 9 0 0
```

The following example of the running configuration illustrates DCC for load balancing.

```
Router# show cable load all

*Nov 11 15:42:18.955: %SYS-5-CONFIG_I: Configured from console by conscable load all
Group  Interval  Method      DCC Init     Threshold
                           Technique    Minimum  Static  Enforce  Ugs   PCMM
1      10        modems      0             5        10%     ---     ---   ---

Current load:

Interface            State      Group Utilization  Reserved Modems Flows Weight
Cable3/0 (0 MHz)     initial    1     0%(0%/0%)    0%       0      0     26

Target assignments:

Interface            State      Group  Target
Cable3/0 (0 MHz)     initial    1

Statistics:

Target interface       State      Transfers
                                  Complete  Pending   Retries   Failures
Cable3/0 (0 MHz)       initial    0         0         0         0
Pending:
Modem          Group  Source interface  Target interface     Retries
```

The following example of the running configuration illustrates DCC for load balancing.

```
Router# show running configuration

Building configuration...
Current configuration : 11889 bytes
!
version 12.3
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$tEvV$8xICVVbFm10hx0hAB7DO90
enable password lab
!
no cable qos permission create
```

```
no cable qos permission update
cable qos permission modems
cable load-balance group 1 threshold load 75 enforce
cable load-balance group 1 threshold stability 75
cable load-balance group 1 policy ugs
cable load-balance group 1 threshold ugs 75
cable load-balance group 1 policy pcmm
cable load-balance group 1 threshold pcmm 75
no aaa new-model
ip subnet-zero
!
!
ip cef
no ip domain lookup
!
!
interface GigabitEthernet0/1
 ip address 10.14.1.130 255.255.0.0
 duplex auto
 speed auto
 media-type rj45
 no negotiation auto
!
interface GigabitEthernet0/2
```

The following example of the show cable load all command illustrates DCC for load balancing.

```
Router# show cable load all

*Nov 11 15:43:39.979: %SYS-5-CONFIG_I: Configured fromconf t
Group  Interval  Method      DCC Init     Threshold
                             Technique    Minimum  Static  Enforce  Ugs   PCMM
1      10        modems      0             5        75%     75%     75%   75%

Current load:

Interface            State      Group Utilization  Reserved Modems Flows Weight
Cable3/0 (0 MHz)     initial    1     0%(0%/0%)    0%       0      0     26

Target assignments:

Interface            State      Group  Target
Cable3/0 (0 MHz)     initial    1

Statistics:

Target interface       State      Transfers
                                  Complete  Pending   Retries   Failures
Cable3/0 (0 MHz)       initial    0         0         0         0

Pending:

Modem          Group  Source interface  Target interface     Retries
```

The following example illustrates a DCC load balancing group with the default DCC initialization technique. This command configures load balancing group 1:

```
Router(config)# cable load-balance group 1 threshold load 10 enforce
```

This configuration creates a dynamic load balancing group with the following default settings:

```
cable load-balance group 1 method modem
cable load-balance group 1 threshold load 10 enforce
cable load-balance group 1 interval 10
cable load-balance group 1 dcc-init-technique 0
```

The following example changes this DCC load balancing configuration to initialization technique 4:

```
Router# cable load-balance group 1 dcc-init-technique 4
```

Note By default, UGS and PCMM policies are not turned on, so that CMs with active voice calls or PCMM calls participate in load balancing.

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

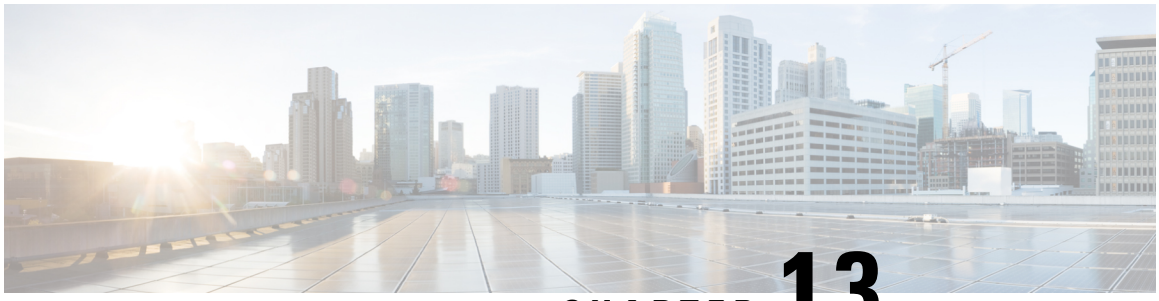
Feature Information for DOCSIS Load Balancing Movements

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 27: Feature Information for DOCSIS Load Balancing Groups

Feature Name	Releases	Feature Information
DOCSIS Load Balancing Movements	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Routers.



CHAPTER 13

DOCSIS 3.0 Downstream Bonding

The DOCSIS 3.0 Downstream Bonding feature helps cable operators offer new, more bandwidth-intensive services by adding one or more additional downstream quadrature amplitude modulation (QAM) channels to the standard broadband DOCSIS system.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 239
- Information About DOCSIS 3.0 Downstream Bonding, on page 240
- How to Configure RCP and RCC Encoding, on page 242
- How to Configure Attribute Masks, on page 251
- How to Enable Service Flow Priority in Downstream Extender Header, on page 255
- Enabling Verbose Reporting for Receive Channel Profiles, on page 257
- Configuration Example for an RCC Template, on page 258
- Additional References, on page 259
- **Feature Information for DOCSIS 3.0 Downstream Bonding** , on page 259

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 28: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About DOCSIS 3.0 Downstream Bonding

DOCSIS 3.0 Downstream Bonding enables high-speed broadband access and helps cable operators offer more bandwidth-intensive services by adding one or more additional downstream quadrature amplitude modulation (QAM) channels to the standard broadband DOCSIS system. This new set of downstream channels is grouped into one larger channel, known as a bonded channel.

Channel bonding combines several RF channels into one virtual channel. Data rates in this virtual channel range from hundreds of megabits to potentially gigabits per second, creating more available bandwidth in the network.

Receive Channel Profile

An RCP is an encoding that represents the receive channels and receive modules of a cable modem. A cable modem communicates to the CMTS one or more RCP encodings within its registration request using either verbose description, which contains complete subtype encoding defined in DOCSIS 3.0, or simple description, which only contains RCP identifiers.

The cable modem reporting method is configurable within the MAC domain and communicated to cable modems via the MDD.

You must define an RCP-ID to describe the cable modem's capabilities for that RCP-ID and to input information about cable modems which are not defined on the system. Once configured the RCP-ID is available to the

entire system since it is not meant to be card specific or mac-domain specific. The path selection module ensures that the RCP ID is accurately transmitted as part of the RCC profile.

The CableLabs MULPI specification defines standard RCPs which are automatically created by the CMTS.

Receive Channel Configuration

A cable modem reports its ability to receive multiple channels with one or more RCP encodings in a REG-REQ or REG-REQ-MP message. Each receive channel profile describes a logical representation of the cable modem’s downstream physical layer in terms of receive channels (RCs) and receive modules (RMs). The CMTS initially configures the cable modem’s receive channels and receive modules with an RCC encoding in the registration response.

This feature supports any arbitrary RCP ID configuration and receive channel configuration on a Cisco cBR Series Converged Broadband Router.

RCC Template

You can configure one or more RCC templates for an RCP. An RCC template configures the physical layer components described by an RCP, including receive modules and receive channels to specific downstream frequencies. The template also specifies the interconnections among receive modules, or between a receive module and a receive channel. An RCC template can be associated only to the cable interface (MAC domain).

A cable modem's RCP ID is matched with an RCC, when RCC templates are configured. A cable modem's RCP ID may be matched with an RCC generated by an RCC template when RCC templates are configured. The path selection module ensures that the RCP ID that is transmitted as part of the RCC profile is accurate.

At time of registration, if there are multiple valid RCCs that can be assigned to the CM after going through the sequence of checks outlined in the CableLabs MULPI specifications then the RCC with the most channels will be the one selected. If there are multiple valid RCCs of equal size then the RCC with the least amount of cable modems will be selected.

Channel Assignment

The CMTS assigns a receive channel configuration encoding to a DOCSIS 3.0-certified cable modem operating in a Multiple Receive Channel (MRC) mode during cable modem registration.

With the implementation of this feature, the DOCSIS 3.0-certified cable modem reports its receiving capabilities and characteristics using the receive channel profile type, length, value (TLV) list in the registration request message. Based on this report, the CMTS assigns an RCC encoding that is compatible with the reported RCP.

Cable modems operating in MRC mode are assigned an RCC encoding associated with an RCP. RCC encodings may be derived from RCC templates or from a wideband-cable interface configuration.

An RCC encoding can also be derived from a wideband interface configuration.

Downstream Traffic Forwarding

DOCSIS 3.0 introduces the concept of assigning downstream service flows of cable modems, which are operating in an MRC mode, to downstream (DS) channels or bonding groups. Forwarding interfaces assigned to service flows (SFs) can be either DS channel interfaces (integrated cable interfaces) or downstream bonding groups (wideband interfaces).

Note Valid interfaces that are available for SF assignment must be a subset of the cable modem's assigned RCC encoding.

Service Flow Priority in Downstream Extended Header

The purpose of the feature is to be able to reflect the traffic priority of downstream packets into the DOCSIS extended header. The priority is derived from the service flow that the packet is mapped to. Priority refers to the service flow priority specified in the CM configuration file, or the Cisco CMTS service class configuration.

The service flow priority can be set using cable modem configuration file, or dynamic configuration.

By default, this feature is disabled on Cisco cBR-8 router, user can use **cable service flow priority** command to enable this feature.

How to Configure RCP and RCC Encoding

The following tasks describe how to configure a receive channel profile and configuration encoding for a receive channel profile:

Configuring the RCP ID

You must configure the RCP IDs with the cable modem capabilities that are not defined in the CMTS. This is done to supplement the standard MULPI RCP IDs already created by the CMTS.

Before you begin

Restrictions

The configurations are subject to RCC Templates and RCP Interactions as follows:

- RCC templates can only be created for an RCP that is already defined on the system. By default the system will contain the RCPs that are specified in the MULPI spec.
- When defining RCC templates for a particular RCP, error checking will be done to ensure that the information being configured in the RCC template does not violate the corresponding RCP information. For example, if the RCP information indicates that there are 2 receive modules then the RCC template configuration will not allow the user to configure more than 2 modules.
- Once an RCP is included in an RCC template users will not be allowed to modify the RCP. Only an RCP which is not being used by any RCC template can be modified
- A valid RCP that can be applied to an rcc-template must contain the following;
    - center-frequency-spacing
    - At least one module which defines the minimum and maximum center frequency range.
    - Rules of inheritance.
    - rcc-template inherit definition from the associated user-defined RCP, such as center-frequency-spacing.
    - rcc-template channel frequencies must fall within the range of the minimum and maximum center frequency per the corresponding RCP module.

- common-module definition is applicable to the rcc-template module referenced with the same index.
- rcc-template module channel frequencies overrides the same channel from the corresponding common-module.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**cable rcp-id** *rcp-id*
	Example:
	`Router(config)#cable rcp-id 00 10 00 01 08`
	`Router(config-rcp)#`
		Defines the RCC template.
		• *rcp-id* - Specifies an RCP ID in Hex.
		This command changes the input mode to the RCC configuration mode.

Step 4	**name** *word* **Example:** `Router(config-rcp)# name rcp-id_1`	**name** —Assigns a name ro the RCP ID • *word*—Use a string to name the RCP ID. **Note** Do not include space between words in the name
Step 5	**center-frequency-spacing** *frequency* **Example:** `Router(config-rcp)#center-frequency-spacing 6`	Assigns a center frequency space to the RCP ID. The valid values are 6 and 8.

Step 6	**module** *module index*
	minimum-center-frequency *Hz*
	maximum-center-frequency *Hz*
	Example:
	`Router(config-rcp)# module 1`
	`minimum-center-frequency 120000000`
	`maximum-center-frequency 800000000`
		Configures a receive module configuration for the selected RCP.
		• *module index* - Specifies the module number for the receivemodule. The valid range is 1 to 12.
		• **minimum-center-frequency** - Specifies the minimum center frequency for the channels of the receivemodule channel.
		• *Hz*- Specifies the center frequency value in Hz. The valid range is from 111000000 to 999000000.
		• **maximum-center-frequency** - Specifies the maximum center frequency for the channels of the receive module channel.

	Command or Action	Purpose
Step 7	**module** *module index* **number-of-adjacent-channels** *Integrer* **Example:** `Router(config-rcp)#module 2 number-of-adjacent-channels 10 Router(config-rcp)#`	Specifies the frequency band for the receive module. The valid values are 1-255.
Step 8	**module** *module index* **connected-module** *module index* **Example:** `Router(config-rcp)# module 1 connected-module 0`	Specifies a receive channel configuration for the selected RCP. • **connected-receive-module**— (Optional) Specifies a nested receive module in the RCC template. Generally, only one receive module is configured for an RCC template. • *module index*—Specifies the module number for the receive module. The valid range is 1 to 12.
Step 9	**number-of-channels** *Number of channel* **Example:** `Router (config-rcp)#number-of-channels 8`	Specifies the number of receive channels in the RCP ID.
Step 10	**primary-capable-channels** *Number of channel* **Example:** `Router(config-rcp)# primary-capable-channels 1`	Specifies the number of receive channels that are defined as primary capable channels.

What to do next

Verify RCP ID configurations using the **show cable rcps** command.

```
Router# show cable rcps
RCP ID  : 00 10 00 01 08
  Name  : rcp-id 1
  Center Frequency Spacing      : 6
  Max number of Channels        : 8
  Primary Capable Channel       : 1
  Number of Modules             : 2
   Module[1]:
     Number of Adjacent Channels: 10
     Minimum Center Frequency-Hz: 111000000
     Maximum Center Frequency-Hz: 999000000
   Module[2]:
     Number of Adjacent Channels: 10
     Minimum Center Frequency-Hz: 120000000
```

```
     Maximum Center Frequency-Hz: 800000000

RCP ID  : 00 10 00 00 02
  Name  : rcp-id 2
  Center Frequency Spacing      : 6
  Max number of Channels        : 2
  Primary Capable Channel       : 1
  Number of Modules             : 1
   Module[1]:
     Number of Adjacent Channels: 10
Minimum Center Frequency-Hz: 111000000
     Maximum Center Frequency-Hz: 867000000
     Connected Module           : 64
```

Configuring the RCC Templates

You must configure an RCC template with a unique RCP ID for a particular CMTS. A valid RCC template consists of a configured RCP ID, RMs, and RCs. There is dependency between the RCC templates and the RCP since information present in the RCP configuration is also present in RCC templates.

Each RCC encoding contains all operational DS channels with their channel parameters, including the frequency match RC attribute specified in the RCC template. An RCC template specifies the intended receive channel assignment in the available DS spectrum.

Note If an RCC template is removed from a MAC domain through configuration, the CMTS removes all RCC encodings derived from the RCC template, and all cable modems assigned to the RCC encoding are marked offline.

Before you begin

At least one RC must be configured as a primary Receive Channel (RC).

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**cable rcc-templates frequency-based** *id*
	Example:
	`Router(config)#cable rcc-templates frequency-based 1`
	`Router(config-rcc-freq-based)#`
		id—Specifies an RCC template. The valid range is 1-64.

	Command or Action	Purpose
Step 4	**rcp-id** *id* **Example:** `Router(config-rcc-freq-based)#rcp-id 00 10 00 01 08`	*id*—Specifies an RCP ID for the RCC template. The valid range is 00 00 00 00 00 to FF FF FF FF. By default, the RCP ID is set to 00 00 00 00 00.
Step 5	**common-module** *module-index* **channel** *grouplist* **start-frequency** *Hz* **Example:** `Router(config-rcc-freq-based)# common-module 1 channels 0-6 start-frequency 555000000`	Specifies module configurations that are common for a selected set of channels assigned to the selected RCP ID. • *Module-index*—Specifies the index value for the receive module. The valid range is 1 to 12. • **channels**—Specifies the list of channels to which the common configurations apply. • *grouplist*—Specifies the list of channels to which a specific list of configurations apply. The range of values are 1-64. • **start-frequency** —Specifies the start frequency value in Hz. • *Hz*—Specifies the frequency value for the start frequency for the common module. Thevalidrangeisfrom 111000000 to 999000000.
Step 6	**rcc-template** *Id* **Example:** `Router(config-rcc-freq-based)# rcc-template 1`	Specifies an RCC template ID to configure the selected RCC template. • *Id*—Specifies the ID of the RCC template. The valid range is from 1-8.
Step 7	**cm-attribute-mask** *value* **Example:** `Router (config-rcc-freq-based-tmplt)# cm-attribute-mask 1`	(Optional) Configured to be used to match against the cm attribute mask define in CM 's configuration file. • *value*—The valid range is 00 00 00 00 00 to FF FF FF FF.
Step 8	**module***module-index* **channel** *grouplist* **start-frequency** *Hz*. **Example:** `Router(config-rcc-freq-based)# common-module 1 channels 0-6 start-frequency 555000000`	Specifies module configurations that are common for a selected set of channels assigned to the selected RCP ID. • *Module-index*—Specifies the index value for the receive module. The valid range is 1 to 12.

	Command or Action	Purpose
		• **channels**—Specifies the list of channels to which the common configurations apply. • *grouplist*—Specifies the list of channels to which a specific list of configurations apply. The range of values are 1-64. • **start-frequency** —Specifies the start frequency value in Hz. • *Hz*—Specifies the frequency value for the start frequency for the common module. The valid range is from 111000000 to 999000000. Repeat Step 3 and Step 7 to configure other frequency based RCC templates.

What to do next

The following configuration examples show the cable rcc-template configuration:

```
cable rcc-templates frequency-based 2
  rcp-id 00 10 00 01 08
  common-module 1 channels 1-4 start-frequency 381000000
  rcc-template 1
  module 1 channels 5-8 start-frequency 501000000
  rcc-template 2
  module 1 channels 5-8 start-frequency 669000000
  rcc-template 3

cable rcc-templates frequency-based 1
  rcp-id 00 10 00 01 08
  rcc-template 1
  cm-attribute-mask 2
  module 1 channels 1-4 start-frequency 381000000
  module 2 channels 5-8 start-frequency 501000000
  rcc-template 2
  module 1 channels 1-4 start-frequency 381000000
  module 2 channels 5-8 start-frequency 669000000
  rcc-template 3
  module 1 channels 1-4 start-frequency 381000000
```

After defining an RCC template, you must assign the template to a cable interface.

Assigning an RCC Template to a MAC Domain (Cable Interface)

The CMTS derives an RCC or RCCs from the RCC template for each MAC Domain Downstream Service Group (MD-DS-SG).

The following information is required for RCC assignment to cable modems:

- RCC templates assigned to the MAC domain.
- DS channel physical parameters including frequency and connected-receive-module index .

- DS channel primary capable indicator.
- DS channel membership to the MD-DS-SG.
- Cable modem membership to the MD-DS-SG.

This section describes how to assign an RCC template to a MAC Domain.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** *slot* / *subslot* / *port* **Example:** `Router(config)# interface cable 1/0/0`	Enters MAC domain configuration mode. • *slot* —Specifies the chassis slot number of the interface line card. • *subslot* —Specifies the secondary slot number of the interface line card. Valid subslot is 0. • *MD index* —Specifies the MAC Domain index number. Valid values are 0-15.
Step 4	**cable rcc-template frequency-based** *Id* **Example:** `Router(config-if)# cable rcc-template frequency-based 1`	Assigns the RCC template to the specified cable interface. • *Id*—Specifies the template you want to assign to the cable interface. The valid range is from 1 to 64.

What to do next

Verify RCC template binding to MD.

The following example shows the RCC template binding using the **show cable mac-domain rcc**

```
Router#show cable mac-domain c1/0/0 rcc

RCC-ID   RCP               RCs MD-DS-SG CMs   WB/RCC-TMPL
1        00 00 00 00 00    4   0        2     WB (Wi1/0/0:0)
2        00 00 00 00 00    4   0        2     WB (Wi1/0/0:1)
3        00 00 00 00 00    4   0        0     WB (Wi1/0/1:2)
4        00 00 00 00 00    4   0        0     WB (Wi1/0/2:3)
8        00 10 00 01 08    8   5        0     RCC-TMPL (1:1)
9        00 10 00 01 08    8   5        0     RCC-TMPL (1:2)
```

```
10        00 10 00 01 08      4   5         0     RCC-TMPL (1:3)
14        00 10 00 01 08      8   5         0     RCC-TMPL (2:1)
15        00 10 00 01 08      8   5         0     RCC-TMPL (2:2)
16        00 10 00 01 08      4   5         0     RCC-TMPL (2:3)
```

The following example shows the RCC template binding using the **show cable mac-domain rcc** *id* command.

```
Router#show cable mac-domain c1/0/0 rcc 8

RCC ID                  : 8
RCP                     : 00 10 00 01 08
Created Via             : rcc-template - 1:1
CM attribute mask       : 0x2
Receive Channels        : 8
  Receive Channel       : 1
    Center Frequency    : 381000000
    Primary Capability  : YES
    Receive Module Conn : 1
  Receive Channel       : 2
    Center Frequency    : 387000000
    Primary Capability  : NO
    Receive Module Conn : 1
  Receive Channel       : 3
    Center Frequency    : 393000000
    Primary Capability  : NO
    Receive Module Conn : 1
  Receive Channel       : 4
    Center Frequency    : 399000000
    Primary Capability  : NO
    Receive Module Conn : 1
  Receive Channel       : 5
    Center Frequency    : 501000000
    Primary Capability  : NO
    Receive Module Conn : 2
  Receive Channel       : 6
    Center Frequency    : 507000000
    Primary Capability  : NO
    Receive Module Conn : 2
  Receive Channel       : 7
    Center Frequency    : 513000000
    Primary Capability  : NO
    Receive Module Conn : 2
  Receive Channel       : 8
    Center Frequency    : 519000000
    Primary Capability  : NO
    Receive Module Conn : 2
Receive Modules         : 2
  Receive Module        : 1
    First Frequency     : 381000000
  Receive Module        : 2
    First Frequency     : 501000000


Router#show cable mac-domain c9/0/2 rcc 9

RCC ID                  : 9
RCP                     : 00 10 00 01 08
Created Via             : rcc-template - 1:2
CM attribute mask       : 0x0
Receive Channels        : 8
  Receive Channel       : 1
    Center Frequency    : 381000000
    Primary Capability  : YES
```

```
    Receive Module Conn : 1
  Receive Channel       : 2
    Center Frequency    : 387000000
    Primary Capability  : NO
    Receive Module Conn : 1
  Receive Channel       : 3
    Center Frequency    : 393000000
    Primary Capability  : NO
    Receive Module Conn : 1
  Receive Channel       : 4
    Center Frequency    : 399000000
    Primary Capability  : NO
    Receive Module Conn : 1
  Receive Channel       : 5
    Center Frequency    : 669000000
    Primary Capability  : NO
    Receive Module Conn : 2
  Receive Channel       : 6
    Center Frequency    : 675000000
    Primary Capability  : NO
    Receive Module Conn : 2
  Receive Channel       : 7
    Center Frequency    : 681000000
    Primary Capability  : NO
    Receive Module Conn : 2
  Receive Channel       : 8
    Center Frequency    : 687000000
    Primary Capability  : NO
    Receive Module Conn : 2
Receive Modules         : 2
  Receive Module        : 1
    First Frequency     : 381000000
  Receive Module        : 2
    First Frequency     : 669000000

Router#show cable mac-domain c1/0/0 rcc 10

RCC ID                  : 10
RCP                     : 00 10 00 01 08
Created Via             : rcc-template - 1:3
CM attribute mask       : 0x0
Receive Channels        : 4
  Receive Channel       : 1
    Center Frequency    : 381000000
    Primary Capability  : YES
    Receive Module Conn : 2
  Receive Channel       : 2
    Center Frequency    : 387000000
    Primary Capability  : NO
    Receive Module Conn : 2
  Receive Channel       : 3
    Center Frequency    : 393000000
    Primary Capability  : NO
    Receive Module Conn : 2
  Receive Channel       : 4
    Center Frequency    : 399000000
    Primary Capability  : NO
    Receive Module Conn : 2
Receive Modules         : 1
  Receive Module        : 2
    First Frequency     : 381000000
```

Verifying the RCC Configuration

To verify the runtime RCCs on a cable interface, use the **show cable mac-domain rcc** command.

```
Router#show cable mac-domain c1/0/0 rcc

RCC-ID   RCP               RCs MD-DS-SG CMs   WB/RCC-TMPL
1        00 00 00 00 00    4   0        2     WB (Wi1/0/0:0)
2        00 00 00 00 00    4   0        2     WB (Wi1/0/0:1)
3        00 00 00 00 00    4   0        0     WB (Wi1/0/1:2)
4        00 00 00 00 00    4   0        0     WB (Wi1/0/2:3)
8        00 10 00 01 08    8   5        0     RCC-TMPL (1:1)
9        00 10 00 01 08    8   5        0     RCC-TMPL (1:2)
10       00 10 00 01 08    4   5        0     RCC-TMPL (1:3)
14       00 10 00 01 08    8   5        0     RCC-TMPL (2:1)
15       00 10 00 01 08    8   5        0     RCC-TMPL (2:2)
16       00 10 00 01 08    4   5        0     RCC-TMPL (2:3)
```

Note A zero (0) value in the RCP or MD-DS-SG field indicates that the RCC encoding is configured directly through a wideband interface configuration and not through any RCC template.

How to Configure Attribute Masks

DOCSIS 3.0 introduces the concept of assigning service flows to channels or bonding groups based on binary attributes. The attribute masks configured on a cable, modular, integrated or wideband interface are called provisioned attribute masks.

The two types of attributes are as follows:

- Specification-defined attributes—Contain default values based on the characteristics of the channel or bonding group.
- Operator-defined attributes—Default to zero.

The operator can configure a provisioned attribute mask for each channel and provisioned bonding group to assign values to the operator-defined binary attributes. The operator can also assign new values to override the default values of the specification-defined attributes.

The operator can configure a required attribute mask and a forbidden attribute mask for a service flow in the cable modem configuration file. These required and forbidden attribute masks are optionally provided on the DOCSIS 3.0 service flows and are matched with the provisioned attribute masks of the interfaces.

Each service flow is optionally configured with the following TLV parameters:

- Service flow required attribute mask—To configure this, assign a service flow to a channel that has a 1-bit in all positions of its provisioned attribute mask corresponding to the 1-bit in the service flow required attribute mask.
- Service flow forbidden attribute mask—To configure this, assign a service flow to a channel that has a 0-bit in all positions of its provisioned attribute mask corresponding to the 1-bit in the service flow forbidden attribute mask.

Additionally, in a cable modem-initiated dynamic service request, the cable modem can include a required attribute mask and a forbidden attribute mask for a service flow. The CMTS assigns service flows to channels or bonding groups so that all required attributes are present and no forbidden attributes are present in the cable modem configuration file.

The table below lists the supported binary attributes for channels and bonding groups.

Table 29: Binary Attributes

Bit Position	Definition
Bit 0	Bonded—This bit is zero for all individual channel interfaces and one for all bonding groups.
Bit 1	Low latency—This bit is set when the interface can provide relatively low latency service. This bit is set to zero for all channels, and left up to the operator to define.
Bit 2	High availability—This bit is set to zero for all channels, and left up to the operator to define.
Bit 3:15	Reserved—Set to zero.
Bit 16:31	Operator defined—Set to zero by default.

You can configure provisioned attribute masks for cable, integrated cable, wideband cable, and modular cable interfaces.

Prerequisites

- To assign an interface to a wideband cable modem's service flow, the interface must be a subset of the cable modem's RCC.
- To assign a service flow to an integrated cable (IC) channel, the corresponding integrated cable interface must be configured and operational.

Restrictions

- The service flow from a narrowband cable modem is always assigned to the primary interface of the cable modem. No attribute checking is performed in this case.

This section describes the following:

Configuring Provisioned Attributes for an Integrated Cable Interface

The default provisioned attribute is zero for an integrated cable interface.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**interface integrated-cable** {*slot/port* | *slot/subslot/port*}:*rf-channel* **Example:** `Router(config)# interface integrated-cable 1/0/0:0`	Specifies the cable interface line card on a Cisco CMTS router: • *slot*—Chassis slot number of the cable interface line card. • *subslot*—subslot number of the cable interface line card. Valid subslot is always 0. • *port*—Downstream port number. • *rf-channel*—RF channel number with a range of 0 to 3.
Step 4	**cable attribute-mask** *mask* **Example:** `Router(config-if)# cable attribute-mask 800000ff`	Specifies the mask for the interface.

Configuring Provisioned Attributes for a Wideband Cable Interface

The default provisioned attribute is 0x80000000 for a wideband cable interface, and the zero bit is automatically added to the wideband cable interface whenever an attribute is configured for that interface.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface wideband-cable** {*slot/port* | *slot/subslot/port*}:*wideband-channel* **Example:** `Router(config)# interface wideband-cable 1/0/1:4`	Specifies the wideband cable interface and enters interface configuration mode:
Step 4	**cable downstream attribute-mask** *mask* **Example:**	Specifies the mask for the interface.

	Command or Action	Purpose
	`Router(config-if)# cable downstream attribute-mask 800000ff`	

Verifying the Attribute-Based Service Flow Assignments

To verify the attribute-based assignment of service flows on a cable interface, use the **show interface cable service-flow** or **show interface wideband-cable service-flow** command as shown in the following example:

```
Router# show interface cable 3/0 service-flow

Sfid  Sid   Mac Address     QoS  Param Index Type  Dir  Curr  Active  DS-ForwIf/
                           Prov Adm   Act             State Time    US-BG/CH
17    4     001c.ea37.9aac  3    3     3     P     US   act   13h21m  CH 3
18    N/A   001c.ea37.9aac  4    4     4     P     DS   act   13h21m  Wi3/0:0
21    6     001c.ea37.9b5a  3    3     3     P     US   act   13h21m  CH 4
22    N/A   001c.ea37.9b5a  4    4     4     P     DS   act   13h21m  Wi3/0:0
23    7     0016.925e.654c  3    3     3     P     US   act   13h21m  CH 3
24    N/A   0016.925e.654c  4    4     4     P     DS   act   13h21m  In3/0:0

Router# show interface wideband-cable 5/1:0 service-flow
Sfid  Sid   Mac Address     QoS Param Index Type  Dir Curr   Active  DS-ForwIf/
                           Prov  Adm  Act            State  Time    US-BG/CH
3     8193  ffff.ffff.ffff   3     3    3   S(s)  DS  act    2h06m   Wi5/1:0
```

The table below shows descriptions for the fields displayed by this command:

Table 30: show interface cable service-flow Field Descriptions

Field	Description
Sfid	Identifies the service flow identification number. **Note** Primary service flow IDs are displayed even for offline cable modems because they are needed for modem re-registration.
Sid	Identifies the service identification number (upstream service flows only).
Mac Address	Identifies the MAC address for the cable modem.
QoS Parameter Index Prov	Identifies the QoS parameter index for the provisioned state of this flow.
QoS Parameter Index Adm	Identifies the QoS parameter index for the Admitted state of this flow.
QoS Parameter Index Act	Identifies the QoS parameter index for the Active state of this flow.
Type	Indicates if the service flow is the primary flow or a secondary service flow. Secondary service flows are identified by an “S” (created statically at the time of registration, using the DOCSIS configuration file) or “D” (created dynamically by the exchange of dynamic service messages between the cable modem and CMTS).
Dir	Indicates if this service flow is DS or US.

Field	Description
Curr State	Indicates the current run-time state of the service flow.
Active Time	Indicates the length of time this service flow has been active.
DS-ForwIf/US-BG/CH BG/DS	Indicates the bonding group ID or the downstream RFID of the forwarding interface assigned to the downstream service flow.

How to Enable Service Flow Priority in Downstream Extender Header

The following tasks describe how to enable service flow priority in downstream extender header:

Enabling Service Flow Priority in Downstream Extender Header

This section describes how to enable service flow priority in downstream extender header on the Cisco cBR-8 routers:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable service flow priority** **Example:** `Router(config)# cable service flow priority`	Enables the service flow priority in downstream extender header.

Verifying the Enablement of the Service Flow Priority in Downstream Extended Header

To verify the enablement of the service flow priority in downstream extended header, use the **show running-config | in service flow** or **show cable modem** [*ip-address* | *mac-address*] **verbose** command as shown in the following example:

```
Router# show running-config | in service flow
cable service flow priority
```

```
Router# show cable modem 100.1.2.110 verbose

MAC Address                          : 0025.2e2d.74f8
IP Address                           : 100.1.2.110
IPv6 Address                         : 2001:420:3800:909:7964:98F3:7760:ED2
Dual IP                              : Y
Prim Sid                             : 1
Host Interface                       : C3/0/0/U0
MD-DS-SG / MD-US-SG                  : N/A / N/A
MD-CM-SG                             : 0x900000
Primary Downstream                   : In3/0/0:32 (RfId : 12320, SC-QAM)
Wideband Capable                     : Y
DS Tuner Capability                  : 8
RCP Index                            : 6
RCP ID                               : 00 00 00 00 00
Downstream Channel DCID RF Channel : 191   3/0/0:32 (SC-QAM)
UDC Enabled                          : N
US Frequency Range Capability        : Standard (5-42 MHz)
Extended Upstream Transmit Power     : 0dB
Multi-Transmit Channel Mode          : N
Upstream Channel                     : US0
Ranging Status                       : sta
Upstream SNR (dB)                    : 39.8
Upstream Data SNR (dB)               : 36.12
Received Power (dBmV)                : -1.00
Timing Offset              (97.6 ns): 1799
Initial Timing Offset                : 1799
Rng Timing Adj Moving Avg(0.381 ns): 0
Rng Timing Adj Lt Moving Avg         : 0
Rng Timing Adj Minimum               : 0
Rng Timing Adj Maximum               : 0
Pre-EQ Good                          : 0
Pre-EQ Scaled                        : 0
Pre-EQ Impulse                       : 0
Pre-EQ Direct Loads                  : 0
Good Codewords rx                    : 8468
Corrected Codewords rx               : 0
Uncorrectable Codewords rx           : 0
Phy Operating Mode                   : atdma
sysDescr                             :
Downstream Power                     : 0.00 dBmV (SNR = ----- dB)
MAC Version                          : DOC3.0
QoS Provisioned Mode                 : DOC1.1
Enable DOCSIS2.0 Mode                : Y
Service Flow Priority                : N
Modem Status                         : {Modem= online, Security=disabled}
Capabilities                         : {Frag=Y, Concat=Y, PHS=Y}
Security Capabilities                : {Priv=, EAE=N, Key_len=}
L2VPN Capabilities                   : {L2VPN=N, eSAFE=N}
L2VPN type                           : {CLI=N, DOCSIS=N}
Sid/Said Limit                       : {Max US Sids=16, Max DS Saids=15}
Optional Filtering Support           : {802.1P=N, 802.1Q=N, DUT=N}
Transmit Equalizer Support           : {Taps/Symbol= 1, Num of Taps= 24}
CM Capability Reject                : {15,22,23,24,25,26,27,28,29,35,36,38}
Flaps                                : 3(Oct 8  16:22:23)
Errors                               : 0 CRCs, 0 HCSes
Stn Mtn Failures                     : 0 aborts, 2 exhausted
Total US Flows                       : 1(1 active)
Total DS Flows                       : 1(1 active)
Total US Data                        : 294 packets, 25903 bytes
Total US Throughput                  : 143 bits/sec, 0 packets/sec
Total DS Data                        : 91 packets, 10374 bytes
Total DS Throughput                  : 0 bits/sec, 0 packets/sec
```

```
LB group ID assigned               : 1
LB group ID in config file         : N/A
LB policy ID                       : 0
LB policy ID in config file        : 0
LB priority                        : 0
Tag                                : d30
Required DS Attribute Mask         : 0x0
Forbidden DS Attribute Mask        : 0x0
Required US Attribute Mask         : 0x0
Forbidden US Attribute Mask        : 0x0
Service Type ID                    :
Service Type ID in config file     :
Active Classifiers                 : 0 (Max = NO LIMIT)
CM Upstream Filter Group           : 0
CM Downstream Filter Group         : 0
CPE Upstream Filter Group          : 0
CPE Downstream Filter Group        : 0
DSA/DSX messages                   : permit all
Voice Enabled                      : NO
DS Change Times                    : 0
Boolean Services                   : 0
CM Energy Management Capable       : N
CM Enable Energy Management        : N
CM Enter Energy Management         : NO
Battery Mode                       : N
Battery Mode Status                :
Number of Multicast DSIDs Support  : 16
MDF Capability Mode                : 2
IGMP/MLD Version                   : MLDv2
FCType10 Forwarding Support        : Y
Features Bitmask                   : 0x0
Total Time Online                  : 6h00m   (6h00m    since last counter reset)
CM Initialization Reason           : POWER_ON
```

Enabling Verbose Reporting for Receive Channel Profiles

A receive channel profile is an encoding that represents the receive channels and receive modules of a cable modem. A cable modem communicates to the CMTS one or more RCP encodings within its registration request using either verbose description, which contains complete subtype encodings defined in DOCSIS 3.0, or simple description, which only contains RCP identifiers.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** {*slot/port* | *slot*/*subslot/port*} **Example:**	Specifies the cable interface line card on a Cisco CMTS router:

	Command or Action	Purpose
	`Router(config)# interface cable7/0/0`	• *slot*—Chassis slot number of the cable interface line card. • *subslot*—subslot number of the cable interface line card. Valid subslot is 0. • *port*—Downstream port number.
Step 4	**cable rcp-control verbose** **Example:** `Router(config-if)# cable rcp-control verbose`	Enables RCP reporting with verbose description.

Configuration Example for an RCC Template

The following sample shows an RCP ID configuration:

```
...
!
cable rcp-id 00 10 00 01 08
  center-frequency-spacing 6
  module 1 minimum-center-frequency 120000000 maximum-center-frequency 800000000 module 1
number-of-adjacent-channels 10
  module 2 minimum-center-frequency 120000000 maximum-center-frequency 800000000 module 2
 number-of-adjacent-channels 10
  number-of-channels 8
  primary-capable-channels 1
!
```

The following sample shows an RCC template configuration:

```
...
!
cable rcc-templates frequency-based 1
  rcp-id 00 10 00 01 08
  rcc-template 1
  cm-attribute-mask 2
  module 1 channels 1-4 start-frequency 381000000
  module 2 channels 5-8 start-frequency 501000000
  rcc-template 2
  module 1 channels 1-4 start-frequency 381000000
  module 2 channels 5-8 start-frequency 669000000
  rcc-template 3
  module 1 channels 1-4 start-frequency 381000000
!
```

The following sample shows an RCC template configuration using the **common-module** option:

```
...
!
cable rcc-templates frequency-based 2
   rcp-id 00 10 00 01 08
   common-module 1 channels 1-4 start-frequency 381000000
```

```
   rcc-template 1
   module 1 channels 5-8 start-frequency 501000000
   rcc-template 2
   module 1 channels 5-8 start-frequency 669000000
   rcc-template 3
!
```

The following sample shows the assignment of an RCC template to MAC Domain:

```
...
!
configure terminal
interface c1/0/0
 cable rcc-templates frequency-based 1
 end

...
```

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

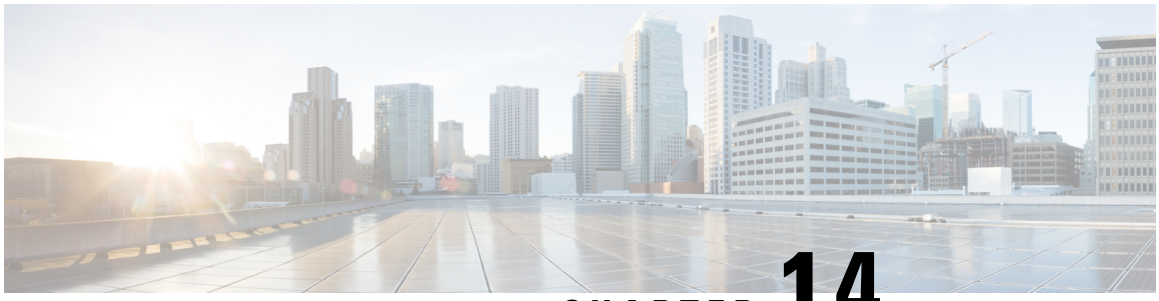
Feature Information for DOCSIS 3.0 Downstream Bonding

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 31: Feature Information for Downstream Interface Configuration

Feature Name	Releases	Feature Information
DOCSIS 3.0 Downstream Bonding	Cisco IOS-XE 3.15.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router .
Service Flow Priority in Downstream Extended Header	Cisco IOS-XE 3.17.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router .



CHAPTER 14

DOCSIS 2.0 A-TDMA Modulation Profiles

This document describes the DOCSIS 2.0 A-TDMA services feature, which provides support for DOCSIS 2.1 Advanced Time Division Multiple Access (A-TDMA) upstream modulation profiles on the router. This feature supplements the existing support for DOCSIS 1.0 and DOCSIS 1.1 Time Division Multiple Access (TDMA) modulation profiles.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 261
- Prerequisites for DOCSIS 2.0 A-TDMA Modulation Profiles, on page 262
- Restrictions for DOCSIS 2.0 A-TDMA Services, on page 263
- Information About DOCSIS 2.0 A-TDMA Services, on page 263
- How to Configure DOCSIS 2.0 A-TDMA Services, on page 266
- Monitoring the DOCSIS 2.0 A-TDMA Services, on page 270
- Configuration Examples for DOCSIS 2.0 A-TDMA services, on page 272
- Additional References, on page 276
- Feature Information for DOCSIS 2.0 A-TDMA Modulation Profile , on page 277

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 32: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for DOCSIS 2.0 A-TDMA Modulation Profiles

- The cable physical plant must be capable of supporting the higher-bandwidth DOCSIS 2.0 A-TDMA modulation profiles.
- Cable modems must be DOCSIS-compliant. If cable modems go offline, or appear to be online but do not pass traffic when in the mixed TDMA/A-TDMA mode, upgrade the modem software to a DOCSIS-compliant version.
- The following are required to support the DOCSIS 2.0 A-TDMA features:
  - Cable modems must be DOCSIS 2.0 capable.
  - The DOCSIS configuration file for a DOCSIS 2.0 cable modem must either omit the DOCSIS 2.0 Enable field (TLV 39), or it must set TLV 39 to 1 (enable). If you set TLV 39 to 0 (disable), a DOCSIS 2.0 CM uses the TDMA mode.
  - The upstream must be configured for either A-TDMA-only or mixed TDMA/A-TDMA mode. To use the 6.4 MHz channel width, the upstream must be configured for A-TDMA-only mode.
- Complete a basic configuration of the router; this includes, at a minimum, the following tasks:
  - Configure a host name and password for the router.
  - Configure the router to support Internet Protocol (IP) operations.
  - Install and configure at least one WAN adapter to provide backbone connectivity.
- Determine a channel plan for the router and all of its cable interfaces.

- Verify that your headend site includes all necessary servers to support DOCSIS and Internet connectivity, including DHCP, ToD, and TFTP servers.
- The system clock on the router should be set to a current date and time to ensure that system logs have the proper timestamp and to ensure that the BPI+ subsystem uses the correct timestamp for verifying cable modem digital certificates.

Restrictions for DOCSIS 2.0 A-TDMA Services

- Does not support virtual channels, as described in DOCSIS 2.0 specification.
- Does not support Synchronous Code Division Multiple Access (S-CDMA) channels.
- Changing the DOCSIS mode of an upstream takes all cable modems on that upstream offline, which forces the cable modems to reregister, so that the CMTS can determine the capabilities of the cable modems on the new channels.

Information About DOCSIS 2.0 A-TDMA Services

DOCSIS 2.0 A-TDMA services improve the maximum upstream bandwidth on existing DOCSIS 1.0 and DOCSIS 1.1 cable networks by providing a number of advanced PHY capabilities that have been specified by the new DOCSIS 2.0 specifications.

DOCSIS 2.0 A-TDMA services incorporate the following advantages and improvements of DOCSIS 2.0 networks:

- Builds on existing DOCSIS cable networks by providing full compatibility with existing DOCSIS 1.0 and DOCSIS 1.1 cable modems. (The registration response (REG-RSP) message contains the DOCSIS version number to identify each cable modem's capabilities.)
- Upstreams can be configured for three different modes to support different mixes of cable modems:
  - An upstream can be configured for TDMA mode to support only DOCSIS 1.0 and DOCSIS 1.1 cable modems.
  - An upstream can be configured for A-TDMA mode to support only DOCSIS 2.0 cable modems.
  - An upstream can be configured for a mixed, TDMA/A-TDMA mode, to support both DOCSIS 1.0/DOCSIS 1.1 and DOCSIS 2.0 cable modems on the same upstream.

Note DOCSIS 2.0 A-TDMA cable modems will not register on a TDMA upstream if an A-TDMA or mixed upstream exists in the same MAC domain, unless the CMTS explicitly switches the cable modem to another upstream using an Upstream Channel Change (UCC) message. DOCSIS 1.0 and DOCSIS 1.1 cable modems cannot register on an A-TDMA-only upstream.

- A-TDMA mode defines new interval usage codes (IUC) of A-TDMA short data grants, long data grants, and Unsolicited Grant Service (UGS) grants (IUC 9, 10, and 11) to supplement the existing DOCSIS 1.1 IUC types.
- Increases the maximum channel capacity for A-TDMA upstreams to 30 Mbps per 6 MHz channel.

- A-TDMA and mixed modes of operation provide higher bandwidth on the upstream using new 32-QAM and 64-QAM modulation profiles, while retaining support for existing 16-QAM and QPSK modulation profiles. In addition, an 8-QAM modulation profile is supported for special applications.
- Supports a minislot size of 1 tick for A-TDMA operations.
- Increases channel widths to 6.4 MHz (5.12 Msymbol rate) for A-TDMA operations.
- A-TDMA and mixed modes of operation provide a more robust operating environment with increased protection against ingress noise and other signal impairments, using a number of new features:
  - Uses to a symbol (T)-spaced adaptive equalizer structure to increase the equalizer tap size to 24 taps, compared to 8 taps in DOCSIS 1.x mode. This allows operation in the presence of more severe multipath and microreflections, and can accommodate operation near band edges where group delay could be a problem.
  - Supports new QPSK0 and QPSK1 preambles, which provide improved burst acquisition by performing simultaneous acquisition of carrier and timing lock, power estimates, equalizer training, and constellation phase lock. This allows shorter preambles, reducing implementation loss.
  - Increases the forward error correction (FEC) T-byte size to 16 bytes per Reed Solomon block (T=16) with programmable interleaving.

Modes of Operation

Depending on the configuration, the DOCSIS 2.0 A-TDMA Service feature supports either DOCSIS or Euro-DOCSIS operation:

- DOCSIS cable networks are based on the ITU J.83 Annex B physical layer standard and Data-over-Cable Service Interface Specifications (DOCSIS, Annex B) specification, which use 6 MHz National Television Systems Committee (NTSC) channel plans. In this mode, the downstream uses a 6 MHz channel width in the 85 to 860 MHz frequency range, and the upstream supports multiple channel widths in the 5 to 42 MHz frequency range.
- EuroDOCSIS cable networks are based on the ITU J.112 Annex A physical layer standard and European DOCSIS (EuroDOCSIS, Annex A) specification, which use 8 MHz Phase Alternating Line (PAL) and Systeme Electronique Couleur Avec Memoire (SECAM) channel plans. In this mode, the downstream uses an 8 MHz channel width in the 85 to 860 MHz frequency range, and the upstream supports multiple channel widths in the 5 to 65 MHz frequency range.

Note The difference between DOCSIS and EuroDOCSIS is at the physical layer. To support a DOCSIS or EuroDOCSIS network requires the correct configuration of the DOCSIS 2.0 A-TDMA Service card, as well as upconverters, diplex filters, and other equipment that supports the network type.

The table below shows the maximum supported DOCSIS 1.1 data rates.

Table 33: Maximum DOCSIS 1.1 Data Rates

Upstream Channel Width	Modulation Scheme	Baud Rate Sym/sec	Maximum Raw Bit Rate Mbit/sec
3.2 MHz	16-QAM QPSK	2.56 M	10.24 5.12
1.6 MHz	16-QAM QPSK	1.28 M	5.12 2.56

The table below shows the maximum supported DOCSIS 2.0 (A-TDMA-mode) data rates.

Table 34: Maximum DOCSIS 2.0 (A-TDMA-mode) Data Rates

Upstream Channel Width	Modulation Scheme	Baud Rate Sym/sec	Maximum Raw Bit Rate Mbit/sec
6.4 MHz	64-QAM	5.12 M	30.72
	32-QAM		25.60
	16-QAM		20.48
	8-QAM		15.36
	QPSK		10.24
3.2 MHz	64-QAM	2.56 M	15.36
	32-QAM		12.80
	16-QAM		10.24
	8-QAM		7.68
	QPSK		5.12
1.6 MHz	64-QAM	1.28 M	7.68
	32-QAM		6.40
	16-QAM		5.12
	8-QAM		3.84
	QPSK		2.56

Modulation Profiles

To simplify the administration of A-TDMA and mixed TDMA/A-TDMA modulation profiles, the DOCSIS 2.0 A-TDMA Service feature provides a number of preconfigured modulation profiles that are optimized for different modulation schemes. We recommend using these preconfigured profiles.

Each mode of operation also defines a default modulation profile that is automatically used when a profile is not specifically assigned to an upstream. The default modulation profiles cannot be deleted. The table below lists the valid ranges according to cable interface and modulation type:

Table 35: Allowable Ranges for Modulation Profiles

Cable Interface	DOCSIS 1.X (TDMA)	Mixed DOCSIS 1.X/2.0	DOCSIS 2.0 (A-TDMA)
Cisco cBR-8 CCAP Line Cards	1 to 400 (default is 21)	1 to 400 (default 121)	1 to 400 (default 221)

Benefits

The DOCSIS 2.0 A-TDMA Service feature provides the following benefits to cable service providers and their partners and customers:

- Full compatibility with DOCSIS 1.0 and DOCSIS 1.1 cable modems (CMs) and cable modem termination systems (CMTS).
- Additional channel capacity in the form of more digital bits of throughput capacity in the upstream path.
- Increased protection against electronic impairments that occur in cable systems, allowing for a more robust operating environment.

How to Configure DOCSIS 2.0 A-TDMA Services

This section contains the following:

Creating Modulation Profiles

This section describes how to create modulation profiles for the different modes of DOCSIS operations, using the preconfigured modulation profile options.

Creating a TDMA Modulation Profile

This section describes how to create a modulation profile for the DOCSIS 1.0/DOCSIS 1.1 TDMA mode of operation, using one of the preconfigured modulation profiles.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**cable modulation-profile** *profile* **tdma** {**mix** | **qam-16** | **qpsk** | **robust-mix**}
	Example:
	`Router(config)# cable modulation-profile 3 tdma mix`
	`Router(config)# cable modulation-profile 4 tdma qpsk`
		Creates a preconfigured modulation profile, where the burst parameters are set to their default values for each burst type:
		Note You can also create custom modulation profiles with the **cable modulation-profile** command by configuring the values for the individual burst parameters. These parameters, however, should not be modified unless you are thoroughly familiar with how changing each parameter affects the DOCSIS MAC layer. We recommend using the preconfigured default modulation profiles for most cable plants.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Creating a Mixed Mode Modulation Profile

This section describes how to create a modulation profile for the mixed TDMA/A-TDMA mode of operation, using one of the preconfigured modulation profiles.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**cable modulation-profile** *profile* **mixed** {**mix-high** | **mix-low** | **mix-mid** | **mix-qam** | **qam-16** | **qpsk** | **robust-mix-high** | **robust-mix-mid** | **robust-mix-qam**}
	Example:
	`Router(config)# cable modulation-profile 143 mixed mix-medium`
	`Router(config)# cable modulation-profile 144 mixed mix-high`
		Creates a preconfigured modulation profile, where the burst parameters are set to their default values for each burst type:
		Note The **robust-mix** profiles are similar to but more robust than the **mix** profiles, so that they are more able to deal with noise on the upstream.
		Note You can also create custom modulation profiles with the **cable modulation-profile** command by configuring the values for the individual burst parameters. These parameters, however, should not be modified unless you are thoroughly familiar with how changing each parameter affects the DOCSIS MAC layer. We recommend using the preconfigured default modulation profiles for most cable plants.

Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Creating an A-TDMA Modulation Profile

This section describes how to create a modulation profile for the DOCSIS 2.0 A-TDMA mode of operation, using one of the preconfigured modulation profiles.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**cable modulation-profile** *profile* **atdma** {**mix-high** | **mix-low** | **mix-mid** | **mix-qam** | **qam-8** | **qam-16** | **qam-32** | **qam-64** | **qpsk** | **robust-mix-high** | **robust-mix-low** | **robust-mix-mid**}
	Example:
	`Router(config)# cable modulation-profile 242 atdma qam-32`
	`Router(config)# cable modulation-profile 243 atdma qam-64`
		Creates a preconfigured modulation profile, where the burst parameters are set to their default values for each burst type:
		Note The **robust-mix** profiles are similar to but more robust than the **mix** profiles, so that they are more able to deal with noise on the upstream.
		Note You can also create custom modulation profiles with the **cable modulation-profile** command by configuring the values for the individual burst parameters. These parameters, however, should not be modified unless you are thoroughly familiar with how changing each parameter affects the DOCSIS MAC layer. We recommend using the preconfigured default modulation profiles for most cable plants.

Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Configuring the DOCSIS Mode and Profile on an Upstream

This section describes how to configure an upstream for a DOCSIS mode of operation, and then to assign a particular modulation profile to that upstream.

Note By default, all upstreams are configured for ATDMA-only mode, using the default modulation profile.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**controller upstream-Cable** *slot/subslot/port* **Example:** `Router(config)# controller upstream-Cable 2/0/1`	Enters controller configuration mode for the interface.
Step 4	**us-channel** *n* **docsis-mode** {**atdma** | **tdma** | **tdma-atdma**} **Example:** `Router(config-controller)# us-channel 0 docsis-mode atdma`	Configures the upstream for the desired DOCSIS mode of operation.
Step 5	**us-channel** *n* **modulation-profile** *primary-profile-number* [*secondary-profile-number*] [*tertiary-profile-number*] **Example:** `Router(config-controller)# us-channel 0 modulation-profile 241`	Assigns up to three modulation profiles to the upstream port. **Note** The type of modulation profiles must match the DOCSIS mode configured for the upstream, using the **us-channel docsis-mode** command.
Step 6	**us-channel** *n* **equalization-coefficient** **Example:** `Router(config-controller)# us-channel 0 equalization-coefficient`	(Optional) Enables the use of a DOCSIS pre-equalization coefficient on the upstream port.
Step 7	**us-channel** *n* **ingress-noise-cancellation** *interval* **Example:** `Router(config-controller)# us-channel 0 ingress-noise-cancellation 400`	(Optional) Configures the interval, in milliseconds, for which the interface card should sample the signal on an upstream to correct any ingress noise that has appeared on that upstream.
Step 8	**us-channel** *n* **maintain-psd** **Example:**	(Optional) Requires DOCSIS 2.0 cable modems that are operating on an ATDMA-only upstream

	Command or Action	Purpose
	`Router(config-controller)# us-channel 0 maintain-psd`	to maintain a constant power spectral density (PSD) after a modulation rate change. **Note** Repeat Step 3, on page 269 through Step 8, on page 269 for each upstream to be configured.
Step 9	**end** **Example:** `Router(config-controller)# end`	Exits controller configuration mode and returns to privileged EXEC mode.

Monitoring the DOCSIS 2.0 A-TDMA Services

This section contains the following:

Displaying Modulation Profiles

To display the modulation profiles that are currently defined on the CMTS, use the **show cable modulation-profile** command without any options:

```
Router# show cable modulation-profile

Mod Docsis  IUC     Type   Pre Diff FEC  FEC  Scrmb  Max Guard Last Scrmb Pre   Pre   RS
    -Mode                  len enco T    k    seed   B   time  CW         offst Type
                                   BYTE BYTE       siz size  short
1   atdma   request 16qam 32  no   0x0  0x10 0x152  0   22    no   yes   0     qpsk1 no
1   atdma   initial 16qam 64  no   0x5  0x22 0x152  0   48    no   yes   0     qpsk1 no
1   atdma   station 16qam 64  no   0x5  0x22 0x152  0   48    no   yes   0     qpsk1 no
1   atdma   a-short 16qam 64  no   0x4  0x4C 0x152  7   22    yes  yes   0     qpsk1 no
1   atdma   a-long  16qam 64  no   0x9  0xE8 0x152  0   22    yes  yes   0     qpsk1 no
1   atdma   a-ugs   16qam 64  no   0x9  0xE8 0x152  0   22    yes  yes   0     qpsk1 no
2   atdma   request 16qam 32  no   0x0  0x10 0x152  0   22    no   yes   0     qpsk1 no
2   atdma   initial 16qam 64  no   0x5  0x22 0x152  0   48    no   yes   0     qpsk1 no
2   atdma   station 16qam 64  no   0x5  0x22 0x152  0   48    no   yes   0     qpsk1 no
2   atdma   a-short 16qam 64  no   0x4  0x4C 0x152  7   22    yes  yes   0     qpsk1 no
2   atdma   a-long  16qam 64  no   0x9  0xE8 0x152  0   22    yes  yes   0     qpsk1 no
2   atdma   a-ugs   16qam 64  no   0x9  0xE8 0x152  0   22    yes  yes   0     qpsk1 no
21  tdma    request qpsk  36  no   0x0  0x10 0x152  0   22    no   yes   0     qpsk  na
21  tdma    initial qpsk  98  no   0x5  0x22 0x152  0   48    no   yes   0     qpsk  na
21  tdma    station qpsk  98  no   0x5  0x22 0x152  0   48    no   yes   0     qpsk  na
21  tdma    short   qpsk  64  no   0x3  0x4C 0x152  12  22    yes  yes   0     qpsk  na
21  tdma    long    qpsk  64  no   0x9  0xE8 0x152  0   22    yes  yes   0     qpsk  na
121 mixed   request qpsk  36  no   0x0  0x10 0x152  0   22    no   yes   0     qpsk  na
121 mixed   initial qpsk  98  no   0x5  0x22 0x152  0   48    no   yes   0     qpsk  na
121 mixed   station qpsk  98  no   0x5  0x22 0x152  0   48    no   yes   0     qpsk  na
121 mixed   short   qpsk  64  no   0x3  0x4C 0x152  12  22    yes  yes   0     qpsk  na
121 mixed   long    qpsk  64  no   0x9  0xE8 0x152  0   22    yes  yes   0     qpsk  na
121 mixed   a-short 64qam 64  no   0x6  0x4C 0x152  6   22    yes  yes   0     qpsk1 no
121 mixed   a-long  64qam 64  no   0x9  0xE8 0x152  0   22    yes  yes   0     qpsk1 no
121 mixed   a-ugs   64qam 64  no   0x9  0xE8 0x152  0   22    yes  yes   0     qpsk1 no
221 atdma   request qpsk  36  no   0x0  0x10 0x152  0   22    no   yes   0     qpsk0 no
221 atdma   initial qpsk  98  no   0x5  0x22 0x152  0   48    no   yes   0     qpsk0 no
221 atdma   station qpsk  98  no   0x5  0x22 0x152  0   48    no   yes   0     qpsk0 no
221 atdma   a-short 64qam 64  no   0x6  0x4C 0x152  6   22    yes  yes   0     qpsk1 no
```

```
221 atdma   a-long  64qam 64  no   0x9  0xE8 0x152  0   22    yes  yes   0     qpsk1 no
```

To display a specific modulation profile in detail, specify the profile number with the **show cable modulation-profile** command:

```
Router# show cable modulation-profile 221

Mod Docsis  IUC     Type  Pre Diff FEC  FEC  Scrmb  Max Guard Last Scrmb Pre   Pre   RS
    -Mode                 len enco T    k    seed   B   time  CW         offst Type
                                  BYTE BYTE        siz size  short
221 atdma   request qpsk  36  no   0x0  0x10 0x152  0   22    no   yes   0     qpsk0 no
221 atdma   initial qpsk  98  no   0x5  0x22 0x152  0   48    no   yes   0     qpsk0 no
221 atdma   station qpsk  98  no   0x5  0x22 0x152  0   48    no   yes   0     qpsk0 no
221 atdma   a-short 64qam 64  no   0x6  0x4C 0x152  6   22    yes  yes   0     qpsk1 no
221 atdma   a-long  64qam 64  no   0x9  0xE8 0x152  0   22    yes  yes   0     qpsk1 no
221 atdma   a-ugs   64qam 64  no   0x9  0xE8 0x152  0   22    yes  yes   0     qpsk1 no
```

Displaying Cable Modem Capabilities and Provisioning

To display the capabilities of the online cable modems and how the modems were provisioned, use the **show cable modem mac** command:

```
Router# show cable modem mac

MAC Address     MAC          Prim  Ver    QoS    Frag Concat PHS Priv DS    US
                State        Sid          Prov                        Saids Sids
1859.334d.7b4c  init(i)      145   DOC1.0 DOC1.0 no   no     no       0     0
1859.334d.fa8c  offline      146   DOC1.0 DOC1.0 no   no     no       0     0
1859.334d.fa02  offline      147   DOC1.0 DOC1.0 no   no     no       0     0
1859.334d.65b0  online(pt)   148   DOC3.0 DOC1.1 yes  yes    yes BPI+ 15    16
1859.334d.6622  offline      149   DOC1.0 DOC1.0 no   no     no       0     0
1859.334d.7a50  init(i)      150   DOC1.0 DOC1.0 no   no     no       0     0
1859.334d.7a2e  offline      151   DOC1.0 DOC1.0 no   no     no       0     0
1859.334d.7d14  online(pt)   152   DOC2.0 DOC1.1 yes  yes    yes BPI+ 15    16
1859.334d.6636  online(pt)   153   DOC2.0 DOC1.1 yes  yes    yes BPI+ 15    16
1859.334d.7cf0  online(pt)   154   DOC2.0 DOC1.1 yes  yes    yes BPI+ 15    16
1859.334d.6742  online(pt)   155   DOC2.0 DOC1.1 yes  yes    yes BPI+ 15    16
1859.334d.7b2a  online(pt)   156   DOC2.0 DOC1.1 yes  yes    yes BPI+ 15    16
1859.334d.7e64  online(pt)   157   DOC2.0 DOC1.1 yes  yes    yes BPI+ 15    16
1859.334d.ede0  online(pt)   158   DOC2.0 DOC1.1 yes  yes    yes BPI+ 15    16
1859.334d.7b8a  online(pt)   159   DOC2.0 DOC1.1 yes  yes    yes BPI+ 15    16
1859.334d.6604  online(pt)   160   DOC2.0 DOC1.1 yes  yes    yes BPI+ 15    16
1859.334d.f93a  online(pt)   161   DOC2.0 DOC1.1 yes  yes    yes BPI+ 15    16
1859.334d.7bf0  online(pt)   162   DOC2.0 DOC1.1 yes  yes    yes BPI+ 15    16
1859.334d.596a  online(pt)   163   DOC2.0 DOC1.1 yes  yes    yes BPI+ 15    16
1859.334d.7d38  online(pt)   164   DOC2.0 DOC1.1 yes  yes    yes BPI+ 15    16
1859.334d.fc64  online(pt)   165   DOC2.0 DOC1.1 yes  yes    yes BPI+ 15    16
1859.334d.6434  online(pt)   166   DOC2.0 DOC1.1 yes  yes    yes BPI+ 15    16
1859.334d.f62a  online(pt)   167   DOC2.0 DOC1.1 yes  yes    yes BPI+ 15    16
!
```

To display how many cable modems of each DOCSIS type are online each upstream, use the **show cable modem mac summary** command:

```
Router# show cable modem mac summary

                           Cable Modem Summary
                           -------------------
                        Mac Version                        QoS Provision Mode
Interface        Total   DOC3.0  DOC2.0  DOC1.1  DOC1.0   Reg/Online   DOC1.1  DOC1.0
Cable3/0/1/U0    20      0       5       0       15       5            5       0
```

```
Cable3/0/1/U1    23      0       9       0       14      9       9       0
Cable3/0/1/U2    21      0       8       0       13      8       8       0
Cable3/0/1/U4    42      0       9       0       33      9       9       0
Cable3/0/1/U5    20      0       15      0       5       15      15      0
Cable3/0/1/U6    18      1       14      0       3       15      15      0
Cable3/0/2/U0    26      0       26      0       0       26      26      0
Cable3/0/2/U1    28      0       28      0       0       28      28      0
Cable3/0/2/U2    24      0       24      0       0       24      24      0
Cable3/0/2/U4    72      0       72      0       0       72      72      0
Cable3/0/3/U0    67      0       63      0       4       63      63      0
Cable3/0/3/U1    85      1       84      0       0       85      85      0
Cable3/0/3/U2    1       0       1       0       0       1       1       0
Cable3/0/4/U0    12      0       1       0       11      1       1       0
Cable3/0/4/U1    39      0       0       0       39      0       0       0
Cable3/0/4/U2    12      0       1       0       11      1       1       0
Cable3/0/4/U4    65      0       11      0       54      11      11      0
Cable3/0/4/U5    10      0       10      0       0       10      10      0
Cable3/0/4/U6    5       0       5       0       0       5       5       0
Cable3/0/5/U0    27      0       27      0       0       27      27      0
Cable3/0/5/U1    27      0       27      0       0       27      27      0
Cable3/0/5/U2    26      0       26      0       0       26      26      0
Cable3/0/5/U4    77      0       77      0       0       77      77      0
Cable3/0/6/U4    14      14      0       0       0       14      14      0
Cable3/0/6/U5    12      12      0       0       0       12      12      0
Cable3/0/6/U6    5       5       0       0       0       5       5       0
```

Configuration Examples for DOCSIS 2.0 A-TDMA services

This section contains the following:

Creating Modulation Profiles Examples

This section contains the following:

Example: DOCSIS 1.0/DOCSIS 1.1 TDMA Modulation Profiles

The following sample configurations show typical modulation profiles for the DOCSIS 1.0/DOCSIS 1.1 TDMA mode of operation:

- Profile 21 is the default profile for TDMA operations.
- Profiles 24 and 25 use the preconfigured 16-QAM and QPSK modulation profiles.
- Profile 26 is a typical QPSK modulation profile using some customized burst parameters.

```
cable modulation-profile 24 tdma qam-16
cable modulation-profile 25 tdma qpsk
cable modulation-profile 26 tdma request 0 16 0 8 qpsk scrambler 152 no-diff 68 fixed
cable modulation-profile 26 tdma initial 5 34 0 48 qpsk scrambler 152 no-diff 128 fixed
cable modulation-profile 26 tdma station 5 34 0 48 qpsk scrambler 152 no-diff 128 fixed
cable modulation-profile 26 tdma short 4 76 12 8 qpsk scrambler 152 no-diff 80 shortened
cable modulation-profile 26 tdma long 8 236 0 8 qpsk scrambler 152 no-diff 80 shortened
```

Example: Mixed TDMA/A-TDMA Modulation Profiles

The following sample configurations show typical modulation profiles for the DOCSIS 1.X/DOCSIS 2.0 mixed TDMA/A-TDMA mode of operation:

- Profile 121 is the default profile for mixed mode operations.
- Profiles 122 through 126 use the preconfigured mixed mode modulation profiles.
- Profile 127 is a typical mixed mode modulation profile some customized burst parameters.

```
cable modulation-profile 121 mixed request 0 16 0 8 qpsk scrambler 152 no-diff 64 fixed
cable modulation-profile 121 mixed initial 5 34 0 48 qpsk scrambler 152 no-diff 32 fixed
cable modulation-profile 121 mixed station 5 34 0 48 qpsk scrambler 152 no-diff 32 fixed
cable modulation-profile 121 mixed short 5 75 6 8 qpsk scrambler 152 no-diff 72 shortened
cable modulation-profile 121 mixed long 8 220 0 8 qpsk scrambler 152 no-diff 80 shortened
cable modulation-profile 121 mixed a-short 0 16 15 99 64qam scrambler 152 no-diff 128
shortened qpsk0 0 18
cable modulation-profile 121 mixed a-long 0 16 15 200 64qam scrambler 152 no-diff 128
shortened qpsk0 0 18

cable modulation-profile 122 mixed mix-high
cable modulation-profile 123 mixed mix-low
cable modulation-profile 124 mixed mix-medium
cable modulation-profile 125 mixed qam-16
cable modulation-profile 126 mixed qpsk

cable modulation-profile 127 mixed request 0 16 0 8 qpsk scrambler 152 no-diff 68 fixed
cable modulation-profile 127 mixed initial 5 34 0 48 qpsk scrambler 152 no-diff 128 fixed
cable modulation-profile 127 mixed station 5 34 0 48 qpsk scrambler 152 no-diff 128 fixed
cable modulation-profile 127 mixed short 6 76 7 8 16qam scrambler 152 no-diff 160 shortened

cable modulation-profile 127 mixed long 8 231 0 8 16qam scrambler 152 no-diff 160 shortened

cable modulation-profile 127 mixed a-short 9 76 6 8 32qam scrambler 152 no-diff 160 shortened
 qpsk1 1 2048
cable modulation-profile 127 mixed a-long 12 231 0 8 64qam scrambler 152 no-diff 132 shortened
 qpsk1 1 2048
```

Example: DOCSIS 2.0 A-TDMA Modulation Profiles

The following sample configurations show typical modulation profiles for the DOCSIS 2.0 A-TDMA mode of operation:

- Profile 221 is the default profile for A-TDMA mode operations.
- Profiles 222 through 226 use the preconfigured A-TDMA mode modulation profiles.
- Profile 227 is a typical A-TDMA mode modulation profile customized burst parameters.

```
cable modulation-profile 221 atdma request 0 16 0 8 qpsk scrambler 152 no-diff 64 fixed
qpsk0 0 18
cable modulation-profile 221 atdma initial 5 34 0 48 qpsk scrambler 152 no-diff 32 fixed
qpsk0 0 18
cable modulation-profile 221 atdma station 5 34 0 48 qpsk scrambler 152 no-diff 32 fixed
qpsk0 0 18
cable modulation-profile 221 atdma short 5 75 6 8 qpsk scrambler 152 no-diff 72 shortened
qpsk0 0 18
cable modulation-profile 221 atdma long 8 220 0 8 qpsk scrambler 152 no-diff 80 shortened
qpsk0 0 18
cable modulation-profile 221 atdma a-short 5 99 10 8 64qam scrambler 152 no-diff 128 shortened
 qpsk0 0 18
cable modulation-profile 221 atdma a-long 15 200 0 8 64qam scrambler 152 no-diff 128 shortened
 qpsk0 0 18

cable modulation-profile 222 atdma qam-8
cable modulation-profile 223 atdma qam-16
cable modulation-profile 224 atdma qam-32
cable modulation-profile 225 atdma qam-64
cable modulation-profile 226 atdma qpsk
```

```
cable modulation-profile 227 atdma request 0 16 0 8 qpsk scrambler 152 no-diff 68 fixed
qpsk0 1 2048
cable modulation-profile 227 atdma initial 0 16 0 0 qpsk no-scrambler no-diff 2 fixed qpsk1
 0 18
cable modulation-profile 227 atdma station 5 34 0 48 qpsk scrambler 152 no-diff 128 fixed
qpsk0 1 2048
cable modulation-profile 227 atdma a-short 9 76 6 8 32qam scrambler 152 no-diff 160 shortened
 qpsk1 1 2048
cable modulation-profile 227 atdma a-long 12 231 0 8 64qam scrambler 152 no-diff 132 shortened
 qpsk1 1 2048
cable modulation-profile 227 atdma a-ugs 3 231 0 8 16qam scrambler 152 no-diff 80 shortened
 qpsk1 1 2048
```

Assigning Modulation Profiles to Upstreams Examples

This section contains the following:

Example: Assigning DOCSIS 1.0/DOCSIS 1.1 TDMA Modulation Profiles

The following sample configuration shows DOCSIS 1.0/DOCSIS 1.1 TDMA modulation profiles being assigned to the upstreams. The TDMA modulation profile (profile 21) is assigned to the upstream controller 2/0/0.

```
controller Upstream-Cable 2/0/0
 us-channel 0 channel-width 1600000 1600000
 us-channel 0 docsis-mode tdma
 us-channel 0 minislot-size 4
 us-channel 0 modulation-profile 21
 no us-channel 0 shutdown
 us-channel 1 channel-width 1600000 1600000
 us-channel 1 docsis-mode tdma
 us-channel 1 minislot-size 4
 us-channel 1 modulation-profile 21
 no us-channel 1 shutdown
 us-channel 2 channel-width 1600000 1600000
 us-channel 2 docsis-mode tdma
 us-channel 2 minislot-size 4
 us-channel 2 modulation-profile 21
 no us-channel 2 shutdown
 us-channel 3 channel-width 1600000 1600000
 us-channel 3 docsis-mode tdma
 us-channel 3 minislot-size 4
 us-channel 3 modulation-profile 21
 no us-channel 3 shutdown
 !
```

Example: Assigning Mixed TDMA/A-TDMA Modulation Profiles

The following sample configuration shows mixed mode TDMA/A-TDMA modulation profiles being assigned to the upstreams. The mixed modulation profile (profile 121) is assigned to the upstream controller 2/0/15.

```
controller Upstream-Cable 2/0/15
 us-channel 0 channel-width 1600000 1600000
 us-channel 0 docsis-mode tdma-atdma
 us-channel 0 minislot-size 4
 us-channel 0 modulation-profile 121
 no us-channel 0 shutdown
 us-channel 1 channel-width 1600000 1600000
 us-channel 1 docsis-mode tdma-atdma
```

```
 us-channel 1 minislot-size 4
 us-channel 1 modulation-profile 121
 no us-channel 1 shutdown
 us-channel 2 channel-width 1600000 1600000
 us-channel 2 docsis-mode tdma-atdma
 us-channel 2 minislot-size 4
 us-channel 2 modulation-profile 121
 no us-channel 2 shutdown
 us-channel 3 channel-width 1600000 1600000
 us-channel 3 docsis-mode tdma-atdma
 us-channel 3 minislot-size 4
 us-channel 3 modulation-profile 121
 no us-channel 3 shutdown
!
```

Example: Assigning DOCSIS 2.0 A-TDMA Modulation Profiles

The following sample configuration shows DOCSIS 2.0 A-TDMA modulation profiles being assigned to the upstreams. The A-TDMA modulation profile (profile 221) is assigned to the upstream controller 2/0/10.

```
controller Upstream-Cable 2/0/10
 us-channel 0 channel-width 1600000 1600000
 us-channel 0 docsis-mode atdma
 us-channel 0 minislot-size 4
 us-channel 0 modulation-profile 221
 no us-channel 0 shutdown
 us-channel 1 channel-width 1600000 1600000
 us-channel 1 docsis-mode atdma
 us-channel 1 minislot-size 4
 us-channel 1 modulation-profile 221
 no us-channel 1 shutdown
 us-channel 2 channel-width 1600000 1600000
 us-channel 2 docsis-mode atdma
 us-channel 2 minislot-size 4
 us-channel 2 modulation-profile 221
 no us-channel 2 shutdown
 us-channel 3 channel-width 1600000 1600000
 us-channel 3 docsis-mode atdma
 us-channel 3 minislot-size 4
 us-channel 3 modulation-profile 221
 no us-channel 3 shutdown
 us-channel 4 channel-width 1600000 1600000
 us-channel 4 docsis-mode atdma
 us-channel 4 minislot-size 4
 us-channel 4 modulation-profile 221
 us-channel 4 shutdown
 us-channel 5 channel-width 1600000 1600000
 us-channel 5 docsis-mode atdma
 us-channel 5 minislot-size 4
 us-channel 5 modulation-profile 221
 us-channel 5 shutdown
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco CMTS Commands	*Cisco IOS CMTS Cable Command Reference* http://www.cisco.com/en/US/docs/ios/cable/command/reference/cbl_book.html

Standards

Standards	Title
SP-RFIv1.1-I09-020830	Data-over-Cable Service Interface Specifications Radio Frequency Interface Specification, version 1.1
SP-RFIv2.0-I03-021218	Data-over-Cable Service Interface Specifications Radio Frequency Interface Specification, version 2.0
SP-OSSIv2.0-I03-021218	Data-over-Cable Service Interface Specifications Operations Support System Interface Specification, version 2.0
SP-BPI+-I09-020830	Data-over-Cable Service Interface Specifications Baseline Privacy Plus Interface Specification, version 2.0
RFC 2233	DOCSIS OSSI Objects Support
RFC 2665	DOCSIS Ethernet MIB Objects Support
RFC 2669	Cable Device MIB

MIBs

MIBs	MIBs Link
• DOCS-BPI-PLUS-MIB • DOCS-CABLE-DEVICE-MIB (RFC 2669) • DOCS-CABLE-DEVICE-TRAP-MIB • DOCS-IF-EXT-MIB • DOCS-IF-MIB (RFC 2670) • DOCS-QOS-MIB • DOCS-SUBMGT-MIB • IGMP-STD-MIB (RFC 2933)	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

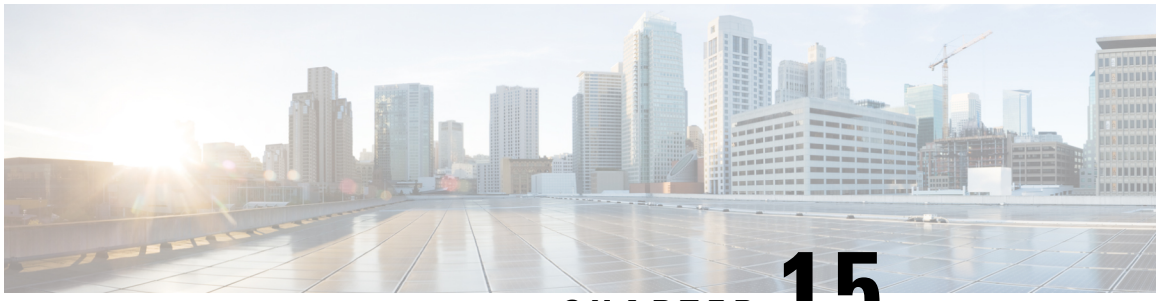
Feature Information for DOCSIS 2.0 A-TDMA Modulation Profile

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 36: Feature Information for DOCSIS 2.0 A-TDMA Modulation Profile

Feature Name	Releases	Feature Information
DOCSIS 2.0 A-TDMA Modulation Profile	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the cisco cBR Series Converged Broadband Routers.



CHAPTER 15

Downstream Resiliency Bonding Group

With more wideband (WB) modems being deployed in cable plants, WB modem resiliency is an important feature. When a comparatively smaller number of cable modems (CMs) observe an impairment on an RF channel, that RF channel stops working. It impacts all the CM using that RF channel, irrespective of whether they reported the impairment on that RF channel. Instead, the solution should be to communicate with the affected cable modems using the good RF channel, without affecting the other cable modems.

The Downstream Resiliency Bonding Group feature allows cable modems with multiple impaired RF channels to be allocated to a dynamically-created wideband interface, which ensures that the performance of the wideband cable modems is not drastically affected.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 280
- Prerequisites for Downstream Resiliency Bonding Group, on page 280
- Restrictions for the Downstream Resiliency Bonding Group, on page 281
- Information About Downstream Resiliency Bonding Group, on page 282
- How to Configure Downstream Resiliency Bonding Group, on page 283
- Verifying Downstream Resiliency Bonding Group Configuration, on page 285
- Troubleshooting the Downstream Resiliency Bonding Group Configuration, on page 289
- Configuration Examples for the Downstream Resiliency Bonding Group, on page 289
- Additional References, on page 293
- Feature Information for Downstream Resiliency Bonding Group, on page 293

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 37: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Downstream Resiliency Bonding Group

- Set aside WB interfaces so that new WB interfaces can be dynamically created from the reserved list of WB interfaces.
- Free up RF bandwidth so that those RF channels can be added to a resiliency bonding group (RBG).
- Remove all existing RBG configuration from the WB interface.

Restrictions for the Downstream Resiliency Bonding Group

- If an existing wideband interface is reserved as a Resiliency Bonding Group (RBG) and later the RBG is removed (through the **no cable ds-resiliency** command), the modems using this RBG go offline and the RBG configuration itself is deleted. Therefore, it is highly recommended that users should not configure an existing BG as an RBG.
- This feature is enabled only when the number of cable modems observing an RF channel impairment is *below* the resiliency threshold. If the number of cable modems on an impaired RF channel is above the resiliency threshold, the impaired RF channel is temporarily removed from the bonding group.
- A cable modem is assigned to an RBG on a first-come-first-served basis. To handle this feature optimally, it is recommended to set aside more WB interfaces and RF channel bandwidth.
- The Cisco CMTS controls the freeing of unused RBGs, when there is no modem using the RGB. The freeing of the unused RGB may take some time and the RGB, which is not completely free cannot be used by the modems. Irrespective of the number of configured RBGs, if all the old RBGs are not completely set free and if the Cisco CMTS tries to move the cable modem to a new RBG, the Cisco CMTS moves the cable modem to the primary DS channel instead of RBG.
- Only SFs on the WB interface associated with the primary SF are moved to an RBG. SFs on other interfaces will not be moved.
- Static SFs are assigned to an RBG on a best effort quality of service (QoS).
- If the **resiliency rf-change-trigger** setting does not have the **secondary** keyword set, only the primary SF is moved to the RBG or a NB interface.
- If the Downstream Resiliency Bonding Group feature is not enabled to use an RBG, only cable modems with impairments on the primary WB interface are moved to the NB interface.
- SFs carrying multicast traffic are not moved.

There may not be enough reserved bonding groups to support all modems facing an impairment at any given time thus the following restrictions must be considered:

- Each RBG has at least two RF channels.
- RBG RF assignments are always a subset of the RF channel assignment of the parent WB interface.
- If an RBG is unavailable for a cable modem, the SF of the CM is moved to a NB interface.
- If a high percentage of cable modems experience an RF impairment and there are no more available bonding group IDs, the impaired RF itself may be removed from the bonding group. Removal of an impaired RF from a parent bonding group is also reflected in the RBG. If an RBG drops to a single RF, all SFs are moved to the NB interface.

The Downstream Resiliency Bonding Group feature has the following cross-functional restrictions:

- All Dynamic service flows, whether they require a committed information rate (CIR) or not, typically voice flows, are created on the NB interface when an RF channel is impaired. Because all SFs assigned to an RBG are best effort only, voice calls may report a quality issue.
- Cable modems participating in the resiliency mode do not take part in load balancing.

- The Downstream Resiliency Bonding Group feature is only supported in the Dynamic Bandwidth Sharing (DBS) mode.

Information About Downstream Resiliency Bonding Group

You can set aside unused bonding groups as RBGs. Ensure that each RF channel is assigned at least 1% of the available bandwidth. Use the **cable rf-channel bandwidth-percent** command to configure the RF channel bandwidth.

Note If the bandwidth-percent is set to 100, the Cisco CMTS does not add any RFs to the RBG. In other words, this feature will not be enabled.

The Cisco CMTS controls the assignment and freeing of unused RBGs. If an RF channel is removed from a WB interface, it is also removed from any associated RBGs.

Note If the wideband interface is in standby mode, the Cisco CMTS does not assign or free up the unused downstream bonding group.

A suspended RF channel is restored for all affected wideband interfaces when a specified number of cable modems report (via CM-STATUS) that the channel connectivity is restored. The Wideband Modem Resiliency feature defines the specified number of cable modems as half of the configured count or percentage of rf-change-trigger, or both. For example, if the count is 20 and the percent is 10, then the number of cable modems reporting recovery should reduce the count to 10 and the percent to 5 for the suspended RF channel to be restored.

Finding a Best-Fit RBG for the Cable Modem

A bonding group is a list of channels that provide a means to identify the channels that are bonded together. The Cisco CMTS assigns a service flow (SF) to an RBG based on the attributes of the SF and the attributes of the individual channels of the bonding group.

In the Downstream Resiliency Bonding Group feature, when a line card receives a CM-STATUS message from the cable modem informing the line card that there is an RF channel impairment, the line card checks for the number of good RF channels and:

- Moves the cable modem to narrowband mode if there is only one available RF channel.
- Moves the cable modem to wideband mode if the cable modem reports all RF channels are in good state.
- Moves the cable modem to an RBG if there are two or more good RF channels, with at least one RF channel impaired, and if the Downstream Resiliency Bonding Group feature is enabled.

When the Cisco CMTS receives a message from the line card to move a cable modem to an RBG, the Cisco CMTS attempts to find an existing RBG or creates an RBG that satisfies the impairment.

Note If two or more RBGs are reserved for the same wideband controller, the Cisco CMTS creates one RBG for each cable modem.

Note The Cisco CMTS creates more than one RBG from a parent WB interface if the user has set aside more than one WB interface as the RBG and the RF bandwidth does not exceed 100%.

If a matching RBG is not found or cannot be created, the Cisco CMTS looks for an RBG with a subset of the required RF channels and if available, the cable modem is assigned to such an RBG.

However, if no such RBG exists, the Cisco CMTS instructs the line card to move the cable modem to NB mode.

How to Configure Downstream Resiliency Bonding Group

This section contains the following:

Enabling Downstream Resiliency Bonding Group

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable rf-change-trigger** {**percent** *value* | **count** *number*} [**secondary**] **Example:** `Router(config)# cable rf-change-trigger percent 50 count 1 secondary`	Specifies the amount of time an event must persist before it triggers an action for the reporting CM.
Step 4	**cable resiliency ds-bonding** **Example:** `Router(config)# cable resiliency ds-bonding`	Enables the downstream resiliency bonding group.
Step 5	**exit** **Example:**	Returns to the global configuration mode.

	Command or Action	Purpose
	`Router(config)# exit`	

What to do next

Note The result of using the **cable rf-change-trigger** command with the **cable resiliency ds-bonding** command is different from using only the **cable rf-change-trigger** command. For more information, see Downstream Resiliency Narrowband Mode Versus Resiliency Bonding Group, on page 285.

Reserving a Resiliency Bonding Group for a Line Card

This section describes reserving a bonding group or a wideband interface for a line card per controller.

Restriction When you reserve a resiliency bonding group using the cable ds-resiliency command, the existing bundle and RF channel configurations on the wideband interface will be removed automatically. Other configurations like admission control, should be removed manually.

After downstream resiliency bonding group is configured, avoid other manual configurations.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface wideband-cable** *slot/subslot/port:wideband-channel* **Example:** `Router(config)# interface wideband-cable 1/0/0:7`	Configures a wideband cable interface.
Step 4	**cable ds-resiliency** **Example:** `Router(config-if)# cable ds-resiliency`	Reserves an individual bonding group or WB interface for usage on a line card, on a per controller basis.
Step 5	**exit** **Example:**	Returns to the global configuration mode.

	Command or Action	Purpose
	`Router(config-if)# exit`	

Verifying Downstream Resiliency Bonding Group Configuration

This section contains the following:

Verifying the Downstream Resiliency Bonding Group

To verify if the Downstream Resiliency Bonding Group feature is enabled, use the **show cable modem resiliency** command as shown in the following example:

```
Router# show cable modem resiliency
                              Orig BG                Curr BG
I/F     MAC Address    ID   I/F           RFs ID   I/F           RFs
------- -------------- ---------------------- ----------------------
C7/0/0  0025.2eaf.843e 897  Wi7/0/0:0     4   898  Wi7/0/0:1     3
C7/0/0  0025.2eaf.8356 897  Wi7/0/0:0     4   899  Wi7/0/0:2     3
C7/0/0  0015.d176.5199 897  Wi7/0/0:0     4   720  In7/0/0:0
```

The **Current BG I/F** field indicates whether Downstream Resiliency Bonding Group feature is enabled and if the cable modems are assigned to a WB interface.

Verifying a Reserved Resiliency Bonding Group

To verify if a BG is reserved for a line card, use the **show cable resiliency** command as shown in the following example:

```
Router# show cable resiliency
             BG   Resil BG                                  RF
Resil BG I/F  ID   State          Count Time             Ctrl  Num
------------ ---- -------------- ----- --------------- ----------
Wi1/0/0:10    10   Free
Wi1/0/0:20    20   Free
Wi7/0/0:1     1    Assigned       3     Nov 3  09:55:49 0     0
                                                              1
                                                              2
Wi7/0/0:2     2    Assigned       3     Nov 3  09:57:09 0     0
                                                              1
                                                              3
```

Downstream Resiliency Narrowband Mode Versus Resiliency Bonding Group

This section provides the sample outputs when using the **cable rf-change-trigger** command with the **cable resiliency ds-bonding** command and using only the **cable rf-change-trigger** command.

Table 38: Downstream Resiliency Narrowband Mode Versus Resiliency Bonding Group - Scenario 1

Effect on	Using only cable rf-change-trigger command (Downstream Resiliency NB Mode)		Using cable rf-change-trigger command with cable resiliency ds-bonding (Downstream Resiliency Bonding Group)	
	Below Threshold	**Above Threshold**	**Below Threshold**	**Above Threshold**
Primary Service Flow	Moves to the primary channel.	Remains on the original bonding group while the impaired downstream channels are not used and are reported as DOWN.	Moves to dynamic bonding group.	Remains on the original bonding group while the impaired downstream channels are not used and are reported as DOWN.
Secondary Service Flows	Remain on the original WB interface.	Remains on the original bonding group while the impaired downstream channels are not used and are reported as DOWN.	Remains on the original bonding group.	Remains on the original bonding group while the impaired downstream channels are not used and are reported as DOWN.

The following is a sample output for a cable modem when the **cable rf-change-trigger** command is used with the **cable resiliency ds-bonding** command and the number of cable modems observing an RF channel impairment is *below* the resiliency threshold:

```
Router# show cable modem

                                                                          D
MAC Address    IP Address     I/F          MAC          Prim RxPwr  Timing Num I
                                           State        Sid  (dBmv) Offset CPE P
0023.be83.1c9e 10.1.11.46     C5/0/0/UB    w-online     922  -0.50  1055   0   N
0023.be83.1caa 10.1.11.28     C5/0/0/UB    w-online     923  0.00   1043   0   N
0025.2ecf.f19c 10.1.11.53     C5/0/0/UB    w-online     925  0.00   1057   0   N
0022.3a30.9fc0 10.1.11.47     C5/0/0/UB    w-online     926  0.00   1055   0   N
001a.c3ff.e3d4 10.1.11.39     C5/0/0/UB    p-online     927  0.00   1307   0   N
0023.be83.1c9a 10.1.11.61     C5/0/0/UB    w-online     928  0.00   1057   0   N
0022.3a30.9fbc 10.1.11.60     C5/0/0/UB    p-online     929  -0.50  1055   0   N
0023.be83.1c8c 10.1.11.38     C5/0/0/UB    w-online     930  0.00   1061   0   N
001e.6bfb.1964 10.1.11.63     C5/0/0/UB    p-online     931  0.50   1305   0   N
0025.2ecf.f196 10.1.11.29     C5/0/0/UB    w-online     932  0.00   1057   0   N
0025.2ecf.f04e 10.1.11.54     C5/0/0/UB    w-online     933  0.00   1054   0   N
0022.3a30.9fc8 10.1.11.43     C5/0/0/UB    w-online     934  0.00   1056   0   N
0025.2ecf.f190 10.1.11.55     C5/0/0/UB    w-online     935  0.00   1059   0   N
0022.3a30.9fd0 10.1.11.52     C5/0/0/UB    p-online     936  0.00   1057   0   N
0022.ce97.8268 10.1.11.31     C5/0/0/UB    w-online     937  -0.50  1056   0   N
0022.ce97.8281 10.1.11.25     C5/0/0/UB    w-online     938  0.00   1058   0   N
001a.c3ff.e4ce 10.1.11.44     C5/0/0/UB    w-online     940  -0.50  1304   0   N
0022.ce9c.839e 10.1.11.32     C5/0/0/UB    w-online     941  -0.50  1305   0   N
0022.cea3.e768 10.1.11.41     C5/0/0/UB    w-online     942  -1.00  1305   0   N
0022.ce9c.8398 10.1.11.33     C5/0/0/UB    w-online     943  0.00   1306   0   N
001a.c3ff.e50a 10.1.11.59     C5/0/0/UB    w-online     944  0.00   1304   0   N
001a.c3ff.e3f8 10.1.11.57     C5/0/0/UB    w-online     945  -1.00  1306   0   N
001e.6bfb.1a14 10.1.11.37     C5/0/0/UB    w-online     946  0.00   1305   0   N
```

Note p-online indicates that cable modem has reported NP RF failure and it is in downstream partial service mode.

```
Router# show cable resiliency

              BG   Resil BG                                  RF
Resil BG I/F  ID   State           Count Time            Ctrl  Num
------------- ---- -------------- ----- --------------- ----------
Wi5/0/0:2     2    Assigned       1     Mar 30 14:46:43 0     0
                                                               1
                                                               2
Wi5/0/0:3     3    Assigned       1     Mar 30 14:46:43 0     0
                                                               1
                                                               2
                                                         1     0
                                                               1
                                                               2
                                                               3
Wi5/0/0:4     4    Free           0
Wi5/0/0:5     5    Free           0

Router# show cable modem resiliency
                               Orig BG                Curr BG
I/F     MAC Address    ID   I/F            RFs ID   I/F            RFs
------- -------------- -------------------- ----------------------
C5/0/0  001a.c3ff.e3d4 258  Wi5/0/0:1     4   259  Wi5/0/0:2     3 <- Dynamic Bonding Group
C5/0/0  0022.3a30.9fbc 257  Wi5/0/0:0     8   260  Wi5/0/0:3     7 <- Dynamic Bonding Group
C5/0/0  001e.6bfb.1964 258  Wi5/0/0:1     4   259  Wi5/0/0:2     3 <- Dynamic Bonding Group
C5/0/0  0022.3a30.9fd0 257  Wi5/0/0:0     8   260  Wi5/0/0:3     7 <- Dynamic Bonding Group
```

The following is a sample output for a cable modem under the following conditions:

- **cable rf-change-trigger** command is used with the **cable resiliency ds-bonding** command
- Number of cable modems observing an RF channel impairment is *below* the resiliency threshold
- There is no available WB interface for the resiliency bonding group:

```
Router# show cable modem
 0025.2ecf.f196 service-flow version

SUMMARY:
MAC Address    IP Address      Host          MAC           Prim  Num Primary    DS
                               Interface     State         Sid   CPE Downstream RfId
0025.2ecf.f196 10.1.11.29      C5/0/0/UB     p-online
      932   0   In5/0/0:0  240
Sfid  Dir Curr  Sid   Sched  Prio MaxSusRate  MaxBrst     MinRsvRate  Throughput
          State       Type
1867  US  act   932   BE     0    0           10000       0           294
1868  DS  act   N/A   N/A    0    0           3044        0           154

Router# show cable resiliency
              BG   Resil BG                                  RF
Resil BG I/F  ID   State           Count Time            Ctrl  Num
------------- ---- -------------- ----- --------------- ----------
Wi5/0/0:2     2    Assigned       6     Mar 30 15:57:09 0     0
                                                               1
                                                               2
                                                               3
                                                         1     0
                                                               2
                                                               3
Wi5/0/0:3     3    Assigned       8     Mar 30 15:53:58 0     0
```

```
                                                                  1
                                                                  2
                                                            1     1
                                                                  2
                                                                  3
Wi5/0/0:4     4    Assigned       2     Mar 30 15:53:58 0        0
                                                                  1
                                                                  2
                                                                  3
                                                            1     1
                                                                  2
                                                                  3
Wi5/0/0:5     5    Assigned       2     Mar 30 15:58:35 0        0
                                                                  1
                                                                  2
                                                                  3
                                                            1     0
                                                                  1
                                                                  3

Router# show cable modem resiliency

                              Orig BG                Curr BG
I/F     MAC Address    ID   I/F           RFs ID   I/F           RFs
------- -------------- ---------------------- ----------------------
C5/0/0  0025.2ecf.f19c 257  Wi5/0/0:0     8   259  Wi5/0/0:2     7
C5/0/0  0025.2ecf.f196 257  Wi5/0/0:0     8   240  In5/0/0:0     <-- move NB for no available
 WB interface
C5/0/0  0025.2ecf.f04e 257  Wi5/0/0:0     8   262  Wi5/0/0:5     7
C5/0/0  0022.3a30.9fbc 257  Wi5/0/0:0     8   260  Wi5/0/0:3     6
C5/0/0  0022.3a30.9fd0 257  Wi5/0/0:0     8   261  Wi5/0/0:4     7
```

Table 39: Downstream Resiliency Narrowband Mode Versus Resiliency Bonding Group - Scenario 2

Effect on	Using only cable rf-change-trigger secondary command (Downstream Resiliency NB Mode)		Using cable rf-change-trigger secondary command with cable resiliency ds-bonding (Downstream Resiliency Bonding Group)	
	Below Threshold	**Above Threshold**	**Below Threshold**	**Above Threshold**
Primary Service Flow	Moves all service flows to the primary channel.	Remains on the original bonding group while the impaired downstream channels are not used and are reported as DOWN.	Moves all service flows to a dynamic bonding group.	Remains on the original bonding group while the impaired downstream channels are not used and are reported as DOWN.
Secondary Service Flows				

The following is a sample output for a cable modem when the **cable rf-change-trigger secondary** command is used with the **cable resiliency ds-bonding** command and the number of cable modems observing an RF channel impairment is *below* the resiliency threshold:

```
Router# show cable modem 0025.2ecf.f196 service-flow

SUMMARY:
MAC Address    IP Address      Host          MAC           Prim  Num Primary    DS
                               Interface     State         Sid   CPE Downstream RfId
0025.2ecf.f196 10.1.11.29      C5/0/0/UB     p-online      955   0   In5/0/0:0  240
Sfid  Dir Curr  Sid   Sched  Prio MaxSusRate  MaxBrst     MinRsvRate  Throughput
          State       Type
```

```
1913  US  act   955   BE     0    10000000    10000       0           425
1915  US  act   956   RTPS   7    0           3044        100000      0
1916  US  act   957   BE     0    0           3044        50000       0
1917  US  act   958   BE     4    0           3044        0           0
1914  DS  act   N/A   N/A    0    100000000   20000       0           0       <-- Primary
Service-Flow
1918  DS  act   N/A   N/A    0    0           3044        0           0       <-- Secondary
 Service-Flow
1919  DS  act   N/A   N/A    0    0           3044        0           0       <-- Secondary
 Service-Flow
1920  DS  act   N/A   N/A    4    4500000     3044        0           0       <-- Secondary
 Service-Flow
UPSTREAM SERVICE FLOW DETAIL:
SFID  SID   Requests   Polls      Grants     Delayed    Dropped    Packets
                                             Grants     Grants
1913  955   83         0          83         0          0          92
1915  956   0          0          0          0          0          0
1916  957   0          0          0          0          0          0
1917  958   0          0          0          0          0          0
DOWNSTREAM SERVICE FLOW DETAIL:
SFID  RP_SFID QID    Flg Policer               Scheduler             FrwdIF
                         Xmits      Drops      Xmits      Drops
1914  33210   131555     90         0          6          0          Wi5/0/0:3 <-- Dynamic
 Bonding Group
1918  33211   131556     0          0          0          0          Wi5/0/0:3
1919  33212   131557     0          0          0          0          Wi5/0/0:3
1920  33213   131558     0          0          0          0          Wi5/0/0:3
```

Troubleshooting the Downstream Resiliency Bonding Group Configuration

Use the following commands to get information on the WB interface, number of CMs in an impaired state, resiliency bonding groups, their associated bonding groups, available RF channels, and the number of CMS and service flows assigned to them:

- **debug cable wbcmts resiliency**
- **debug cable wbcmts resiliency report**
- **show cable resiliency**
- **show cable modem resiliency**
- **show cable modem wideband rcs-status**
- **show cable modem service-flow verbose**
- **show cable resil-rf-status**
- **show cable modem summary wb-rfs**

Configuration Examples for the Downstream Resiliency Bonding Group

The following is an example of the configuration of the Downstream Resiliency Bonding Group feature:

```
cable rf-change-trigger count 10 secondary
cable resiliency ds-bonding
!
```

```
controller Upstream-Cable 9/0/1
us-channel 0 frequency 13200000
us-channel 0 channel-width 6400000 6400000
us-channel 0 power-level -1
us-channel 0 docsis-mode atdma
us-channel 0 minislot-size 8
us-channel 0 modulation-profile 221
no us-channel 0 shutdown
us-channel 1 frequency 19600000
us-channel 1 channel-width 6400000 6400000
us-channel 1 power-level -1
us-channel 1 docsis-mode atdma
us-channel 1 minislot-size 8
us-channel 1 modulation-profile 221
no us-channel 1 shutdown
us-channel 2 frequency 26000000
us-channel 2 channel-width 6400000 6400000
us-channel 2 power-level -1
us-channel 2 docsis-mode atdma
us-channel 2 minislot-size 8
us-channel 2 modulation-profile 221
no us-channel 2 shutdown
us-channel 3 frequency 32400000
us-channel 3 channel-width 6400000 6400000
us-channel 3 power-level -1
us-channel 3 docsis-mode atdma
us-channel 3 minislot-size 8
us-channel 3 modulation-profile 221
no us-channel 3 shutdown
!
controller Integrated-Cable 9/0/1
max-carrier 128
base-channel-power 34
rf-chan 0
  type DOCSIS
  frequency 381000000
  rf-output NORMAL
  power-adjust -2
  docsis-channel-id 1
  qam-profile 1
rf-chan 1 3
  type DOCSIS
  frequency 387000000
  rf-output NORMAL
  power-adjust 0
  docsis-channel-id 2
  qam-profile 1
rf-chan 32 35
  type DOCSIS
  frequency 477000000
  rf-output NORMAL
  power-adjust 0
  docsis-channel-id 33
  qam-profile 1
rf-chan 64 67
  type DOCSIS
  frequency 501000000
  rf-output NORMAL
  power-adjust 0
  docsis-channel-id 65
  qam-profile 1
rf-chan 96 99
  type DOCSIS
  frequency 669000000
```

```
  rf-output NORMAL
  power-adjust 0
  docsis-channel-id 97
  qam-profile 1
!
interface Cable9/0/1
downstream Integrated-Cable 9/0/1 rf-channel 0-3
downstream Integrated-Cable 9/0/1 rf-channel 32-35
upstream 0 Upstream-Cable 9/0/1 us-channel 0
upstream 1 Upstream-Cable 9/0/1 us-channel 1
upstream 2 Upstream-Cable 9/0/1 us-channel 2
upstream 3 Upstream-Cable 9/0/1 us-channel 3
cable upstream bonding-group 1
  upstream 0
  upstream 1
  upstream 2
  attributes 80000000
cable upstream bonding-group 2
  upstream 0
  upstream 1
  attributes 80000000
cable upstream bonding-group 3
  upstream 1
  upstream 2
  attributes 80000000
cable upstream bonding-group 4
  upstream 0
  upstream 2
  attributes 80000000
cable upstream bonding-group 5
  attributes 80000000
cable bundle 1
no cable mtc-mode
cable privacy accept-self-signed-certificate
end
!
interface Integrated-Cable9/0/1:0
cable bundle 1
cable rf-bandwidth-percent 65
!
interface Wideband-Cable9/0/1:0
cable bundle 1
cable privacy accept-self-signed-certificate
cable rf-channels channel-list 0-3 bandwidth-percent 20
!
interface Integrated-Cable9/0/1:1
cable bundle 1
cable rf-bandwidth-percent 65
!
interface Wideband-Cable9/0/1:1
cable bundle 1
cable privacy accept-self-signed-certificate
cable rf-channels channel-list 32-35 bandwidth-percent 20
!
!
interface Wideband-Cable9/0/1:60
cable ds-resiliency
!
interface Wideband-Cable9/0/1:61
cable ds-resiliency
!
interface Wideband-Cable9/0/1:62
cable ds-resiliency
```

```
!
```

The following is a sample output for the **show cable modem** command to display impaired cable modems below the resiliency threshold value:

```
Router# show cable modem

MAC Address    IP Address      I/F           MAC           Prim  RxPwr  Timing Num I
                                             State         Sid   (dBmv) Offset CPE P
e448.c70c.96d5 80.17.150.6     C9/0/1/U2     p-online      1     0.00   1784   0   N
e448.c70c.96f3 80.17.150.14    C9/0/1/U1     w-online      2     -1.00  1797   0   N
68ee.9633.0699 80.17.150.31    C9/0/1/U0     w-online      3     -1.00  2088   1   N
e448.c70c.96e7 80.17.150.29    C9/0/1/U3     p-online      4     -0.50  1785   0   N
e448.c70c.982b 80.17.150.18    C9/0/1/U2     w-online      5     0.00   1780   0   N
e448.c70c.9804 80.17.150.13    C9/0/1/U3     w-online      6     -0.50  1788   0   N
e448.c70c.9819 80.17.150.30    C9/0/1/U0     w-online      7     -1.00  1782   0   N
e448.c70c.980d 80.17.150.17    C9/0/1/U0     w-online      8     -1.00  1787   0   N
```

Note p-online indicates that the cable modem has reported NP RF failure and it is in downstream partial service mode.

The following is a sample output when RBGs are created:

```
Router# show cable resiliency

              BG    Resil BG                                  RF
Resil BG I/F  ID    State          Count Time             Ctrl  Num
------------- ----- -------------- ----- ---------------- ----------
Wi9/0/1:60    28989 Assigned       1     Jan 9  07:35:08 1     0
                                                                1
                                                                2
Wi9/0/1:61    28990 Assigned       1     Jan 9  07:36:54 1     0
                                                                1
                                                                3
Wi9/0/1:62    28991 Free           0
```

The following is a sample output when cable modems service flows are assigned to RBGs:

```
Router# show cable modem resiliency

                              Orig BG                Curr BG
I/F     MAC Address    ID    I/F           RFs ID    I/F           RFs
------- -------------- ----------------------- ----------------------
C9/0/1  e448.c70c.96d5 28929 Wi9/0/1:0     4   28989 Wi9/0/1:60    3
C9/0/1  e448.c70c.96e7 28929 Wi9/0/1:0     4   28990 Wi9/0/1:61    3
```

The following is a sample output of the **show cable modem** command when the impaired cable modems have recovered:

```
Router# show cable modem

MAC Address    IP Address      I/F           MAC           Prim  RxPwr  Timing Num I
                                             State         Sid   (dBmv) Offset CPE P
e448.c70c.96d5 80.17.150.6     C9/0/1/U2     w-online      1     0.00   1784   0   N
e448.c70c.96f3 80.17.150.14    C9/0/1/U1     w-online      2     -1.00  1797   0   N
68ee.9633.0699 80.17.150.31    C9/0/1/U0     w-online      3     -1.00  2088   1   N
e448.c70c.96e7 80.17.150.29    C9/0/1/U3     w-online      4     -0.50  1785   0   N
e448.c70c.982b 80.17.150.18    C9/0/1/U2     w-online      5     0.00   1780   0   N
```

```
e448.c70c.9804 80.17.150.13    C9/0/1/U3     w-online       6     -0.50  1788   0    N
e448.c70c.9819 80.17.150.30    C9/0/1/U0     w-online       7     -1.00  1782   0    N
e448.c70c.980d 80.17.150.17    C9/0/1/U0     w-online       8     -1.00  1787   0    N
```

The following is a sample output of the **show cable resiliency** command when the impaired cable modems have recovered:

```
Router# show cable resiliency

               BG    Resil BG                                  RF
Resil BG I/F   ID    State          Count Time              Ctrl  Num
-------------- ----- -------------- ----- ----------------- ----------
Wi9/0/1:60     28989 Free           1     Jan 9  07:35:08
Wi9/0/1:61     28990 Free           1     Jan 9  07:36:54
Wi9/0/1:62     28991 Free           0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco CMTS Command Reference	http://www.cisco.com/c/en/us/td/docs/cable/cmts/cmd_ref/b_cmts_cable_cmd_re

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

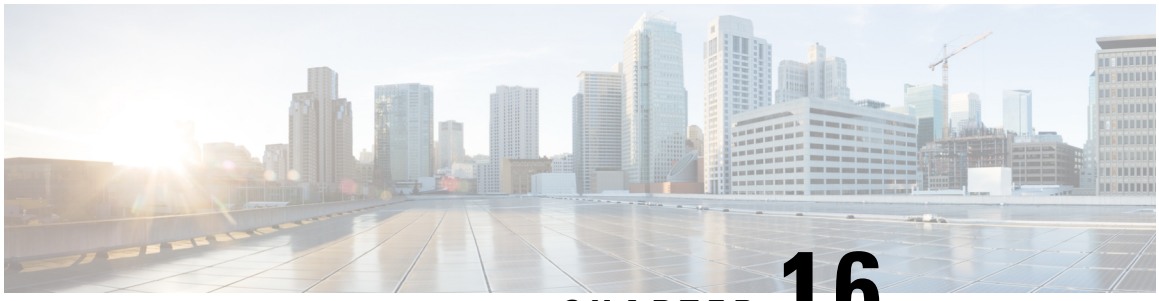
Feature Information for Downstream Resiliency Bonding Group

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 40: Feature Information for Downstream Resiliency Bonding Group

Feature Name	Releases	Feature Information
Downstream Resiliency Bonding Group	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 16

Downstream Channel ID Assignment

First Published: April 17, 2015

The DOCSIS downstream channel ID (DCID) is defined as an 8-bit identifier for recognizing a Downstream Channel within a MAC Domain. All CMTS downstream channels are assigned a DCID by default that may be subsequently changed by configuration. It is used in most DOCSIS downstream packet headers and its valid range is from 1 to 255 (0 is reserved for network management purposes).

Note All downstream channels in a MAC domain must have a unique DCID within the MAC domain.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 295
- Information About Downstream Channel ID Assignment on the Cisco CMTS Routers, on page 296
- How to Configure Downstream Channel ID Assignment on the Cisco CMTS Routers, on page 299
- Additional References, on page 302
- Feature Information for Downstream Channel ID Assignment, on page 302

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 41: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About Downstream Channel ID Assignment on the Cisco CMTS Routers

These are the downstream channel ID assignment features:

- Unique DCIDs are provided for all channels within a single controller by default.

Note DCID values for downstream channels in the same MAC Domain must be unique. If a MAC Domain only contains channels from a single controller, the default DCID values will be sufficient. If a MAC Domain contains channels from multiple controllers, DCID conflicts may be encountered within the MAC Domain. DCID conflicts may be resolved by changing the DCID value of the conflicting channels within the controller configuration or by enabling the automatic channel ID assignment feature.

- The default DCID value for each downstream channel within a controller is equivalent to rf-chan number plus one. For example, the default value for rf-chan 0 is 1, for rf-chan 1 is 2.

Manual Downstream Channel ID Assignment

When using the manual DCID provisioning feature, every downstream channel in the system is assigned a default DCID value equivalent to (controller QAM id + 1). The table below shows the default DCID ranges per downstream controller.

Table 42: Default Downstream Channel IDs Per Slot/Subslot/Controller

	0/0	1/0	2/0	3/0	6/0	7/0	8/0	9/0
DS Controller 0	1-128	1-128	1-128	1-128	1-128	1-128	1-128	1-128
DS Controller 1	1-128	1-128	1-128	1-128	1-128	1-128	1-128	1-128
DS Controller 2	1-128	1-128	1-128	1-128	1-128	1-128	1-128	1-128
DS Controller 3	1-128	1-128	1-128	1-128	1-128	1-128	1-128	1-128
DS Controller 4	1-128	1-128	1-128	1-128	1-128	1-128	1-128	1-128
DS Controller 5	1-128	1-128	1-128	1-128	1-128	1-128	1-128	1-128
DS Controller 6	1-128	1-128	1-128	1-128	1-128	1-128	1-128	1-128
DS Controller 7	1-128	1-128	1-128	1-128	1-128	1-128	1-128	1-128

The default DCID value can be replaced with a user configurable value. The configuration is available in the downstream controller per channel. The current DCID values for the channels within a downstream controller can be viewed in the dcid column of the **show controller Integrated-Cable rf-chan** command output. The example shows channels with default DCID values. When a DCID value is changed in the configuration, the new value appears in the output below.

```
Router#show controllers integrated-Cable 3/0/0 rf-channel 1-127
Chan State Admin Frequency  Type   Annex Mod  srate Interleaver  dcid  power  output
 1   NPRE   UP   99000000   DOCSIS  B    256   5361  I32-J4       2    37     NORMAL
 2   NPRE   UP   105000000  DOCSIS  B    256   5361  I32-J4       3    37     NORMAL
 3   NPRE   UP   111000000  DOCSIS  B    256   5361  I32-J4       4    37     NORMAL
 4   NPRE   UP   117000000  DOCSIS  B    256   5361  I32-J4       5    37     NORMAL
```

```
 5   NPRE   UP   123000000  DOCSIS  B    256   5361  I32-J4       6    37     NORMAL
 6   NPRE   UP   129000000  DOCSIS  B    256   5361  I32-J4       7    37     NORMAL
 7   NPRE   UP   135000000  DOCSIS  B    256   5361  I32-J4       8    37     NORMAL
 8   NPRE   UP   141000000  DOCSIS  B    256   5361  I32-J4       9    37     NORMAL
 9   NPRE   UP   147000000  DOCSIS  B    256   5361  I32-J4       10   37     NORMAL
 10  NPRE   UP   153000000  DOCSIS  B    256   5361  I32-J4       11   37     NORMAL
 11  NPRE   UP   159000000  DOCSIS  B    256   5361  I32-J4       12   37     NORMAL
 12  NPRE   UP   165000000  DOCSIS  B    256   5361  I32-J4       13   37     NORMAL
 13  NPRE   UP   171000000  DOCSIS  B    256   5361  I32-J4       14   37     NORMAL
 14  NPRE   UP   177000000  DOCSIS  B    256   5361  I32-J4       15   37     NORMAL
 15  NPRE   UP   183000000  DOCSIS  B    256   5361  I32-J4       16   37     NORMAL

Router#
```

Automatic Downstream Channel ID Assignment on the Cisco CMTS Routers

It is possible to automatically assign a unique set of downstream channel IDs to meet all DOCSIS requirements by enabling the Automatic DCID Assignment feature. When enabled, Downstream channel DCIDs will be automatically assigned when the channels are added to a fiber node and associated with a MAC Domain. Therefore, the use of fiber node configuration is a prerequisite for this feature.

Service Impact

Changing the DOCSIS downstream channel ID causes cable modems to re-register. Cable modems receive MAC Domain Descriptor (MDD) and Upstream Channel Descriptor (UCD) messages with a changed DCID in their headers.

- Enabling the automatic DCID assignment displays the following message:

```
WARNING: Enabling automatic DCID assignment will cause modems to flap and will apply
to all fiber nodes on this CMTS.
```

- Disabling the automatic DCID assignment displays the following message:

```
WARNING: Disabling automatic DCID assignment will no longer enforce channel-id uniqueness
 at fiber nodes. Channel ID changes may require manual verification to prevent conflicts.
```

- If there is a DCID conflict with another channel in the MAC Domain, the following error message is displayed:

```
ERROR: <slot>/<subslot>/<controller> rf-channel <channel>: The downstream channel id
conflicts with interface In<slot>/<subslot>/<controller>:channel. Downstream channel
id must be unique in a CGD.
```

- After automatic DCID assignment is configured, if there is a DCID conflict when a downstream channel that belongs to a fiber node is added to a MAC Domain, the automatic DCID feature tries to resolve the conflict by assigning another automatic DCID and the following message is displayed:

```
WARNING: The downstream channel id conflict for <slot>/<subslot>/<controller> rf-channel
 <channel> was resolved by Automatic DCID Assignment.
Please run "interface <md-slot>/<md-subslot>/<md-index>" followed by
"<slot>/<subslot>/<controller> rf-channel <channel>" again in order to add the channel.
```

  To add the channel, use this channel grouping domain (CGD) command again:

  cable downstream *x/y/z* **rf-channel** *channel*

- If automatic DCID is configured and the channel does not belong to a fiber node, or if automatic DCID cannot resolve the conflict, the following message is displayed:

```
WARNING: The downstream channel id conflict for <slot>/<subslot>/<controller> rf-channel
 <channel> could not be resolved by Automatic DCID Assignment.
```

To resolve this issue, add the channel to a fiber node.

How to Configure Downstream Channel ID Assignment on the Cisco CMTS Routers

The following sections describe how to configure downstream channel ID assignment.

Configuring Manual Downstream Channel ID Assignment

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface controller integrated-Cable** *slot*/*subslot*/*port* **Example:** `Router(config)# interface controller integrated-Cable 1/0/1`	Enters controller configuration mode for the Channel Grouping Domain host line card.
Step 4	**rf-chan** *downstream QAM ID* • Alternatively, use the **rf-chan** *starting downstream QAM ID ending downstream QAM ID* command to set the range of downstream channel IDs. **Example:** `Router(config-controller)# rf-chan 0`	Enters the rf-channel configuration mode.
Step 5	**docsis-channel-id** *DCID* **Example:**	Configures the downstream channel's DCID to the specified value, for the RF channel.

	Command or Action	Purpose
	`Router(config-rf-chan)#docsis-channel-id 1`	For the rf-channel range that was configured using the **rf-chan** *starting downstream QAM ID ending downstream QAM ID* command, the **docsis-channel-id** *DCID* command configures the DCIDs for the rf-channels in that range.

Configuring Automatic Downstream Channel ID Assignment

Automatic DCID assignment should be permanently configured. However, if you need to remove the feature, use the **no** or **default** commands.

Note The **no** or **default** form of the command is not written to startup-config file.

In this case, the DCIDs are retained as computed for all channels, and are not set to the defaults of the channels. Save the configuration containing the newly-assigned DCIDs to the startup-config file by using the **write memory** command.

When you enable automatic DCID assignment, any DCID conflict arising due to adding a channel to a MAC Domain is resolved automatically.

Restriction

- After running the **cable downstream-channel-id automatic** command in the configuration, manually editing the configuration file in an editor to add RF channels to the fiber nodes could cause DCID conflicts. The feature assumes all channels in fiber nodes have unique automatic DCIDs in global configuration mode. If the configuration is manually edited and the feature does not verify the unique DCIDs, the DCIDs of the newly-added channels may conflict with those of the existing channels. To fix any DCID conflicts, undo and re-apply the global automatic DCID configuration.

  Note Re-applying global automatic DCID configuration is a disruptive operation.

  To avoid DCID conflicts, edit the configuration to configure the fiber nodes, then run the **cable downstream-channel-id automatic** command so all channels have unique automatic DCIDs.

  Make additions to the fiber nodes on the Cisco uBR10012 router command line interface with the automatic DCID configured.

- The **cable downstream-channel-id automatic** command should not be manually edited in to the startup-config file, since it does not guarantee unique DCIDs for channels in the fiber node.

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable downstream-channel-id automatic** **Example:** `Router(config)# cable downstream-channel-id automatic`	Specifies automatic assignment of the DCIDs by the Cisco CMTS.

Example

This example displays the restriction on manually editing configurations:

```
Router# show run | include automatic
cable downstream-channel-id automatic

router# show cable fiber-node 3
-------------------------------------------------------------------------------
Fiber-Node 3
Channel(s) : downstream Integrated-Cable 1/0/2: 0-3, 32-35, 64-67, 96-99
Channel ID(s): 1 2 3 4 33 34 35 36 65 66 67 68 97 98 99 100
Upstream-Cable 1/0/2
FN Config Status: Configured (status flags = 0x01)
MDD Status: Valid
Router#
```

If you manually edit the startup-config file in an editor to add a downstream controller, for example, 1/0/3, it causes a conflict.

```
Router> configure terminal
Router# cable fiber-node 3
Router# downstream integrated-Cable 1/0/3
```

If this downstream controller is added, the automatic DCID assignment feature automatically resolves it. However, since the startup-config file was manually edited to add the downstream controller, the automatic DCID assignment feature is unable to resolve it. This causes a DCID conflict when the edited startup-config file is loaded and invalidates the fiber node.

```
down Modular-Cable 5/0/0 rf-channel 0
DS frequency is not unique.
DS channel id is not unique.
Warning: D3.0 CMs cannot get w-online with an invalid fiber-node.
router#
```

What to do next

Run the **show cable fibernode** command to view DCIDs assigned to all the channels in the fiber node.

```
Router# show cable fiber-node 3
Fiber-Node 3
Channel(s) : downstream Integrated-Cable 1/0/2: 0-3, 32-35, 64-67,
96-99
Channel ID(s): 1 2 3 4 33 34 35 36 65 66 67 68 97 98
99 100
Upstream-Cable 1/0/2
FN Config Status: Configured (status flags = 0x01)
MDD Status: Valid
```

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

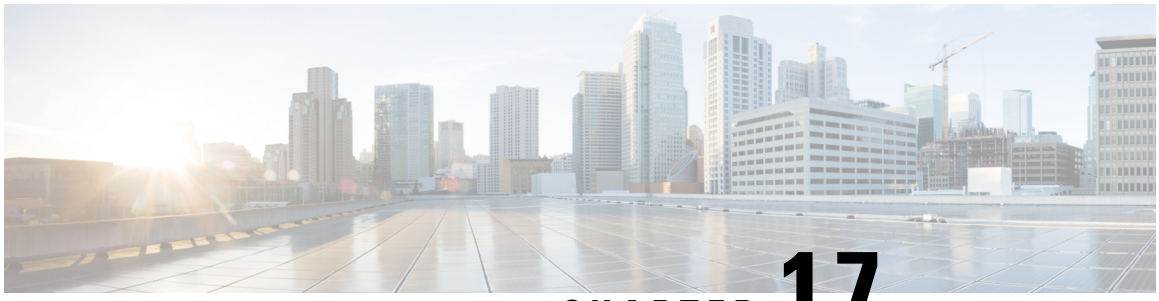
Feature Information for Downstream Channel ID Assignment

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 43: Feature Information for Downstream Channel ID Assignment

Feature Name	Releases	Feature Information
Downstream Channel ID Assignment	Cisco IOS XE Everest 16.5.1	This feature was integrated in to the Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router.



CHAPTER 17

Upstream Channel Bonding

The Upstream Channel Bonding (USCB) feature helps cable operators offer higher upstream (US) bandwidth per cable modem (CM) user by combining multiple radio frequency (RF) channels to form a larger bonding group at the MAC layer.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 303
- Prerequisites for Upstream Channel Bonding , on page 304
- Restrictions for Upstream Channel Bonding , on page 304
- Information About Upstream Channel Bonding, on page 305
- How to Configure Upstream Channel Bonding , on page 314
- Configuration Example for Upstream Channel Bonding , on page 330
- Verifying the Upstream Channel Bonding Configuration, on page 332
- Additional References, on page 333
- Feature Information for Upstream Channel Bonding , on page 333

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 44: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Upstream Channel Bonding

- Enable downstream channel bonding before configuring the Upstream Channel Bonding feature on a Cisco cable modem termination system (CMTS) router.
- Ensure that the CM is registered in Multiple Receive Channel (MRC) mode before configuring upstream channel bonding on a Cisco CMTS router.
- Ensure that the CM is DOCSIS 3.0 certified.

Restrictions for Upstream Channel Bonding

The following are the general restrictions for the Upstream Channel Bonding feature:

- Only the static bonding groups are supported.
- Only the upstream channels belonging to the same MAC domain can be added to an upstream bonding group.

Note A maximum of 16 upstream channels can be configured for each MAC Domain, which are divided into two groups:

- Group 1: upstream channel 0-7
- Group 2: upstream channel 8-15

The **upstream bonding-group** should include all the upstream channels either from Group 1 or Group 2 only.

- Committed information rate (CIR) oversubscription is not supported on USCB groups.

  Cisco CMTS allows oversubscription of the available bandwidth for individual upstream channels. However, oversubscription of bandwidth is not supported for USCB groups.

  An individual upstream may get oversubscribed due to static CIR service flows created for voice traffic. This may cause the DOCSIS 3.0 CMs with USCB to come online on single channel US bonding group (also known as default bonding group).

  This problem is mainly encountered in the voice deployments using static service flows. It is, therefore, recommended to choose from the following voice deployments such that the CIR is allocated (or released) when a voice call is attempted (or dropped):

  1. Dynamic Quality of Service (DQoS) Lite
  2. Packet Cable (PC) DQoS
  3. Packet Cable Multimedia (PCMM)

  These deployments avoid the individual upstream oversubscription and CMs come online on expected bonding groups.

Information About Upstream Channel Bonding

DOCSIS 3.0-based upstream channel bonding is a method for increasing upstream bandwidth up to a maximum of 120 Mbps raw throughput per CM user in a cable communications system that includes a Cisco CMTS router and multiple CMs. The upstream channel bonding method enables a CM to transmit data to a Cisco CMTS router on multiple upstream channels simultaneously.

Channel bonding is a method by which smaller bandwidth upstream channels are bonded together to create a larger upstream bonding group in the MAC domain. A MAC domain is a logical sub-component of a Cisco CMTS router and is responsible for implementing all DOCSIS functions on a set of downstream and upstream channels.

The Upstream Channel Bonding feature supports upstream traffic in Multiple Transmit Channel (MTC) mode for data and video services as these services require more bandwidth than voice-based services. Voice-based services either use the traditional single upstream channel or a single upstream channel bonding group configuration. Any traffic contract that exceeds 30 Mbps requires upstream channel bonding as the physical capacity of a single RF channel in DOCSIS cannot exceed 30 Mbps.

The Upstream Channel Bonding feature is supported on the Cisco cBR-8 router. Upstream data from the subscriber comes through the upstream ports (US0-US19) that are automatically configured on the cable interface line card. The cable interface line card processes the data and sends it across the backplane to the WAN card and out to the Internet.

The table below lists the downstream and upstream frequency supported on the cable interface line card.

Table 45: Downstream and Upstream Frequency

Line Card	Downstream Frequency	Upstream Frequency
Cisco cBR-8 CCAP	55-999 MHz[1]	The upstream frequency range for the Cisco cBR-8 CCAP line card is from 5 to 85 MHz irrespective of the region and Annexure configuration.

[1] This frequency range is subjected to the frequency restriction of the attached EQAM device.

Multiple Transmit Channel Mode

Multiple Transmit Channel mode is a CM capability that enables CMs to send upstream traffic on multiple upstream channels. You can enable the MTC mode on a cable interface line card:

- MTC mode for all CMs in a MAC domain—The MTC mode for all CMs in a MAC domain is enabled by default on an upstream bonding capable cable interface line card.

Multiple Receive Channel Mode

MRC mode is a CM capability that enables CMs to receive downstream traffic on multiple downstream channels. The MRC mode is enabled by default on an upstream bonding capable cable interface line card. You can enable or disable the MRC mode in the MAC domain during or after the CM registration using the **cable mrc-mode** command.

Dynamic Range Window and Transmit Power Levels for Upstream Channel Bonding

The dynamic range window functionality is based on the CableLabs DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification and DOCSIS 3.0 Specification. This requires a DOCSIS 3.0 CM to have upstream transmit channel power level within a 12 dB range for all channels in its transmit channel set (TCS).

DOCSIS 1.x or 2.0 CMs operating with a single upstream channel, in non-MTC mode, have a higher maximum transmit power level than DOCSIS 3.0 CMs operating in the MTC mode with two or more upstream channels. That is, the maximum transmit power level per channel is reduced in the MTC mode.

When the upstream attenuation exceeds the maximum transmit power level, a DOCSIS 3.0 CM attempting to register in the MTC mode may fail to come online, or register in partial mode. The CM fails to register when the transmit power level of all upstream channels in its TCS exceeds the maximum transmit power level. If the CM has some upstream channels that are within the maximum transmit power level, the CM may come online in partial mode. However, the upstream channels that exceed the maximum transmit power level are marked as down and cannot be used for upstream traffic.

To verify the transmit power levels on a CM, use the show cable modem command with the verbose keyword. This command displays the following transmit power values for each assigned upstream channel:

- Reported Transmit Power—This is the reported transmit power level by the CM for each upstream channel.
- Minimum Transmit Power—This is the minimum transmit power level that the CM in the MTC mode could transmit at for the upstream channel.

- Peak Transmit Power—This is the maximum transmit power level that the CM in the MTC mode could transmit at for the upstream channel.

To support upstream channel bonding, the minimum transmit power must be less than or equal to the reported transmit power, and the reported transmit power must be less than or equal to the peak transmit power. The peak transmit power and minimum transmit power levels are derived from the CM TCS assignment and each individual upstream channel configuration.

If the minimum transmit power is higher than the reported transmit power, or the reported transmit power is higher than the peak transmit power, the CM may not come online or may register in partial mode.

You can troubleshoot this transmit power problem in the following two ways:

- Insert an additional amplifier to reduce the upstream attenuation so that the upstream transmit power falls within the allowed transmit power range (12 dB).
- Disable the MTC mode. To switch the CM from the MTC mode to non-MTC mode, disable the bonded-bit (bit-0) in type, length, value (TLV) 43.9.3 using the CM configuration file.

Extended Transmit Power

During the early deployment of DOCSIS 3.0 CMs, additional power is required from the CMs in order to compensate for the attenuation in the upstream path. CMs should transmit at extended power level than that defined in DOCSIS. This scenario is generally observed when USCB is enabled at the Cisco CMTS and the DOCSIS 3.0 CMs are operating in MTC mode.

Additional upstream power provides the operator with a power margin that helps overcome the upstream signal loss, reduces the cable plant operational cost, and enables rapid deployment of DOCSIS 3.0 CMs.

The Cisco CMTS supports the following features with which the CMs can transmit data at an extended power:

- Cisco Extended Transmit Power Feature
- DOCSIS Extended Transmit Power Feature

Cisco Extended Transmit Power Feature

The Cisco Extended Transmit Power feature supports DOCSIS 3.0 CMs operating in MTC mode to transmit at a higher power level than the power level specified in the *DOCSIS 3.0 Specification*. This feature is supported only with Cisco DPC3000 CMs.

The Cisco Extended Transmit Power feature enables cable operators to have better control on the cable modems that register in 4-channel or 2-channel MTC mode or in non-MTC mode to transmit at a higher power level than the DOCSIS-defined maximum power level. The cable operator can configure extended transmit power using the **cable tx-power-headroom** command in global configuration mode.

DOCSIS Extended Transmit Power Feature

The DOCSIS Extended Transmit Power feature supports extended upstream transmit power capability as defined in the DOCSIS3.0 Specification. This feature allows the CMs to transmit at a high extended power level to counter the attenuation in the US channel.

The table below lists the new TLVs supported by the DOCSIS Extended Transmit Power feature.

Table 46: TLVs for DOCSIS Extended Power Feature

TLV Name	Type	Length	Value
Extended Upstream Transmit Power Support	16	1	0—Extended Upstream Transmit Power Support Off 1—Extended Upstream Transmit Power Support On 2-255—Reserved
Extended Upstream Transmit Power CM Capability	5.40	1	0, 205-244 (units of one-quarter dB)

The Cisco CMTS sends TLV16 to inform the CM if the DOCSIS Extended Transmit Power feature is enabled. The CM in turn, sends TLV5.40 to the Cisco CMTS to communicate its extended power capability. After the negotiations are complete, the CM can transmit at an extended power.

DOCSIS Extended Transmit Power feature is enabled by default. Use the cable upstream ext-power command to enable or disable this feature. For more information on how to enable or disable DOCSIS Extended Power feature, see Configuring DOCSIS Extended Transmit Power Feature, on page 329.

Note DOCSIS Extended Transmit Power feature takes precedence, if both Cisco Extended Transmit Power feature and DOCSIS Extended Transmit Power feature are configured.

Reduced Transmit Channel Set

The Reduced Transmit Channel Set feature enables the Cisco CMTS router to reduce upstream channel set assignment based on the total power budget of the CM. For example, a reduction from four to two upstream channels gains 3 dB headroom. Further reduction from two channels to a single channel gains another 3 dB headroom, and the CM starts operating in non-MTC mode.

In order to take advantage of the reduced upstream channel set, the corresponding static bonding groups must be configured. For example, a MAC domain is configured with a bonding group having four channels. A CM with the reduced channel set of two is unable to match to the 4-channel bonding group, and can only be matched to a bonding group with two channels or less.

The Reduced Transmit Channel Set feature is helpful when a DOCSIS 3.0 CM is required to increase its total transmit power by 3 dB. For example, a DOCSIS 1.0 or 2.0 CM supports a maximum transmit power of 58 dBmV for Quadrature Phase Shift Keying (QPSK) modulation, while a DOCSIS 3.0 CM supports a maximum transmit power of 61 dBmV. In this case, the DOCSIS 3.0 CM operating in 4-channel MTC mode has a reduction in the maximum transmit power per upstream channel. This feature enables the Cisco CMTS router to support reduced input power level by 6 dB to prevent upstream path attenuation.

T4 Multiplier

T4 multiplier is the T4 timeout multiplier value of the default T4 timeout values as defined in for cable modems that are in the MTC mode. The default value is derived from the number of channels in the modem transmit channel set. You can change the default T4 multiplier value using the cable upstream ranging-poll command in cable interface configuration mode.

The T4 timeout multiplier values range is from 1 to 10. If the T4 multiplier value is equal to 1, the cable modem will T4 time out in 30 seconds (that is, 1 x 30 = 30). If you change the T4 multiplier to 4, then the new T4 timeout value will be 120 seconds (that is, 4 x 30 = 120).

Note If the T4 timeout multiplier is not configured from the range (1 - 10), then the CMTS uses the T4 timeout value of modem as T4 timeout value. For example, if the T4 timeout of the modem is 90 seconds, then the CMTS applies 3 as the T4 multiplier.

In the MTC mode, you can increase the T4 timeout value in order to reduce the router overhead associated with processing of ranging request (RNG-REQ) slots and ranging response messages. If an RNG-RSP message does not contain a T4 timeout multiplier value, then the CM uses the default T4 timeout value.

Fiber Node Configuration for Upstream Channel Bonding

The fiber node configuration on a Cisco CMTS router is used to define MAC domain downstream service groups (MD-DS-SGs) and MAC domain upstream service groups (MD-US-SGs) as defined in DOCSIS 3.0. Only the DOCSIS 3.0 certified modems use this information.

In hybrid fiber coaxial (HFC) networks, all CMs connected to the same coaxial segment of a fiber node reach the same set of downstream and upstream channels on one or more Cisco CMTS routers located at the headend.

A CM is physically connected to only one fiber node. The fiber node must include at least one primary-capable controller for the CM connected to the fiber node to be operational.

New TLVs for Upstream Channel Bonding

The table below lists the new CableLabs defined type, length, values (TLVs) for the Upstream Channel Bonding feature.

Table 47: New TLVs for Upstream Channel Bonding

TLV Name	Type	Length	Value
CM vendor ID	43.8	3	Per vendor definition
Cable modem attribute mask	43.9	n	Cable modem attribute mask subtype encodings

A Cisco CMTS can have multiple upstream channel bonding groups (USBG) configured. Each of these bonding groups can include upstream channels with different upstream frequencies. Some bonding groups can include channels with frequencies within the extended frequency range (see Table 45: Downstream and Upstream Frequency, on page 306). An HFC network consists of several types of CMs, each supporting standard or extended upstream frequencies.

When you register a CM, the Cisco CMTS does not assign bonding groups based on the upstream frequency range supported by that CM. The assignment of the bonding groups is done to balance the CM count on each of the bonding groups. This may lead to assignment of a bonding group, in the extended frequency range, to a CM that lacks the extended frequency support. As a result, the CM will not be able to register. This scenario is generally observed in the Cisco cBR-8 CCAP line card deployment (containing a mix of CMs), which supports frequency as high as 85MHz (see Table 45: Downstream and Upstream Frequency, on page 306).

If the Cisco CMTS assigns a USBG with a channel within the extended frequency range to a CM limited to the standard frequency range, that CM may not be able to register on that upstream bonding group. Use the TLV 43.9.3 (CM US Required Attribute Mask) or TLV 43.9.4 (CM US Forbidden Attribute Mask) as a workaround. These TLVs enable the Cisco CMTS to assign CM to a USBG, which is in the upstream frequency range supported by that CM.

The default attributes (in hexadecimal) on a CM Attribute Mask (TLV 43.9) are “80 00 00 00", which means by default the mask is all zeroes with the bonding bit enabled. The first four bytes are pre-defined while the last four bytes are user defined. In order to enable Cisco CMTS to assign bonding groups based on the frequency range supported by CMs, complete these steps:

1. Configure a mask, using TLV 43.9.3 or TLV 43.9.4, by modifying the last four bytes. The mask should be configured such that a unique attribute is assigned to each of the bonding groups.
2. Apply this mask to the CM configuration file. CMs supporting extended frequency, can register with any USBGs, irrespective of the configured frequency range of the USBG. CMs supporting standard frequency, can only register with USBGs that are configured with standard frequency range.

Apply the mask you have configured above, to the CMs that support standard or extended frequency ranges. However, the ONLY CMs that need to employ the attribute mask are the ones with the standard frequency range, since they will not be able to register with the USBG configured with extended upstream frequency range. No attribute mask on the extended frequency supporting CMs means that these modems will be assigned any USBG.

The Cisco CMTS uses this mask, received in the CM configuration file during registration, to decide which USBG should be assigned to the CM.

Upstream Weighted Fair Queuing

The upstream weighted fair queuing (WFQ) is a quality of service (QoS) feature that enables the Cisco CMTS router to allocate optimum bandwidth to upstream service flows based on the WFQ parameter configurations. To enable upstream WFQ, you must configure either the class-based or activity-based WFQ on a cable interface.

The following WFQ parameter configurations are supported:

Class-Based Weighted Fair Queuing

In the class-based weighted fair queuing configuration, allocation of available bandwidth is dependent on the service flows that are active in a service class. A service class is a group of queuing attributes configured on the Cisco CMTS router. The class must have at least one active service flow. The class receives its portion of the available bandwidth based on the weight of the class. By default, each class (0 to 7) has a weight of “class + 1.” For example, the class 0 has a weight of 1, and class 1 has a weight of 2.

Activity-Based Weighted Fair Queuing

In the activity-based weighted fair queuing configuration, allocation of available bandwidth is based on the service class and the total number of service flows that are active in a map for the service class. A service class with higher number of service flows receives the larger percentage of bandwidth.

Custom Weight for Service Flow Priorities

The weighted fair queuing functionality helps the Cisco CMTS router share the available bandwidth based on the weight of the service flow priorities specified for outstanding requests from an upstream service flow. Priority refers to the service flow priority specified in the CM configuration file, or the Cisco CMTS service

class configuration. By default, the weight of a priority is equal to "priority+1." For example, priority 0 has a weight of 1, and priority 1 has a weight of 2. A higher priority provides more weight to the outstanding request. The custom weight can be specified for a total of eight priorities (0 to 7) in a service class.

The priority parameter refers to the priority of traffic in a service flow ranging from 0 (the lowest) to 7 (the highest). In the upstream traffic, all of the pending high priority service flows are scheduled for transmission before low priority service flows. You can configure the weight for priorities based on how much weight is appropriate per priority.

The table below lists the default weight for each service flow priority.

Table 48: Default Weight of Service Flow Priorities

Service Flow Priority	Default Weight
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8

Upstream Scheduler and Service Flows

A DOCSIS-qualified Cisco CMTS router can provide varied upstream scheduling modes for different packet streams or applications using upstream service flows. A service flow represents either an upstream or a downstream flow of data. A unique service flow ID (SFID) identifies each service flow. Each service flow can have its own quality of service (QoS) parameters, such as maximum throughput, minimum guaranteed throughput, and priority. In the case of upstream service flows, you can also specify a scheduling mode.

Scheduling is a process that enables the Cisco CMTS router to receive bandwidth requests and grant timeslots to CMs for the upstream traffic. The Cisco CMTS router periodically creates a grant map for each enabled upstream channel. The map grants individual timeslots to enable CMs to place packets on the upstream channels.

DOCSIS 3.0 describes a method by which a CM creates an upstream service flow. The following scheduling types enable the Cisco CMTS router to allocate bandwidth for upstream service flows:

- Unsolicited grant service (UGS)
- Solicited grant service

The unsolicited grant service is primarily used for voice. In the case of UGS, the CM does not have to explicitly request grants from the Cisco CMTS router whereas in the solicited grant service the CM has to explicitly request grants from the Cisco CMTS router. The solicited grant service is primarily used for best effort (BE) services.

Unlike DOCSIS 2.0, DOCSIS 3.0 allows multiple outstanding requests per service flow. For more information about the upstream scheduler, see the *Upstream Scheduler Mode for the Cisco CMTS Routers* feature guide at the following URL:

http://www.cisco.com/en/US/docs/ios/cable/configuration/guide/cmts_upstm_sch_md_ps2209_TSD_Products_Configuration_Guide_Chapter.html

Upstream Service Flow Fairness

The service flows in the same class receive approximately the same amount of bandwidth. Fairness resolves the bandwidth distribution disparity among various service flows including:

- non-bonded service flow vs bonded service flows
- Service flows on modems of different vendors, i.e Intel/TI vs Broadcom
- Service flows associated with different sized bonding groups, i.e. 1,2 4 channels

The upstream scheduler supports flow based queuing. When Upstream Service Flow Fairness is configured, the upstream scheduler determines the order and amount of BW a service flow should receive based on it's current consumption relative to other flows in the flows in the same class.

Use the **cable upstream qos fairness** command to configure the Upstream Service Flow Fairness feature. Use this command in interface configuration mode (or MAC Domain configuration mode).

Distribution of Traffic across all Channels in a USBG

When upstream channel bonding (USCB) is enabled, the Distribution of Traffic across all Channels in a USBG feature can be used to balance the bandwidth utilization across upstream channels on one upstream bonding group.

This feature balances the utilization only if there is one upstream channel bonding group configured per MAC domain.

Restrictions:

- This feature is supported only on one upstream bonding group under a MAC domain. When multiple upstream bonding groups are configured under a MAC domain, the utilization is unfair.
- All the channels must be configured in one upstream bonding group under the same MAC domain.
- This feature is used only for UB-online cable modems.

The USCB Balancing Scheduler may be enabled or disabled using the **cable upstream balance-scheduler** command in the interface (config-if) configuration mode.

DOCSIS 3.0 Load Balancing with USBG Smaller than Cable Modem Capabilities

When using USCB in a service group with USBGs containing fewer upstream channels than the total upstream channel set with DOCSIS 3.0 load balancing enabled, the CMTS can assign a Transmit Channel Set (TCS) to DOCSIS 3.0 cable modems for potential use which falls outside of the configured USBG. The CMTS will try to bind smaller UBGs and default single channel bonding groups into a bigger channel set in order to increase the cable modem services. For example, a DOCSIS 3.0 cable modem receiving the larger TCS can use these additional channels for dynamic service flow addition. The DOCSIS 3.0 Load Balancing feature can also move cable modems to upstream channels that are not explicitly configured with USBGs as a result of the larger TCS.

If you activate DOCSIS 3.0 Load Balancing while using upstream bonding, ensure that the upstream bonding group configuration is embedded and aligned by performing the following:

- Configure USBGs, which is matched to cable modem capabilities within the service group, such as a 4 channel USBG, 2 channel USBG, and 3 channel USBG as applicable.
- Ensure that configured USBGs are optimal for the upstream channel set based on modem capabilities within the service group. For example, if four upstream channels are available, channels 0+1 and 2+3 should each be an USBG to avoid dynamic TCS creating sub optimal bonding scenarios.
- Alternatively, you can choose to shut down any upstream channels that is not configured in USBGs which is not be used for bonding.

Cisco cBR-8 CCAP Line Card Rate Limiting

The rate limiting functionality enables you control the aggregated rate and CPU consumption of upstream traffic for DOCSIS 3.0 bonded service flows on the Cisco cBR-8 CCAP line card. The rate limiting functionality is configured by default on the Cisco cBR-8 CCAP line card. However, the default configuration can be modified using the **cable upstream rate-limit-ccf** command.

The rate limiting functionality uses the following two rate limiting methods:

- Aggregated rate limiting—This is based on Peripheral Component Interconnect (PCI) bus aggregated throughput. The throughput is per line card for all bonded service flows. You can modify the default throughput and burst rate configuration. The maximum allowed throughput is 115 Mbps.
- CPU-based rate limiting—This method controls the CPU consumed by Continuous Concatenation and Fragmentation (CCF) and ensures that the line card functions properly when traffic is overloaded with bonded service flows. The default configuration allocates 50 per cent of CPU to CCF. You can modify the default CPU threshold value and burst rate as required.

SID Tracking

The service ID (SID) tracking functionality enables you to track events related to upstream bandwidth requests and processing of grants. The SID tracker module can track events for a maximum of two service flows per MAC domain. The SID tracker module tracks up to 40,000 events per service flow on a cable interface line card.

You can enable SID tracking for the following types of events:

- DOCSIS 2.0 bandwidth request
- DOCSIS 3.0 bandwidth request
- Grant
- Pending grant (due to traffic congestion)
- Pending grant (due to shaping)

You can enable SID tracking using the **track keyword** along with the **debug cable interface sid** command. To verify SID tracking, use the **show interface cable upstream debug** command in privileged EXEC mode.

Service ID Clusters

A Cisco CMTS router can assign one or more service ID clusters to the upstream bonded service flows (upstream service flows assigned to an upstream bonding group) at the time of service flow creation. A SID

cluster contains one SID per upstream in a bonding group. A CM uses one of the SIDs defined in the SID cluster for the upstream interface when the CM sends a bandwidth request. The CM chooses a SID or a SID cluster based on the SID cluster switching criteria.

For example, assume that a CM has ranged on upstream channels from 1 to 4. The Cisco CMTS router creates a bonded service flow and assigns a single SID cluster to each upstream channel. That is SID1 for UP1, SID2 for UP2, SID3 for UP3, and SID4 for UP4. Now, the CM can send a bandwidth request using any of the four upstream channels. That is, the CM can request bandwidth on any of the upstream interfaces in the SID cluster using the SID defined for the particular upstream. The Cisco CMTS router grants bandwidth to the CM using any combination of upstream channels.

How to Configure Upstream Channel Bonding

Note Before configuring the Upstream Channel Bonding feature, ensure that the fiber node is configured. The fiber node must be configured in accordance with the physical plant topology.

The following tasks describe how to configure Upstream Channel Bonding on the Cisco cBR-8 router:

Enabling MTC Mode on a Cisco CMTS Router

To section explains how to enable the MTC mode on a Cisco CMTS router.

Default MTC Mode Configuration on a Cisco CMTS Router

By default, the MTC mode is configured on a cable interface line card. With this default configuration, the Cisco CMTS router enables the MTC mode on a per MAC domain basis depending on the configuration file of each CM. When the CM configuration file has the bonded-bit (bit-0) enabled in TLV 43.9.3 (cable modem upstream required attribute mask), the Cisco CMTS router enables the CM to come online in the MTC mode. If the CM configuration file does not have the bonded-bit on, the CM comes online in non-MTC mode.

For more information on how to add the required attribute in the CM configuration file, see Example: Enabling MTC Mode for a Single CM Using the CM Configuration File, on page 331.

Enabling MTC Mode for All CMs

Note This MTC mode configuration supersedes the default MTC mode configuration (per CM basis) with the required attribute. To disable the MTC mode for all CMs in a MAC domain, use the **no** form of the **cable mtc-mode** command. If the MTC mode is enabled and the forbidden mask of the upstream bonding in TLV 43.9.4 is disabled, the CM does not support the Upstream Channel Bonding feature.

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** {*slot/subslot/port* | *slot/subslot/cable-interface-index* | *slot/port* | *slot/cable-interface-index*} **Example:** `Router(config)# interface cable 7/0/0`	Specifies the cable interface line card on a Cisco CMTS router.
Step 4	**cable mtc-mode** **Example:** `Router(config-if)# cable mtc-mode`	Enables MTC mode at the MAC interface for all CMs.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits cable interface configuration mode and returns to privileged EXEC mode.

Configuring UCSB Required Attribute

If the CM configuration file has TLV 43.9.3 (CM upstream required attribute mask) configured and bonded bit is set to 1, then the modem comes UB-online on a MAC domain basis. If the CM configuration file has no TLV 43.9.3 or the bonded bit is not set to 1, then the modem comes online with a single upstream channel on a MAC domain basis.

Note Without this configuration, the modem comes UB-online on the MAC domain regardless of whether the TLV 43.9.3 is configured in the modem configuration file.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface cable** { *slot/subslot/port* | *slot/subslot/cable-interface-index* | *slot/port* | *slot/cable-interface-index* } **Example:** `Router(config)# interface cable 7/0/0`	Specifies the cable interface line card on a Cisco CMTS router.
Step 4	**cable mtc-mode required-attribute** **Example:** `Router(config-if)# cable mtc-mode required-attribute`	Enable enforcement of required CM attribute on UCSB.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits cable interface configuration mode and returns to privileged EXEC mode.

Creating a Bonding Group

An upstream bonding group is created by combining multiple upstream channels together on a cable interface line card.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** {*slot/subslot/port* | *slot/subslot/cable-interface-index* | *slot/port* | *slot/cable-interface-index*} **Example:** `Router(config)# interface cable 7/0/0`	Specifies the cable interface line card on a Cisco CMTS router.
Step 4	**cable upstream bonding-group** *id* **Example:** `Router(config-if)# cable upstream bonding-group 200`	Creates the bonding group on the specified cable interface.
Step 5	**end** **Example:**	Exits cable interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config-if)# end`	

What to do next

After creating an upstream bonding group, you must add upstream channels to the bonding group.

Adding Upstream Channels to a Bonding Group

Restriction DOCSIS 3.0-certified CMs support only four upstream channels on an upstream bonding group. These CMs do not accept additional upstream channels that are added to a bonding group.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** {*slot/subslot/port* | *slot/subslot/cable-interface-index* | *slot/port* | *slot/cable-interface-index*} **Example:** `Router(config)# interface cable 7/0/0`	Specifies the cable interface line card on a Cisco CMTS router.
Step 4	**cable upstream bonding-group** *id* **Example:** `Router(config-if)# cable upstream bonding-group 200`	Creates the bonding group on the specified interface.
Step 5	**upstream** *number* **Example:** `Router(config-upstream-bonding)# upstream 1`	Enters upstream bonding configuration submode and adds an upstream channel to the upstream bonding group. **Note** Upstream channel needs to be bonded to mac-domain first before adding it to the bounding group. For detailed configuration steps of the upstream channel bonding, please refer to Configuration Example for Upstream Channel Bonding

	Command or Action	Purpose
		A maximum of 16 upstream channels can be configured for each MAC Domain, which are divided into two groups: • Group 1: upstream channel 0-7 • Group 2: upstream channel 8-15 The **upstream bonding-group** should include all the upstream channels either from Group 1 or Group 2 only.
Step 6	**end** **Example:** `Router(config-upstream-bonding)# end`	Exits upstream bonding configuration submode and returns to privileged EXEC mode.

Adding Upstream Channel Ports to a Fiber Node

You must add upstream channel controllers to a fiber node in order to complete the basic upstream channel bonding configuration on a cable interface line card. The fiber node must contain all upstream and downstream controllers reached by the CMs.

Restriction

- Configuration of a fiber node is valid only if all upstream channels inside the fiber node have different upstream frequencies.
- For any two upstream channels mapped to the upstream cable controllers in the same fiber node where a spectrum group is assigned to one upstream channel, and a frequency is assigned to the other upstream channel, any overlap between any bands associated with the spectrum group of the upstream channel and the frequency of the upstream channel will result in an invalid fiber node configuration. That is a fixed frequency cannot overlap with another upstream channel's available spectrum group bands.

Note The fiber node configuration must be done in accordance with the physical plant topology.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**cable fiber-node** *fiber-node-id* **Example:** `Router(config)# cable fiber-node 2`	Enters fiber node configuration mode.
Step 4	**upstream Upstream-Cable** *slot/subslot/port* **Example:** `Router(config-fiber-node)# upstream Upstream-Cable 7/0/1`	Specifies the upstream channel ports for a fiber node.
Step 5	**end** **Example:** `Router(config-fiber-node)# end`	Exits fiber node configuration mode and returns to privileged EXEC mode.

Configuring the Class-Based Weighted Fair Queuing

In the case of a class-based configuration, allocation of available bandwidth is dependent on the service flows that are active in a service class.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** {*slot/subslot/port* | *slot/subslot/cable-interface-index* | *slot/port* | *slot/cable-interface-index*} **Example:** `Router(config)# interface cable 7/0/0`	Specifies the cable interface line card on a Cisco CMTS router.
Step 4	**cable upstream qos wfq class** **Example:** `Router(config-if)# cable upstream qos wfq class`	Enables class-based weighted fair queuing.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits cable interface configuration mode and returns to privileged EXEC mode.

Configuring the Activity-Based Weighted Fair Queuing

In the activity-based configuration, allocation of available bandwidth is based on the service class and the total number of service flows that are active in a map for the service class.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** {*slot/subslot/port* | *slot/subslot/cable-interface-index* | *slot/port* | *slot/cable-interface-index*} **Example:** `Router(config)# interface cable 7/0/0`	Specifies the cable interface line card on a Cisco CMTS router.
Step 4	**cable upstream qos wfq activity** **Example:** `Router(config-if)# cable upstream qos wfq activity`	Enables activity-based weighted fair queuing.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits cable interface configuration mode and returns to privileged EXEC mode.

Configuring Custom Weights for Service Flow Priorities

The WFQ functionality helps the Cisco CMTS router share the available bandwidth based on the weight of the service flow priorities specified for outstanding requests from an upstream service flow.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**interface cable** {*slot/subslot/port* | *slot/subslot/cable-interface-index* | *slot/port* | *slot/cable-interface-index*} **Example:** `Router(config)# interface cable 7/0/0`	Specifies the cable interface line card on a Cisco CMTS router.
Step 4	**cable upstream qos wfq weights** *priority0-priority7* **Example:** `Router(config-if)# cable upstream qos wfq weights 10 20 30 40 50 60 70 80.`	Enables custom weight configuration for all the service flow priorities in a service class. **Note** You must specify custom weight values for all the eight service flow priorities (0 to 7) when you modify the default weights of priorities. The valid range is from 1 to 255.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits cable interface configuration mode and returns to privileged EXEC mode.

Configuring the SID Cluster

This section explains how to configure and assign a SID cluster to an upstream bonded service flow.

Note Configure the **cable sid-cluster-group num-of-cluster 2** command to achieve desired upstream bonded speeds. Alternatively, use a large upstream Max Traffic burst value in the cable modem file (such as 30 kB). The Max Concat burst value in the cable modem file need not be changed because DOCSIS 3.0 uses continuous concatenations and fragmentation (CCF) and can therefore use the default value of 3044 in the Max Concat field.

Note If the **cable sid-cluster-group** command is not used, the router accepts the default SID cluster configuration. By default, only one SID cluster is configured. Similarly, if the **cable sid-cluster-switching** command is not used, the router accepts the default SID cluster switchover criterion. That is, only one request can be made using the SID cluster.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** {*slot/subslot/port* | *slot/subslot/cable-interface-index* | *slot/port* | *slot/cable-interface-index*} **Example:** `Router(config)# interface cable 7/0/0`	Specifies the cable interface line card on a Cisco CMTS router.

Step 4	**cable sid-cluster-group [dynamic | req-multiplier** *value* **| num-of-cluster** *number*]
	Example:
	`Router(config-if)# cable sid-cluster-group dynamic`
	`Router(config-if)# cable sid-cluster-group req-multiplier 12`
	`Router(config-if)# cable sid-cluster-group num-of-cluster 2`
		Creates a SID cluster group.

Step 5	**cable sid-cluster-switching [max-outstanding-byte** *value* **| max-request** *value* **| max-time** *seconds* **| max-total-byte** *value***]**
	Example:
	`Router(config-if)# cable sid-cluster-switching max-outstanding-byte 4444`
	`Router(config-if)# cable sid-cluster-switching max-request 222`
	`Router(config-if)# cable sid-cluster-switching max-time 444`
	`Router(config-if)# cable sid-cluster-switching max-total-byte 67890`
		Specifies SID cluster switchover criteria.

Step 6	**end** **Example:** `Router(config-if)# end`	Exits cable interface configuration mode and returns to privileged EXEC mode.

What to do next

Use the **show running-config all** command to verify the SID cluster configuration. Following is a sample output of the command:

```
Router# show running-config all
.
.
.
cable sid-cluster-group num-of-cluster 1
cable sid-cluster-group dynamic
cable sid-cluster-group req-multiplier 4
```

Configuring the Channel Timeout for a Cable Modem

The channel timeout configuration allows you to specify the maximum time that a CM can spend performing initial ranging on the upstream channels described in the Registration Response (REG-RSP) and REG-RSP-MP messages. The default channel timeout value (60 seconds) is automatically configured.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** {*slot/subslot/port* | *slot/subslot/cable-interface-index* | *slot/port* | *slot/cable-interface-index*} **Example:** `Router(config)# interface cable 7/0/0`	Specifies the cable interface line card on a Cisco CMTS router.
Step 4	**cable init-channel-timeout** *value* **Example:** `Router(config-if)# cable init-channel-timeout 160`	Specifies the maximum time that a CM can spend performing initial ranging on the upstream channels.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits cable interface configuration mode and returns to privileged EXEC mode.

Configuring Cable Upstream Resiliency

The cable upstream resiliency module ensures that a CM remains operational if one or more non-primary upstream service flows of the CM enter temporary or persistent error states. This module enables a Cisco CMTS router to handle various events and maintain the transmit channel set of each CM.

In the event of the primary upstream service flow failure, the upstream resiliency module forces the CM to go offline.

For a Multiple Transmit Channel (MTC) modem, the (NRTPS), Real-time Polling Service (RTPS), (UGS), and (UGS-AD) upstream service flows on an impaired upstream channel is moved to another good upstream channel in the cable modem without resetting the cable modem.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable upstream resiliency data-burst polling-interval** *number* **Example:** `Router(config)# cable upstream resiliency data-burst polling-interval 60`	Configures the polling interval for data-burst resiliency in seconds. The range is from 5 to 3600. The default configuration for polling-interval is 60.
Step 4	**interface cable** {*slot/subslot/port* | *slot/subslot/cable-interface-index* | *slot/port* | *slot/cable-interface-index*} **Example:** `Router(config)# interface cable 7/0/0`	Specifies the cable interface line card on a Cisco CMTS router.

Step 5	**cable upstream resiliency** {**channel-down-detect** *number* | **data-burst snr** *number* **ufec** *number* **cfec** *number* **hysteresis** *number* | **modem-offline-detect** *number* | **on-failure** {**disable-channel** | **extended-ranging** | **reset-modem**} | **sf-move** {**NRTPS** | **RTPS** | **UGS** | **UGS-AD**} }
	Example:
	`Router(config-if)# cable upstream resiliency channel-down-detect 68`
	`Router(config-if)# cable upstream resiliency modem-offline-detect 16`
	`Router(config-if)# cable upstream resiliency on-failure disable-channel`
	`Router(config-if)# cable upstream resiliency sf-move NRTPS`
	`Router(config-if)# cable upstream resiliency sf-move RTPS`
		Configures upstream resiliency for bonded upstream service flows.

	Command or Action	Purpose
	Router(config-if)# **cable upstream resiliency sf-move UGS** Router(config-if)# **cable upstream resiliency sf-move UGS-AD** Router(config-if)# **cable upstream resiliency data-burst snr 24 ufec 1 cfec 0 hysteresis 3**	
Step 6	**end** **Example:** Router(config-if)# **end**	Exits cable interface configuration mode and returns to privileged EXEC mode.

Configuring Rate Limiting on the Cisco cBR-8 CCAP Line Card

The rate limiting functionality is configured by default on the Cisco cBR-8 CCAP line card. However, the default configuration can be modified using the cable upstream rate-limit-ccf command.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.
Step 3	**cable upstream rate-limit-ccf** [**aggregated-burst** *value* | **aggregated-throughput** *value* | **cpu-burst** *value* | **cpu-threshold** *value*] **Example:** Router(config)# **cable upstream rate-limit-ccf aggregated-burst 25000** Router(config)# **cable upstream rate-limit-ccf aggregated-throughput 540000** Router(config)# **cable upstream rate-limit-ccf cpu-burst 30** Router(config)# **cable upstream rate-limit-ccf cpu-threshold 60**	Configures rate limiting parameters for upstream bonded service flows on a cable interface line card.

	Command or Action	Purpose
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Enabling Upstream Related Events for CM Status Reports

You can enable upstream related CM status events only on a cable interface line card. You can enable the following upstream related CM status events per interface using the cable cm-status enable command:

- T4 time-out
- T3 re-tries exceeded
- Successful ranging after T3 re-tries exceeded

For details on how to enable upstream and downstream related CM status events, see the Wideband Modem Resiliency feature guide at the following URL:

http://www.cisco.com/en/US/docs/ios/cable/configuration/guide/ubr_wm_resiliency.html

Modifying the Bonding Group Attributes

Bonding group attributes are automatically configured for each upstream bonding group. You can modify them using the **attributes** command in upstream bonding configuration mode.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** {*slot/subslot/port* | *slot/subslot/cable-interface-index* | *slot/port* | *slot/cable-interface-index*} **Example:** `Router(config)# interface cable 7/0/0`	Specifies the cable interface line card on a Cisco CMTS router.
Step 4	**cable upstream bonding-group** *id* **Example:** `Router(config-if)# cable upstream bonding-group 200`	Creates the bonding group on the specified cable interface and enters the upstream bonding configuration mode.

	Command or Action	Purpose
Step 5	**attributes** *value* **Example:** `Router(config-upstream-bonding)# attributes eeeeeeee`	Modifies the attribute value for the specified bonding group.
Step 6	**end** **Example:** `Router(config-upstream-bonding)# end`	Exits upstream bonding configuration mode and returns to privileged EXEC mode.

Modifying the Ranging Poll Interval on Upstream Channels

You can change the default ranging poll interval (20 seconds) on upstream channels using the cable upstream ranging-poll command in cable interface configuration mode. You can also specify the T4 timeout multiplier value using this command.

For information on T4 Multiplier, see T4 Multiplier, on page 308 .

Note We recommend that you do not modify the default ranging poll interval unless required. With the default configuration, a DOCSIS 2.0 CM in non-MTC mode performs ranging on one upstream channel every 20 seconds.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** {*slot/subslot/port* | *slot/subslot/cable-interface-index* | *slot/port* | *slot/cable-interface-index*} **Example:** `Router(config)# interface cable 7/0/0`	Specifies the cable interface line card on a Cisco CMTS router.
Step 4	**cable upstream ranging-poll** [**interval** *value* | **t4-multiplier** *timeout_value*] **Example:**	Specifies the ranging poll interval for upstream channels.

	Command or Action	Purpose
	`Router(config-if)# cable upstream ranging-poll interval 24000 t4-multiplier 4`	**Note** If **t4-multiplier** *timeout_value* is not configured, then the CMTS uses the the T4 timeout of the modem. For example, if the T4 timeout of the modem is 90 seconds, then the CMTS will apply 3 as T4 multiplier for the modem.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits cable interface configuration mode and returns to privileged EXEC mode.

Configuring the Reduced Channel Set Assignment

You need to configure the transmit power offset budget to enable the Cisco CMTS router to reduce upstream channel set assignment based on the total power budget of the CM.

Note The threshold value specified for the power budget offset (max-channel-power-offset) must be less than the power threshold value (power-adjust continue) that determines the value of the Ranging Status field in the Ranging Response (RNG-RSP) messages that the Cisco CMTS router sends to the CM. You can specify the power threshold value using the **cable upstream power-adjust** command.

Before you begin

- Configure extended transmit power using the **cable tx-power-headroom** command in global configuration mode.
- Ensure that corresponding static bonding groups are configured.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** {*slot/subslot/port* | *slot/subslot/cable-interface-index* | *slot/port* | *slot/cable-interface-index*} **Example:**	Specifies the cable interface line card on a Cisco CMTS router.

	Command or Action	Purpose
	`Router(config)# interface cable 7/0/0`	
Step 4	**cable upstream max-channel-power-offset** *dB-value* **Example:** `Router(config-if)# cable upstream max-channel-power-offset 2`	Specifies the power offset value for upstream channels.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits cable interface configuration mode and returns to privileged EXEC mode.

Configuring DOCSIS Extended Transmit Power Feature

The DOCSIS Extended Transmit Power feature is enabled by default on the Cisco CMTS. However, the default configuration can be modified using the cable upstream ext-power command.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** {*slot/subslot/port* | *slot/subslot/cable-interface-index* | *slot/port* | *slot/cable-interface-index*} **Example:** `Router(config)# interface cable 7/0/0`	Specifies the cable interface line card on a Cisco CMTS router.
Step 4	**cable upstream ext-power** **Example:** `Router(config-if)# cable upstream ext-power`	Enables the DOCSIS Extended Transmit Power feature on the Cisco CMTS. Using the **no** form of this command disables the DOCSIS Extended Transmit Power feature.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

The following debug commands help you troubleshoot an improper upstream channel bonding configuration and its related features:

- **debug cable cm-status**—Provide debugging information about CM status messages on the Cisco CMTS routers.
- **debug cable mdd**—Provides debugging information about MAC domain descriptor (MDD).
- **debug cable md-sg**—Provides information about service group debugging messages.
- **debug cable ubg**—Provides debugging information about upstream bonding groups.

Configuration Example for Upstream Channel Bonding

The following example shows how to configure the basic upstream channel bonding on the Cisco cBR-8 CCAP line card interface 7/0/0 on the Cisco cBR-8 router:

```
controller Upstream-Cable 7/0/0
us-channel 0 frequency 10000000
us-channel 0 channel-width 3200000 3200000
us-channel 0 ingress-noise-cancellation 50
us-channel 0 docsis-mode atdma
us-channel 0 minislot-size 2
us-channel 0 modulation-profile 221
us-channel 0 equalization-coefficient
no us-channel 0 shutdown
us-channel 1 frequency 16400000
us-channel 1 channel-width 6400000 6400000
us-channel 1 ingress-noise-cancellation 50
us-channel 1 docsis-mode atdma
us-channel 1 minislot-size 1
us-channel 1 modulation-profile 221
us-channel 1 equalization-coefficient
no us-channel 1 shutdown
us-channel 2 frequency 22800000
us-channel 2 channel-width 6400000 6400000
us-channel 2 docsis-mode atdma
us-channel 2 minislot-size 1
us-channel 2 modulation-profile 221
us-channel 2 equalization-coefficient
no us-channel 2 shutdown
us-channel 3 frequency 29200000
us-channel 3 channel-width 6400000 6400000
us-channel 3 docsis-mode atdma
us-channel 3 minislot-size 1
us-channel 3 modulation-profile 221
us-channel 3 equalization-coefficient
no us-channel 3 shutdown
us-channel 4 channel-width 1600000 1600000
us-channel 4 docsis-mode tdma
us-channel 4 minislot-size 4
us-channel 4 modulation-profile 21
us-channel 4 shutdown
us-channel 5 channel-width 1600000 1600000
us-channel 5 docsis-mode atdma
us-channel 5 minislot-size 4
us-channel 5 modulation-profile 221
us-channel 5 shutdown
!
```

```
interface Cable7/0/0
load-interval 30
downstream Integrated-Cable 7/0/0 rf-channel 0
downstream Integrated-Cable 7/0/0 rf-channel 8
downstream Integrated-Cable 7/0/0 rf-channel 16
upstream 0 Upstream-Cable 7/0/0 us-channel 0
upstream 1 Upstream-Cable 7/0/0 us-channel 1
upstream 2 Upstream-Cable 7/0/0 us-channel 2
upstream 3 Upstream-Cable 7/0/0 us-channel 3
no cable upstream 0 equalization-error-recovery
no cable upstream 1 equalization-error-recovery
no cable upstream 2 equalization-error-recovery
no cable upstream 3 equalization-error-recovery
cable upstream 7 attribute-mask 1FF
cable upstream bonding-group 1
upstream 0
upstream 1
upstream 2
attributes 80000000
cable bundle 1
cable map-advance static 2000
cable sync-interval 121
cable reduction-mode mta-battery enable
cable privacy accept-self-signed-certificate
end

cable fiber-node 1
description Feed Mac Domain: Cable7/0/0
downstream Integrated-Cable 7/0/0
upstream Upstream-Cable 7/0/0
```

Note Bonded channels are typically from the same connector; however, channels from different connectors in the same MAC domain can also be bonded together. A single MAC domain can support multiple channel bonding groups.

Note Up to 8 frequencies can be stacked to one upstream-cable controller. Once the upstream-cable controller has 8 frequencies stacked, no more frequency left for the adjacent upstream-cable controller.

Example: Enabling MTC Mode for a Single CM Using the CM Configuration File

The following example shows how to enable the MTC required attribute using the CM configuration file:

```
03 (Net Access Control) = 1
Unknown Type 005 = 01 01 01
18 (Maximum Number of CPE) = 4
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference) = 1
   S06 (QoS Parameter Set Type) = 7
   S10 (Min Reserved Traffic Rate)= 500000
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference) = 2
```

```
   S06 (QoS Parameter Set Type) = 7
   S10 (Min Reserved Traffic Rate) = 1000000
29 (Privacy Enable) = 0
43 (Vendor Specific Options)
   S08 (Vendor ID) = ff ff ff
   S009 (Unknown sub-type) = 03 04 80 00 00 00
```

Verifying the Upstream Channel Bonding Configuration

Use the following **show** commands to verify the upstream channel bonding configuration:

- **show cable mac-domain upstream-service-group**
- **show cable fiber-node**
- show interface cable upstream
- **show interface cable service-flow**
- **show cable modem**

To verify the runtime statistics of the upstream service group on a cable interface line card, use the **show cable mac-domain upstream-service-group** command.

To verify the configuration of a fiber node, use the **show cable fiber-node** command.

To verify the bonding groups configured on a cable interface line card, use the **show interface cable upstream** command.

To verify upstream bonding information on a cable interface line card, use the **show interface cable service-flow** command.

To verify the transmit power levels on a CM, use the **show cable modem** command.

Verifying Weighted Fair Queuing for Upstream Service Flows

To verify WFQ parameters configured for upstream service flows on a cable interface line card, use the **show interface cable mac-scheduler** command.

Verifying Rate Limiting for Upstream Bonded Service Flows

To verify the rate limiting criteria configured on the Cisco cBR8 CCAP line card for upstream bonded service flows, use the **show cable rate-limit-ccf** command.

Note The **show cable rate-limit-ccf** command is applicable only to the Cisco cBR8 CCAP cable interface line card.

Verifying Extended Power Transmission

To verify that a CM is transmitting at a higher power level, use the **show cable modem** command.

To list all the CMs that are transmitting at higher power level, use the **show cable modem extended-power** command.

Additional References

The following sections provide references related to the Upstream Channel Bonding feature.

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

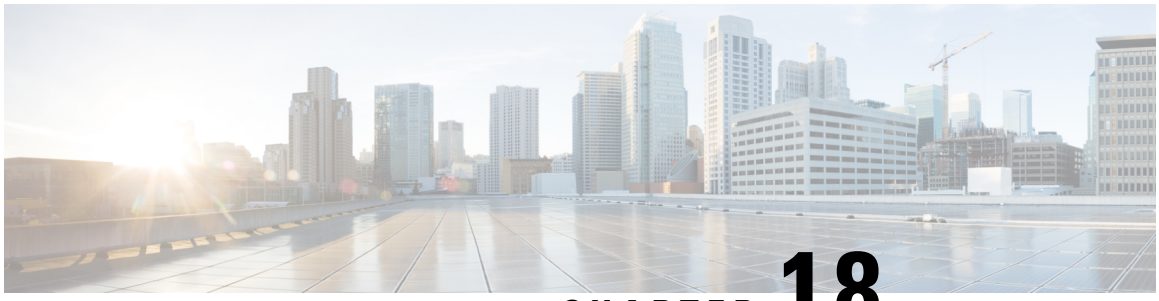
Feature Information for Upstream Channel Bonding

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 49: Feature Information for Upstream Channel Bonding

Feature Name	Releases	Feature Information
Upstream Channel Bonding	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router.



CHAPTER 18

Spectrum Management and Advanced Spectrum Management

This chapter describes the spectrum management features supported for the Cisco Cable Modem Termination System (CMTS) routers. Spectrum management support is divided into two main groups:

- Guided and scheduled spectrum management features (supported in software)
- Intelligent and advanced spectrum management features (supported in hardware only on specific cable interfaces)

- Finding Feature Information, on page 335
- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 336
- Prerequisites for Spectrum Management, on page 336
- Restrictions for Spectrum Management, on page 337
- Information About Spectrum Management, on page 339
- How to Configure Spectrum Management, on page 354
- Monitoring Spectrum Management, on page 373
- Configuration Examples, on page 378
- Additional References, on page 385
- Feature Information for Spectrum Management and Advanced Spectrum Management, on page 386

Finding Feature Information

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 50: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Spectrum Management

Ensure that your network is designed to support reliable broadband data transmission. At minimum, your network must include:

- A Dynamic Host Configuration Protocol (DHCP) server to assign IP addresses to cable modems or set-top boxes on the hybrid fiber-coaxial (HFC) network. This can be a Cisco CMTS router that has been configured to act as the DHCP server.
- If you are not using cable interface line cards with integrated upconverters, you must install the appropriate IF-to-RF external upconverter between the Cisco CMTS router and the combiner.

Note The term “combiner” refers to all cables, amplifiers, and taps at the headend or cable distribution center that connect the Cisco CMTS router to the HFC network.

- Diplex filters installed in the downstream RF path between the cable modems and the cable interface cards in the router. RG-59 headend coaxial cable with the maximum braid available (60 percent + 40 percent braid), double foil, and the correct connector for this cable.
- Avoid frequencies with known ingress problems such as amateur radio bands or short-wave bands.
- Avoid hostile spectrums below 20 MHz.
- When designing your channel plan, allow extra bands for frequency hopping.
- Use the receive power level setting to perform slight equalization adjustments.
- Due to the nature of CATV technology, upstream noise management is a significant issue. We recommend that you follow the rigorous North American plant maintenance procedures documented in the NCTA Supplement on Upstream Transport Issues (available from the National Cable and Telecommunications Association, https://www.ncta.com) to adjust return amplifiers and lasers.

Restrictions for Spectrum Management

This section describes the restrictions for the following spectrum management features:

Shared Spectrum Groups

- Advance spectrum management does not support inter-line-card shared spectrum groups.
- Guided spectrum management does support inter-line-card shared spectrum groups.

Dynamic Upstream Modulation

- The Cisco CMTS router has one preconfigured (primary) modulation profile that defines a typical profile for quadrature phase-shift keying (QPSK) modulation. To use the Dynamic Upstream Modulation feature, you must create a secondary modulation profile that has a higher modulation scheme than the preconfigured profile. The Three Step Dynamic Modulation feature allows you to create and use a third modulation profile. However, the third modulation profile is optional.
- Upstream modulation profiles are assigned to upstream ports and affect all cable modems on those upstream ports.
- Modulation profiles affect the physical layer of the cable network, so only trained technicians who are familiar with the Data-over-Cable Service Interface Specifications (DOCSIS) specifications should create modulation profiles.
- When using the Dynamic Upstream Modulation feature with Voice over IP (VoIP) services, frequent changes to the upstream modulation or channel width could briefly impact the quality of voice calls.

Fixed-Frequency Spectrum Groups with Advanced Spectrum Management

Do the following to configure fixed-frequency spectrum groups:

```
Router(config)#controller upstream-cable 9/0/15
Router(config-controller)#us-channel 0 spectrum-group n
Router(config-controller)#us-channel 0 channel-width 1600000
```

Limitations on Upstream Modulation Parameters for PacketCable VoIP Calls

We recommend the use of a channel width that is 1.6Mhz, 3.2Mhz, or 6.4Mhz while configuring upstreams for PacketCable operations and VoIP calls. (All DOCSIS channel widths and upstream parameter combinations are supported, but not optimum when offering VoIP.)

N+1 Redundancy Support

N+1 redundancy requires the working and protect cable interface line cards to be identical. This ensures that the protect interface supports the same exact configuration as the working interface.

When protecting cards that support intelligent and advanced spectrum management, a switchover preserves the spectrum management configuration, and the protect interface initially uses the same upstream frequency as the working interface. The protect interface does not begin using the advanced spectrum management features until the system stabilizes to avoid any unnecessary frequency hops or channel width changes.

Intelligent and Advanced Spectrum Management Support

- Cable interfaces use standard DOCSIS, EuroDOCSIS, and the extended Japanese frequency ranges (5 to 85 MHz for upstream interfaces) to support the intelligent and advanced spectrum management features.
- Intelligent and advanced spectrum management features are supported only in the DOCSIS 1.0 and DOCSIS 1.1 Time Division Multiple Access (TDMA) mode of operation. These features cannot be used when a cable interface is operating in the DOCSIS 2.0 mixed, Advanced TDMA (A-TDMA), and Synchronous Code Division Multiple Access (S-CDMA) modes of operation. Similarly, these features are also not available when the cable interface is configured to use multiple logical channels. However, these restrictions do not apply for guided spectrum management.
- Upstream channels must meet the carrier-to-noise plus interference ratio (CNiR [CNR]), and carrier-to-ingress power ratio values given in the DOCSIS specifications. The minimum value for both parameters is 25 dB in the 5 to 65 MHz frequency range.
- The intelligent and advanced spectrum management features do not support inter-line card shared spectrum groups. Spectrum management features require that upstream ports on different line cards have their own RF domain (a unique set of non-overlapping frequencies).
- N+1 redundancy is not supported on any cable interface line card that has defined spectrum groups, which typically is the normal configuration for advanced spectrum management.
- The intelligent and advanced spectrum management feature is activated by assigning spectrum groups on cards with built-in spectrum analyzer.

Information About Spectrum Management

Spectrum management allows a Cisco Cable Modem Termination System (CMTS) to sense upstream plant impairments, report them to a management entity, and automatically correct them where possible. The spectrum management feature performs these functions without reducing throughput or latency and without creating additional packet overhead on the radio frequency (RF) plant.

In particular, because the cable interfaces on the router receive upstream packets, it can directly detect upstream transmission errors. The router can also indirectly monitor the condition of the plant by keeping a record of modem state changes, such as the number and frequency of cable modems that are “flapping” (modems that either miss a station maintenance message or that go offline and then come back online).

Note For more information about the cable modem flapping and how to monitor the cable modem flap list, see the Flap List Troubleshooting for the Cisco CMTS Routers .

Spectrum management can prevent long-term service interruptions caused by upstream noise events in the cable plant. It is also used for fault management and troubleshooting the cable network. When cable modems are detected to go online and offline by flap detectors, the cable operators can look at the flap list and spectrum tables to determine the possible causes.

Because of the nature of cable television (CATV) technology, upstream noise management is a significant issue. Frequency bands must have a sufficient CNR (CNiR) and carrier-to-ingress power ratio to support the transmission of QPSK and quadrature amplitude modulation (QAM) data. The DOCSIS sets the minimum value for both of these ratios to 25 dB in the 5 to 65 MHz frequency range. If the CNR (CNiR) drops below 25 dB on a particular channel due to noise, the cable modem on that channel degrades and can drop off the hybrid fiber-coaxial (HFC) network.

This overview contains the following subsections:

- Spectrum Management Measurements, on page 340—Provides an overview of fundamental concepts and terms that are used in spectrum management.
- Upstream Signal Channel Overview, on page 343—Describes how signals are sent and how changes occur in upstream channels.
- Upstream Segments and Combiner Groups, on page 344—Describes sparse and dense segments and combiner groups.
- Frequency Management Policy, on page 345—Describes the types of noise impairments and how to counteract ingress noise with spectrum groups and frequency hopping.
- Guided and Scheduled Spectrum Management, on page 347—Describes the following guided and scheduled spectrum management features: frequency hopping capabilities, dynamic upstream modulation (signal-to-noise ratio-based), and input power levels.
- Intelligent and Advanced Hardware-Based Spectrum Management, on page 352—Describes spectrum management features that are supported by a number of cable interface line cards that have onboard spectrum management hardware. These features include a real-time spectrum analyzer, CNR-based, proactive frequency hopping, and a more robust dynamic upstream modulation.
- Benefits, on page 352—Describes the spectrum management features provided on the Cisco CMTS router platforms.

Spectrum Management Measurements

Measuring the signal-to-noise ratio (SNR [MER]) and carrier-to-noise ratio (CNR [CNiR]) are the major ways of determining the quality of a downstream or upstream signal. The following sections provide an overview of these two ratios, as well as explaining the differences between them, and some additional values that might be useful:

Signal and Carrier Noise Ratios

Measuring the Modulation Error Ratio (MER [SNR]) and CNR (CNiR) of a downstream or upstream is the first step in determining the quality of the signal, and whether spectrum management needs to be performed to correct any errors. The following are brief descriptions of these two values:

- Modulation Error Ratio (MER [SNR])—This is an estimate of the signal strength on the upstream after ingress noise cancellation is performed. This means that the MER (SNR) takes into account a variety of modulation impairments, including frequency response distortions (such as in-channel amplitude tilt and ripple), group delay, microreflections, and phase noise. The MER (SNR) is a good gauge of the overall end-to-end quality of the cable network, because it includes the impact that the transmitter circuitry, receiver circuitry, and transmission media have on the upstream signal.
- Carrier-to-Noise Ratio (CNR)—This is an ratio of the measured modulated power, in dB, on the upstream (before ingress noise cancellation is done) that compares the channel power to the noise power.

  The term CNiR is part of the CableLabs nomenclature for the CNR measurement. Therefore these two terms, CNR and CNiR, can be used interchangeably.

  The CNR (CNiR) measurement is usually provided only by an external spectrum analyzer, but the cable interface line cards that support intelligent and advanced hardware spectrum management features also provide CNR (CNiR) measurement.

  The following two types of CNR (CNiR) measurements are supported on the Cisco CMTS:

  - CNR (CNiR) measured for a particular upstream—This is the overall CNR (CNiR) for all of the cable modems on an upstream, which is determined by measuring the RF power of the upstream receiver at the cable interface. This value is always just a snapshot in time for a particular upstream. The cable interface measures the RF power at a time when no bursts are expected from the cable modems, but it can be skewed by a small number of cable modems that are experiencing or creating signal problems.
  - Per-modem CNR (CNiR)—This is the CNR (CNiR) for a particular cable modem, which is signal strength of the burst transmissions of the modem at the upstream receiver of the cable interface. The per-modem CNR (CNiR) measurement is a very accurate measure of a particular cable modem's signal, but you should not use a single modem's CNR (CNiR) to make assumptions about other cable modems on that upstream or about the upstream itself. However, you can get a good picture of the upstream's signal quality by polling the CNR (CNiR) for a number of cable modems over a representative time period.

Tip Changing the channel width has a direct impact on the CNR (CNiR). Doubling the channel width (for example, from 400 KHz to 800 KHz) decreases the CNR (CNiR) for an upstream by approximately 3 dB. Cutting the channel width in half (for example, from 3.2 MHz to 1.6 MHz) increases the CNR (CNiR) for an upstream by approximately 3 dB.

Differences Between the MER (SNR) and CNR (CNiR) Values

In a perfect network, such as a test lab where the only impairment is additive white Gaussian noise (AWGN), you can expect the CNR (CNiR) and MER (SNR) values to be comparable throughout all of the allowable power levels and frequency ranges. In a live network, however, it is expected that the MER (SNR) value should be a few dB lower than the CNR (CNiR) value, given that the MER (SNR) value takes into account noise impairments and distortions that are not accounted for by the CNR (CNiR) power measurements.

In general, when the CNR (CNiR) value is in the 15 to 25 dB range, you can expect the MER (SNR) value to have a comparable value. The difference between the MER (SNR) and CNR (CNiR) values is expected to be larger when the CNR (CNiR) value falls outside of the 15 to 25 dB range.

The table below provides a comparison between the MER (SNR) and CNR (CNiR) values, listing the major reasons for why the MER (SNR) and CNR (CNiR) values might diverge on an active network that is passing live traffic:

Table 51: Comparison of MER (SNR) and CNR (CNiR) in a DOCSIS Cable Network

Signal-to-Noise (SNR)	Carrier-to-Noise (CNR)
Post-detection measurement of the RF signal.	Pre-detection measurement of the RF signal.
Measurement of the baseband domain.	Measurement of the RF frequency domain.

Signal-to-Noise (SNR)	Carrier-to-Noise (CNR)
Includes the effect of signal distortions and impairments on the signal. These include: • Group delay in the channel such as occurs during operation near the diplexer band edge. • Channel amplitude variation and echoes. • Data collisions. • Microreflections. • Narrow band ingress in the channel. • Non-linearities in the cable plant. • Phase noise. • Poor selection of the preamble. • Poor symbol fidelity in the transmission of a a cable modem, despite a good MER (SNR) value. • Unrecoverable carrier offsets. • Unrecoverable symbol timing offsets.	Measures only the RF modulated carrier power versus noise power.
Provides an indication of overall, end-to-end network quality (what the transmitter, receiver, and transmission media are doing to the signal).	Provides an indication of network performance (what the transmission media or network is doing to the signal).
Average over time with current data traffic patterns, useful for tracking long-term trends in signal quality.	Real-time spectrum analysis.
Reflects the CNR (CNiR) value as part of its value.	Does not reflect the MER (SNR) value as part of its value.
Averaged over 10,000 symbols, and an accurate reading requires that short and long grants are being transferred.	Unaffected by the type of traffic being transmitted.
Does not use packets with uncorrectable FEC errors to determine its value. Bursts of uncorrectable errors, therefore, could result in a deceptively high MER (SNR) value.	Unaffected by uncorrectable FEC packet bursts.
DOCSIS specifications do not define any required MER (SNR) values for upstreams and downstreams.	Minimum downstream CNR of 35 dB in a 6-MHz band (44 dB in DOCSIS 2.0 for 8-MHz band) Minimum upstream CNR (CNiR) of 25 dB.

Additional Measurements

In addition to MER (SNR) and CNR (CNiR) values, you should be aware of and monitor the following indicators of signal quality:

- MER—This is the measure of RF signal quality, in dB, which is equivalent to SNR and similar to CNR (CNiR) under additive white Gaussian noise (AWGN) impairments. However, MER is preferred for data networks, because it also includes additional factors that affect the signal, such as analog-to-digital and digital-to- analog conversions, rounding errors, distortions, and signal impairments such as phase noise, group delay, and jitter. For this reason, the DOCSIS 2.0 RF specification adds a requirement for the minimum MER value for a signal, supplementing the existing CNR (CNiR) minimum requirements.

  A simple formula for calculating the MER value for an upstream is:

  ```
  MER = 20 x log (RMS error magnitude / Average symbol magnitude)
  ```

  You can also calculate the Error Vector Modulation (EVM) to find the equivalent value expressed as a percentage of noise on an upstream:

  ```
  EVM = Average error magnitude / Max symbol magnitude * 100
  ```

  See the DOCSIS 2.0 specification for more complete information on calculating and using the MER value.

- FEC Counters—These are counters that keep track of how many correctable and uncorrectable FEC errors occur on the upstream. The FEC error counters are useful for tracking fast transient errors such as impulse noise that are not usually reflected in MER (SNR) or CNR (CNiR) values.

  A correctable error count of more than 1 percent can be used as a warning sign of possible physical plant or cable modem problems that might be developed. An uncorrectable error count of more than 1 percent can indicate an existing problem that is blocking traffic on the upstream. Cable interface line cards that support the intelligent and advanced spectrum management features can use the FEC counters as one of the indicators to be monitored to determine whether an upstream must change frequencies so as to correct noise problems.

- Microreflections—Additional copies of a signal that arrive at the receiver, usually at different times and attenuated by different amounts, causing the receiver to misidentify the incoming signal's true phase and amplitude. Microreflections typically are caused by impedance mismatches in the physical cable plant, and can indicate either equipment that has been degraded by weather or other causes, or equipment that has not been installed correctly.

Upstream Signal Channel Overview

The upstream channel is characterized by many cable modems transmitting to the CMTS. These signals operate in a burst mode of transmission. Time in the upstream channel is slotted. The CMTS provides time slots and controls the usage for each upstream interval. The CMTS periodically broadcasts Upstream Channel Descriptor (UCD) messages to all cable modems. The UCD message contains the upstream frequency and transmission parameters associated with an upstream channel. These messages define upstream channel characteristics including the upstream frequencies, symbol rates and modulation schemes, forward error correction (FEC) parameters, and other physical layer values.

Cisco supports all DOCSIS error-correction encoding and modulation types and formats. Upstream signals are demodulated using QPSK or QAM. QPSK carries information in the phase of the signal carrier, whereas QAM uses both phase and amplitude to carry information.

Sending data reliably in the upstream direction is an issue. Because upstream spectrum varies greatly between cable plants, select upstream parameters based on your cable plant's return paths. Select or customize upstream

profiles for the maximum trade-off between bandwidth efficiency and upstream channel robustness. For example, QAM-16 requires approximately 7 dB higher CNR (CNiR) to achieve the same bit error rate as QPSK, but it transfers information at twice the rate of QPSK.

Note The above specifications are based on predetermined sets of frequencies that may or may not have an adequate CNR (CNiR) at any given time.

Upstream frequencies can be assigned as follows:

- Fixed—Configuring a spectrum group disables the fixed upstream frequency setting.
- Single subband—The CMTS administrator can define a center frequency and symbol rate such that the boundaries of the upstream carrier stay within the subband. The frequency and symbol rate can change within the boundary in response to noisy line conditions, based on the defined upstream parameters.
- Multiple subbands—The data carrier can remain in a particular subband for a duration of time and then hop to another subband based on the defined upstream parameters.

Tip Measurement of noise power levels with a spectrum analyzer should be part of the procedure in initially selecting and setting up frequency allocations. We recommend having fixed frequency settings during early deployment, at least until amplifier cascade adjustments or plant repair have become infrequent enough that they no longer significantly affect the nodes connected to the upstream port.

Upstream Frequency Changes

As stated in the DOCSIS radio frequency interface (RFI) specification, RF channel migration or upstream frequency change occurs when a change in the UCD message is broadcast to all cable interfaces.

The speed of channel migration via the UCD message is typically less than 20 milliseconds (ms). During this time, upstream transmission is interrupted until the cable interface transmitter adjusts to its new frequency. Data is stored in the cable interface buffers during this time and is sent when the frequency hop is complete.

Station maintenance intervals are used to perform per modem keepalive polling. The CMTS polls each cable modem at least once every 30 seconds, with the default being once every 20 seconds. When ingress noise causes loss of keepalive messages from a configurable percentage of all cable interfaces, resulting in missed polls, a new frequency is selected from the allocation table and a UCD update is performed. The migration time is 2 msec for any upstream UCD update. After the UCD is updated, the hop occurs. The system must wait until a hop threshold time interval has elapsed before it can change the UCD a second time.

Upstream Segments and Combiner Groups

The Cisco routers divide a cable plant into downstream channels. Downstream channels contain upstream segments. Each upstream segment typically serves more than one fiber node. Upstream segments can be defined as one of the following:

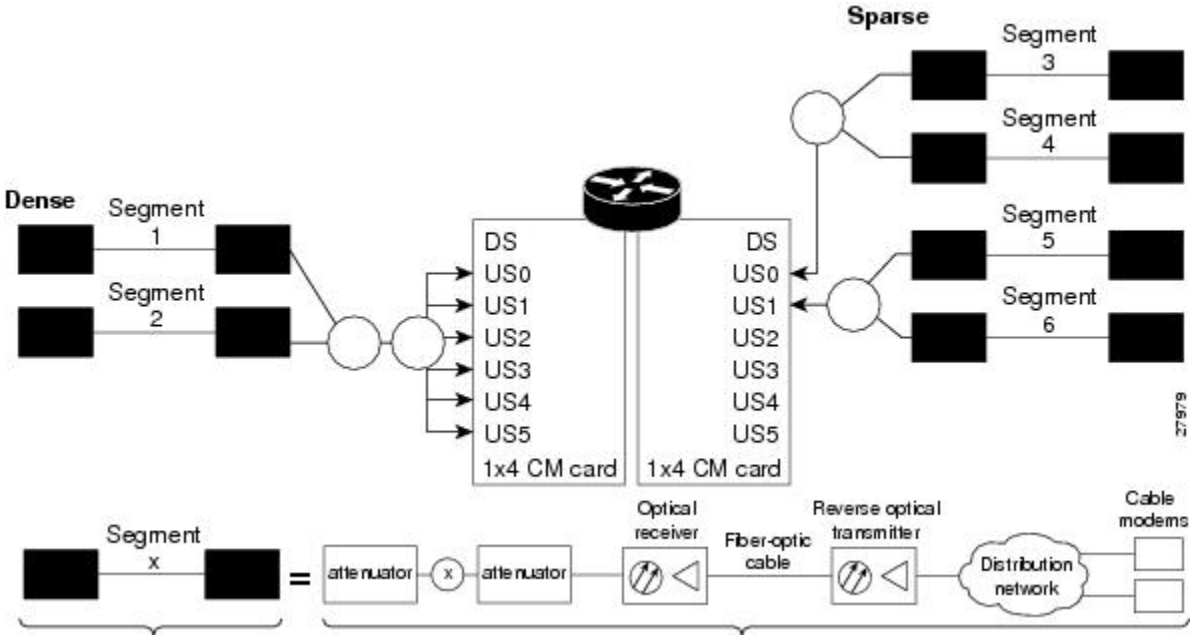
- Sparse segment—Containing one upstream channel per upstream segment.
- Dense segment—Containing multiple upstream channels per upstream segment; frequencies must be different.

Note A cable interface line card can support sparse or dense segments, or both.

Defining sparse segments allows the cable operator to share upstream bandwidth among fiber nodes with fewer subscribers. Defining dense segments allows the cable operator to provide larger upstream bandwidth to fiber nodes with many subscribers.

The figure below illustrates sparse versus dense segments.

Figure 17: Sparse Versus Dense Segment Illustrations



As shown in the figure above, the downstream segment can contain multiple upstream segments. Two fiber nodes can be in one downstream segment but in different upstream segments.

The return path of several fiber nodes can be combined at a single point to form a single RF frequency domain called a combiner group. The CMTS software allows a frequency hop table called a spectrum group to be associated with a combiner group.

Note A combiner group refers to an RF topology point. A spectrum group refers to the frequency hop table associated with a combiner group.

Frequency Management Policy

Spectrum management applies a common frequency-management policy to a set of upstream ports to ensure that data is delivered reliably over the cable plant. Cable plant operators must make noise measurements and determine the cable plant's spectrum management policy. Different modulation schemes, upstream frequency techniques, and symbol rates are used based on the cable plant characteristics and the cable interface line card in the chassis.

See the following sections for more information about these topics:

Noise Impairments

Upstream noise impairments such as signal degradation on cable networks can negatively affect service to subscribers. Two-way digital data signals are more susceptible than one-way signals to stresses in the condition of the HFC network. Degradation in video signal quality might not be noticeable in one-way cable TV service, but when two-way digital signals share the network with video signals, digital signals can be hampered by:

- Impulse and electrical signal ingress—Noise can enter the network from electrical sources within a residence or from high-voltage lines that run near cable television cabling. Two types of ingress noise include broadband and narrowband. Broadband noise is generally of lower frequency (below 10 MHz) and results in harmonic rolloff. Narrowband noise is a more significant interference source. Cable equipment and infrastructure often pick up noise from amateur radio transmissions, citizen band radios, or high-power shortwave broadcast signals. Implement a signal leakage maintenance program to locate and repair areas of signal ingress.
- Amplifier noise—Amplifiers add noise to the HFC network that typically goes unnoticed in video signals, but degrades digital data signals if amplifiers are improperly configured. The larger the network, the higher the probability of amplifier noise affecting signals.
- Noise funneling—The upstream data path to the headend is susceptible to interference from the entire network. All upstream noise ultimately ends up at the headend because the cumulative nature of noise becomes concentrated at the headend. As a network serviced by a single RF receiver increases in size, the probability of noise funneling also increases.
- Variable transmit levels—Temperature affects signal loss over coaxial cable. This can cause variations of 6 to 10 dB per year.
- Clipping—The lasers in fiber-optic transmitters can stop transmitting light when input levels are excessive. Excessive input levels introduce bit errors in both the upstream and downstream transmissions. If a laser is overdriven as briefly as a fraction of a second, clipping can occur.

To adjust your return amplifiers and lasers, follow rigorous plant maintenance procedures documented in the NTSC Supplement on Upstream Transport Issues or appropriate cable plant standard.

Spectrum Groups and Frequency Hopping

We recommend that CMTS administrators configure upstream frequency hopping to counteract long-term, narrowband noise. Cisco CMTS routers support a combination of guided frequency hopping and time-scheduled frequency hopping.

The frequency hop to proactively avoid noise ingress is sometimes called frequency agility. Frequency agility is configured and activated using spectrum groups. Spectrum management supports the creation of a number of cable spectrum groups, allowing multiple upstream ports in a single spectrum group. Each spectrum group defines the table of frequencies to be used in a specific frequency plan. Upstream frequencies can be a fixed single frequency, a single continuous range of frequencies (band), or multiple ranges (or bands) of frequencies.

The cable interface does not operate until you assign a frequency to the upstream, which can be done either by configuring and assigning a spectrum group or assigning a fixed frequency. The spectrum group takes precedence, so if you configure both a spectrum group and a fixed frequency on an upstream, the spectrum group overrides the fixed upstream frequency setting.

From the interface point of view, a spectrum group also represents the set of upstreams connected to the same group of fiber nodes. The spectrum manager software in Cisco routers examines all the RF parameters that

have been configured on an upstream to determine whether the upstream frequencies need to be managed together. For example, if you configure a spectrum group with several fixed frequencies, but those frequencies are all within the configured channel width, the spectrum manager software combines the frequencies into a single band.

The upstream ports use the spectrum group to determine which frequencies are available if frequency hopping is needed to deal with noise or other path impairments. The types of frequency hopping techniques are guided, time-scheduled, and combined guided and time-scheduled. See the Frequency Hopping Capabilities, on page 347 for more information on the types of frequency hopping techniques.

Note When each upstream port has its own RF domain, the group is called a nonshared spectrum group. When multiple upstream ports share the same RF domain, the group is called a shared spectrum group.

Guidelines for Spectrum Management

In general, when defining your spectrum, use the following guidelines:

- Avoid frequencies with known ingress problems, such as amateur radio bands or short-wave bands.
- Avoid a hostile spectrum below 20 MHz.
- Allow extra bands for frequency hopping.
- Take the possible channel widths into account when creating frequency bands. The range of frequencies being used must be able to hop between at least two different frequencies when using the channel width that is configured on the upstream.
- Place upstream ports in the same combiner group in a shared spectrum group.
- Use the receive power level setting to perform slight equalization adjustments.
- If you combine multiple upstream ports to provide increased bandwidth, you must avoid overlapping frequency bands. Each port should be using a discrete band of frequencies that does not overlap the bands being used by other ports in the group. We recommend adding at least 20 KHz between the ending frequency of one band and the starting frequency of the next band, to ensure that the bands do not overlap.

Guided and Scheduled Spectrum Management

Guided and scheduled spectrum management constitutes a set of basic features for all currently supported cable interface line cards. These features are considered basic because they are available for all cable interfaces, and constitute the elementary, cornerstone features upon which the intelligent and advanced spectrum management features are built.

See the following sections for more information about each feature:

Frequency Hopping Capabilities

Noise in the upstream transmission line, that is from the consumer to the service provider, can degrade data transmission from the subscriber's home. If the noise impairment is of substantial duration, it may cause the cable modem to temporarily lose communication with the headend facility. As a contingency plan, the multiple service operators (MSOs) can reserve multiple channels or upstream frequencies for their subscribers. If one channel suffers too much interference, the CMTS requests that the cable modems "hop" to another channel.

To provide frequency hopping capability, Cisco CMTS routers contain a spectrum manager that continuously monitors the noise in unused upstream channels. If the CNR (CNiR) reaches an unacceptable level on a particular channel, the spectrum manager automatically assigns a new upstream channel to the cable modem using that channel.

Cisco CMTS routers support the following techniques for upstream frequency hopping when the frequency band in use is not clean:

- Guided frequency hopping—In guided frequency hopping (also known as blind hopping), the spectrum manager automatically assigns a new upstream channel frequency when a configurable threshold of station maintenance (keepalive) messages fails. Failed station maintenance messages represent an impairment of the upstream channel due to noise, plant, or equipment failure. Explicit frequency subbands and associated input power levels are assigned in a spectrum group in guided frequency hopping.
- Time-scheduled frequency hopping—Frequency reassignment is scheduled by the time of day or by a specific day of the week.
- Combined guided and time-scheduled frequency hopping.

Note Frequency hopping is not effective against broadband noise phenomena such as impulse noise.

Time-scheduled and guided hopping techniques are independent concepts:

- The spectrum is controlled by a script, not a frequency table.
- The available spectrum is time-scheduled as an option.
- A guided hopping frequency is selected from the available spectrum at the current time.

You can configure and activate frequency hopping by using spectrum groups. You can create up to 40 cable spectrum groups, each containing multiple upstream ports. The configured channel width is used for each upstream frequency.

After you have created one or more spectrum groups for your cable network, you can add characteristics to them, providing you with more definitive control over frequency usage and frequency hopping.

You can configure hopping thresholds. For example, the frequency hop threshold percentage method prevents a single failing cable modem from affecting service to other working cable modems. As long as a high enough threshold is configured, the system does not hop endlessly due to a single cable modem failing to respond to 90 percent of its station maintenance (keepalive) messages.

You can also configure the minimum period between frequency hops, with a default setting of 30 seconds. If the destination channel is expected to be impaired, you can reduce the minimum period between frequency hops to a small value, such as 10 seconds. This allows the frequency hop to continue more rapidly until a clear channel is found. If excessive frequency hop is an issue, you can increase the minimum period between hops.

To configure different techniques of frequency hopping, see the Creating and Configuring Spectrum Groups, on page 355.

Note Spectrum management is not supported for one-way (telco return) cable modems, because spectrum management capabilities focus on the upstream path over an HFC network.

Note After the spectrum-band is changed, the spectrum management does not rearrange the frequency for each US channel if the previous frequency belongs to the range of new spectrum-band, which means that the US frequency will not be changed; if the previous frequceny is out of range of new spectrum-band, those US channels will not get frequencies.

Time-Scheduled Frequency Hopping

You can specify upstream channel frequency reassignment based on a configured time of every day or of a specific day of the week. If your cable plant has an upstream noise characteristic on a weekly cycle, use time-scheduled spectrum allocation. With a time-scheduled policy, a single frequency becomes valid at any given time.

Dynamic Upstream Modulation (MER [SNR]-Based)

This section describes the operation of this feature, which is based on evaluating the MER (SNR) of an upstream.

Note A more advanced version of dynamic upstream modulation, which uses the carrier-to-noise ratio (CNR [CNiR]), is supported on the cards that support intelligent and advanced spectrum management.

Feature Overview

Cisco cable interface line cards monitor the MER (SNR) values and the forward error correction (FEC) counters in the active return path of each upstream port. The Dynamic Upstream Modulation feature determines whether upstream channel signal quality can support the modulation scheme configured, and adjusts to the most robust modulation scheme when necessary. When return path conditions improve, this feature returns the upstream channel to the higher modulation scheme that includes the modulation profile.

A modulation profile is a collection of burst profiles that are sent out in a UCD message to configure modem transmit parameters for the upstream. The Dynamic Upstream Modulation feature adjusts the modulation profiles of an upstream channel based on upstream signal quality.

The Dynamic Upstream Modulation feature is configured on interfaces with fixed upstream frequencies or on interfaces with assigned spectrum groups.

The following examples show two different configurations of the Dynamic Upstream Modulation feature, using two and three modulation profiles.

Example Showing Dynamic Upstream Modulation Using Two Modulation Profiles

You can configure the Dynamic Upstream Modulation feature on the Cisco CMTS router using the following primary and secondary modulation profiles:

- The primary modulation profile uses 64-QAM or 16-QAM, which is a more bandwidth-efficient modulation scheme and has a higher throughput than a QPSK profile.
- The secondary modulation profile uses QPSK, which uses a more robust modulation scheme, but is not bandwidth-efficient.

We recommend that the primary profile use 64-QAM or 16-QAM modulation and the secondary use QPSK. However, this is optional as both modulation profiles can either be QPSK or QAM. It is not mandatory for one profile to be QAM and the other QPSK, but modulation profile switchover is tied to the QAM and QPSK thresholds.

Example Showing Dynamic Upstream Modulation Using Three Modulation Profiles

You can configure the Dynamic Upstream Modulation feature on the Cisco CMTS router using the following primary, secondary, and tertiary modulation profiles:

- The primary modulation profile uses 64-QAM, which is a more bandwidth-efficient modulation scheme and has a higher throughput than a 16-QAM profile.
- The secondary modulation profile uses 16-QAM, which is a more bandwidth-efficient modulation scheme and has a higher throughput than a QPSK profile.
- The tertiary modulation profile uses QPSK, which uses a more robust modulation scheme, but is not bandwidth-efficient.

We recommend that the primary profile use 64-QAM modulation, the secondary profile use 16-QAM, and the tertiary profile uses QPSK. However, this is optional as the modulation profiles can either be QPSK or QAM. It is not mandatory that one is QPSK and the other two are QAM, but modulation profile switchover is tied to the QAM and QPSK thresholds.

Criteria for Switching Modulation Profiles

The Dynamic Upstream Modulation feature uses the following criteria to determine whether it should switch from the primary modulation profile (the more bandwidth-efficient, but less robust profile) to the secondary modulation profile (more robust, but less bandwidth-efficient profile) or to the (optional) tertiary modulation profile (most robust, but less bandwidth-efficient profile):

The modulation switch from the primary profile (high performance) to the secondary profile (mid-level performance) uses the following criteria:

- The upstream MER (SNR) is less than or equal to MER (SNR) threshold one and the percentage of correctable FEC (cFEC) errors is greater than or equal to the correctable FEC error threshold or the percentage of uncorrectable FEC (uFEC) errors is greater than or equal to the uncorrectable FEC error threshold.

Before switching back to the primary profile from the secondary profile, the following criteria must be satisfied:

- The upstream MER (SNR) is greater than or equal to the sum of MER (SNR) threshold one and the hysteresis value and the percentage of correctable FEC errors is less than or equal to the correctable FEC error threshold and the percentage of uncorrectable FEC errors is less than or equal to the uncorrectable FEC error threshold and the hop period equals to the default value of 15 seconds.

The modulation switch from the secondary profile (mid-level performance) to the tertiary profile (most robust) uses the following criteria:

- The upstream MER (SNR) is less than or equal to MER (SNR) threshold two and the percentage of correctable FEC (cFEC) errors is greater than or equal to the correctable FEC error threshold or the percentage of uncorrectable FEC (uFEC) errors is greater than or equal to the uncorrectable FEC error threshold.

Before switching back to the secondary profile from the tertiary profile, the following criteria must be satisfied:

- The upstream MER (SNR) is greater than or equal to the sum of MER (SNR) threshold two and the hysteresis value and the percentage of correctable FEC errors is less than or equal to the correctable FEC error threshold and the percentage of uncorrectable FEC errors is less than or equal to the uncorrectable FEC error threshold.

The modulation switch from the primary profile to the tertiary profile uses the following criteria:

- The upstream MER (SNR) is less than or equal to MER (SNR) threshold two and the percentage of correctable FEC (cFEC) errors is greater than or equal to the correctable FEC error threshold or the percentage of uncorrectable FEC (uFEC) errors is greater than or equal to the uncorrectable FEC error threshold.

Before switching back to the primary profile from the tertiary profile, the following criteria must be satisfied:

- The modulation switch from the tertiary profile to the primary profile is a two-step process:
  1. The modulation switch happens from tertiary profile to the primary profile, when the upstream MER (SNR) is greater than or equal to the sum of MER (SNR) threshold one and the hysteresis value.
  2. After a 15-second (non-configurable) delay, the modulation switch occurs from secondary profile to the primary profile, when the upstream MER (SNR) remains greater than or equal to the sum of MER (SNR) threshold one and the hysteresis value.

If the only problem is that the upstream is experiencing a large number of uncorrectable errors, then a situation could occur where the router continues to switch back and forth between profiles. The uncorrectable errors occur with the primary profile, so the router switches to the secondary profile. The secondary profile does not experience any problems, so the router switches back to the primary profile. But the uncorrectable errors reoccur and the router switches back to the secondary profile, and this cycle continues indefinitely.

To avoid this problem, make sure that the cable plant is capable of supporting the modulation scheme being used in the primary profile (for example, 64-QAM). If you cannot guarantee successful operation on an upstream using this modulation scheme, then you should select a primary profile that uses a more bandwidth-efficient set of burst parameters (such as QPSK). The Cisco IOS software includes predefined modulation profiles that can be used for the primary, secondary, and tertiary profiles.

Input Power Levels

The input power level, *power-level-dBmV*, is an option in the **cable spectrum-group** command. The option allows you to specify the expected upstream input power levels on the upstream receivers on the CMTS when the cable modems are hopping from one fixed frequency to another or from one band to another. Each upstream channel width has an associated upstream input power level in dBmV. The power level is the modem transmit power that each spectrum group can use when an upstream frequency change is necessary. The input power level may be set at the time of the frequency hop.

Specifying an input power level is done so that the cable modems do not have to increase or decrease their transmit power with every hop. The cable operator can perform minor power equalizations as a function of frequency. The valid range is –10 to 10dBmV. The power level value should be changed only if you want to change the power level as part of spectrum management. Some cable plants may want to change only the input power level, and not the frequency, on a daily time schedule.

Intelligent and Advanced Hardware-Based Spectrum Management

Several cable interface line cards include hardware-based spectrum management features that provide enhancements to the basic features supported by the other Cisco cable interface line cards.

Intelligent Spectrum Management Enhancements

The following features are part of the intelligent spectrum management feature set:

- Integrates a DOCSIS cable interface line card with an onboard spectrum analyzer that continuously analyzes the upstream spectrum quality in the DOCSIS frequency range of 5 to 42 MHz.
- Includes hardware-assisted frequency hopping, providing for more intelligent and faster frequency selection than software-only solutions.
- Reduces the response time to ingress noise that could cause modems to drop offline.
- Eliminates blind frequency hopping by initiating frequency hops to known clean channels.
- Improves frequency agility to help eliminate dropped packets and thereby maintain full upstream data rates.
- Supports frequency agility in dense-mode combining environments across a shared spectrum.
- Restricts frequency hopping to a set of discrete fixed frequencies or to a range of frequencies, as desired.
- Allows frequency hop conditions to be customized for specific plant environments and requirements.
- Optionally schedules frequency hops to take advantage of known usage patterns or plant conditions.
- Optionally dynamically reduces channel width to allow cable modems to remain online, even in noisy upstream conditions.

Benefits

The spectrum management features provided on the Cisco CMTS router platforms provide several key system benefits:

- Improves response time to ingress noise impairments that appear in the upstream return path.
- Boosts the percentage of modems online.
- Mitigates the impact of ingress to subscriber services.
- Saves time and effort by MSO staff when troubleshooting minor plant outages.
- Increases cable plant reliability.
- Maximizes spectrum utilization.

Guided and Scheduled Spectrum Management Benefits

The following summarizes the specific benefits of the guided and scheduled spectrum management features that are supported for all Cisco CMTS router platforms.

Input Power Levels

Allows the cable plant operator to perform minor power level equalization as a function of frequency.

Frequency Hopping Capabilities

Proactively countermeasures upstream noise impairments by assigning a new upstream channel to the cable modem. MSOs can take advantage of this feature especially when they have less than an optimal carrier-to-noise ratio in the upstream frequencies or when their cable plants exhibit random bursts of ingress noise that affect reliability.

Dynamic Upstream Modulation

- Reduces the risk associated with transitioning to QAM-16 modulation in the return path and provides assurance that subscribers remain online and connected during return path impairments.
- Checks that the active upstream signal quality can support the configured modulation scheme and proactively adjusts to the more robust modulation scheme when necessary.
- Eliminates the necessity to hop channels for cable modems to stay online by automatically switching from the primary modulation profile to the secondary modulation profile.

Intelligent and Advanced Spectrum Management Benefits

The following summarizes the specific benefits of the advanced spectrum management features that are supported on Cisco CMTS routers using supported cable interface line cards.

Dynamic Channel Width Change

- Improves the DOCSIS upstream channel availability by finding the maximum possible channel width for an upstream when noise conditions make the current channel width unusable.
- Provides the maximum RF spectrum utilization efficiency for current plant conditions.
- Customizable range of channel widths that can be used to respond to noise problems.

Intelligent Frequency Hopping

- Proactively changes upstream frequency for an interface before noise conditions become severe enough to force cable modems offline.
- Dedicated hardware intelligent frequency hopping performs “look-ahead” to choose new upstream frequency to find a stable channel.
- Flexible priority configuration allows hopping decision criteria to be tailored to the individual cable plant environment.
- Improves responsiveness to ingress impairments, by matching the hopping decision criteria to the fluctuating plant conditions.
- Pinpoints CNR (CNiR) variations with per-modem accuracy to isolate problematic cable modems.
- Sustains or even improves subscriber online percentages through user-programmable proactive channel management techniques.

Dynamic Upstream Modulation

- Reduces the risk associated with switching between QPSK and QAM-16 modulation in the upstream to respond to ingress noise, so that subscribers remain online and connected.
- Checks the current upstream signal to ensure that it can support the configured modulation scheme, and proactively adjusts to the secondary more robust modulation scheme when necessary.
- Improves DOCSIS upstream channel availability and provides maximum RF spectrum utilization efficiency.
- Eliminates unnecessary frequency hopping by switching modulation profiles to one that allows cable modems to remain online while using the currently assigned upstream.
- Provides assurance that subscribers remain online and connected during periods of return path impairments.

SNMP Interface

- Provides a way to remotely obtain the current status of noise on an upstream. This information can then be inserted into third-party or custom reporting and graphing applications.
- Provides visibility to ingress and impulse noise under the carrier frequency on a per-port basis.
- Provides an easy-to-use, distributed method to remotely gather real-time display of the DOCSIS upstream spectrum for individual cable modems and set-top boxes (STBs).
- Reduces the reliance on costly spectrum analyzers at every headend or hub.
- Quickly provides spectrum views through an intuitive interface, without the complicated setup time of a spectrum analyzer.
- Allows the technician to troubleshoot the network remotely, as opposed to having to be physically present to connect and use a spectrum analyzer.

Default Hop Priority

For Intelligent and Advanced Spectrum Management feature, the default hop priority is as given below:

- Frequency, modulation, and channel width (when using spectrum groups on spectrum cards).
- Modulation, guided frequency hop, and channel width (when using analyzer cards with spectrum groups).
- Modulation only (when not using spectrum groups [fixed frequency]).

How to Configure Spectrum Management

This section describes the configuration tasks that are most commonly performed when using the spectrum management features on the Cisco CMTS platforms. See the following sections for the configuration tasks that are appropriate for your platform and cable interface line cards.

Guided and Scheduled Spectrum Management Configuration Tasks

The following tasks configure the guided and scheduled spectrum management features that are supported on all Cisco CMTS platforms:

Creating and Configuring Spectrum Groups

A spectrum group defines the frequencies that an upstream is allowed to use when frequency hopping is done, as well as other parameters that control the frequency hops. When creating and configuring spectrum groups, you can specify the following parameters:

- Frequencies that are assigned to the group. The cable interface uses these frequencies to determine what frequencies are available to use when frequency hopping is needed. You can specify either a list of fixed frequencies or a band of frequencies, or both. The Cisco CMTS uses the following rules when adding frequencies to a spectrum group:
  - When specifying a fixed frequency, the Cisco CMTS assumes it is a center frequency with a 6.4-MHz channel width to allow that frequency to operate at all possible channel widths. For example, specifying a frequency of 17,700,000 Hz is equivalent to specifying a frequency band from 14,500,000 Hz to 20,900,000 Hz (a band that is 6.4 MHz wide).
  - If you configure multiple fixed frequencies or bands of frequencies that overlap, the spectrum group combines them into one band. For example, if you specify a fixed frequency of 17,700,000 Hz and a band from 15,800,000 Hz to 25,200,000 Hz, the spectrum group is configured with one band from 14,500,000 Hz to 25,200,00 Hz.
  - If you want more control over a spectrum group's frequencies, configure bands of frequencies with the same width as the desired channel width. For example, if you want to use a center frequency of 17,700,000 Hz with a 3.2-MHz channel width, specify a band that ranges from 16,100,000 Hz to 19,300,000 Hz. To ensure you configure non-overlapping bands, separate the bands by a minimum of 20 KHz.
- Upstream input power level—(Optional) Power level, in dBmV, that the upstream should use when hopping to a new frequency. (Some cable plants might want to change only the input power level, and not the frequency, on a daily time schedule.)
- Hop threshold—(Optional) Percentage of cable modems that start missing station maintenance messages before a frequency hop can occur. Configure the hop threshold percentage as needed to prevent a single failing cable interface from affecting service to other good cable interfaces. This ensures that the system does not hop endlessly because one cable modem is generating 90 percent of the errors and 90 percent of the traffic.
- Hop period—(Optional) Minimum time period that must elapse between frequency hops. This allows you to specify a time period long enough to allow an upstream to stabilize before another frequency hop can be performed.
- Scheduled hop time—(Optional) Time of day at which a frequency hop should be scheduled.

Tip Before adding a list of upstream frequencies (or frequency hop tables), start by determining which upstream ports are assigned to a combiner group. Refer to the Example: Determining the Upstream Ports Assigned to a Combiner Group, on page 379 for an example.

To create and configure a spectrum group, use the following procedure.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable spectrum-group** *group-number* **time**[*day hh:mm:ss*] **frequency** *up-freq-Hz* [*power-level-dBmV*] **Example:** `Router(config)# cable spectrum-group 4 time Monday 12:00:00 frequency 40000000`	Creates the spectrum group (if it does not already exist), and adds the specified fixed frequency to the group.
Step 4	**cable spectrum-group** *group-number* **time** [*day hh:mm:ss*] **band** *up-freq-Hz up-freq2-Hz [power-level-dBmV*] **Example:** `Router(config)# cable spectrum-group 4 band 20000000 24000000 13`	Creates the spectrum group (if it does not already exist), and adds the specified band of frequencies to the group. **Note** Repeat Step 3, on page 356 and Step 4, on page 356 as needed for each fixed frequency and frequency band that should be a member of this spectrum group. You must assign at least two fixed frequencies, or a frequency band that contains at least two center frequencies, to a spectrum group before frequency hopping can occur.
Step 5	**cable spectrum-group** *group-number* **hop period** *seconds* **Example:** `Router(config)# cable spectrum-group 4 hop period 60`	Specifies the minimum time, in seconds, between frequency hops. **Note** We recommend a configuration of 30 seconds when using a Cisco uBR-MC5X20S/U/H BPE.
Step 6	**cable spectrum-group** *group-number* **hop threshold** [*percent*] **Example:** `Router(config)# cable spectrum-group 4 hop threshold 25`	Specifies the frequency hop threshold for a spectrum group. • **percent**—(Optional) Frequency hop threshold as a percentage of station maintenance messages that are lost. Valid range is from 1 to 100 percent, with a default of 50 percent.

	Command or Action	Purpose
Step 7	**cable spectrum-group** *group-number* **Example:** `Router(config)# cable spectrum-group 4`	(Optional) Specifies that the upstream ports in a spectrum group should use a unique upstream frequency.
Step 8	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Assigning a Spectrum Group to One or More Upstream Ports

After a spectrum group has been created and configured, you must assign it to one or more upstream ports before the group's frequency spectrum is used for frequency hopping.

To assign a spectrum group to one or all upstream ports on a controller interface, use the following procedure.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**controller upstream-cable** *slot/subslot/port* **Example:** `Router(config)# controller upstream-cable 9/0/15`	Enters controller configuration mode.
Step 4	**us-channel** *us -channel_num* **spectrum-group** *spectrum-group-num* **Example:** `Router(config-controller)# us-channel 0 spectrum-group 1`	Assigns the specified spectrum group as the default group for all upstream on this controller interface.
Step 5	**us-channel** *us -channel_num* **channel-width** *value* **Example:** `Router(config-controller)# us-channel 0 channel-width 1600000`	Configures the channel-width for the specified upstream channel spectrum group.
Step 6	**end** **Example:**	Exits controller interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config-controller)# end`	

What to do next

Tip To verify the spectrum group configuration, use the **show cable spectrum-group** command in privileged EXEC mode.

Configuring Shared Spectrum Groups (Fiber Node Groups) for DOCSIS 3.0

This feature supports shared spectrum groups that cross multiple cable interface line cards on the Cisco CMTS router, and shared spectrum groups within a single cable interface line card.

For additional information about configuring fiber node groups on the Cisco CMTS, see:

- http://www.cisco.com/c/en/us/td/docs/cable/cbr/configuration/guide/b_cbr_layer2_docsis/spectrum_management.html#task_1044164
- http://www.cisco.com/c/en/us/td/docs/cable/cbr/configuration/guide/b_cbr_layer2_docsis/spectrum_management.html#task_4BF4590BA65D4CF1851A4DA0C8A9ADB2

Configuring Dynamic Upstream Modulation (MER [SNR]-Based)

To use the Dynamic Upstream Modulation feature on cable interface line cards that support only the MER (SNR) version of this feature, you must do the following:

1. Create a primary modulation profile. This typically is a more bandwidth-efficient but a less robust profile.
2. Optionally create a secondary modulation profile. This typically is a less bandwidth-efficient but a moderately robust profile.
3. Optionally create a tertiary modulation profile. This typically is a less bandwidth-efficient but a more robust profile.
4. Assign the profiles to the desired cable interfaces and upstreams.

Tip When creating the modulation profiles, we recommend that you use the predefined modulation profiles, as opposed to manually specifying each burst parameter for each modulation profile.

Restriction

- The Dynamic Upstream Modulation feature is supported only for DOCSIS 1.0 or DOCSIS 1.1 TDMA-only modulation profiles for advanced spectrum management.
- The DOCSIS 2.0 mixed-mode or ATDMA-only mode modulation profiles are supported only for basic spectrum management (MER [SNR]-based) and not for advanced spectrum management.
- The Three Step Dynamic Modulation feature supports only basic spectrum management features. It does not support modulation profile changes based on CNR (CNiR) thresholds and CNR (CNiR) measurements.
- The Dynamic Upstream Modulation feature is not enabled for single modulation profile configurations.
- You can configure only two modulation profiles when an upstream is already assigned to a spectrum group for frequency hopping. The spectrum group here implies advanced spectrum management and/or the use of CNR (CNiR).
- A single profile is automatically removed from the configuration if three modulation profiles are assigned to an upstream interface before assigning spectrum group, based on the following conditions:
  - The robust profile is dropped if the upstream port is using a high performance profile.
  - The high performance profile is dropped if the upstream port is using a mid-level or robust profile.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable modulation-profile** *profile* {**mixed** | **tdma** | **atdma** **Example:** `Router(config)# cable modulation-profile 3 mixed`	Creates the primary modulation profile for use on a DOCSIS 1.0 or DOCSIS 1.1 TDMA or A-TDMA upstream. **Note** Repeat this command to create the secondary and tertiary profile for use on a DOCSIS 1.0 or DOCSIS 1.1 TDMA or A-TDMA upstream.

	Command or Action	Purpose
		Note You can also create custom modulation profiles with the **cable modulation-profile** command by configuring the values for the individual burst parameters. These parameters, however, should not be modified unless you are thoroughly familiar with how changing each parameter affects the DOCSIS MAC layer. We recommend using the preconfigured default modulation profiles for most cable plants.
Step 4	**controller upstream-cable** *slot/subslot/port* **Example:** `Router(config)# controller upstream-cable 9/0/15`	Enters controller configuration mode.
Step 5	**us-channel** *n* **modulation-profile** *primary-profile-number* [*secondary-profile-number*] [*tertiary-profile-number*] **Example:** `Router(config-controller)# us-channel 0 modulation-profile 21 121 221`	Assigns a primary modulation profile, and the optional secondary and tertiary modulation profiles, to the specified upstream port.
Step 6	Use one of the following commands: • **us-channel** *n* **threshold snr-profiles** *threshold1-in-db threshold2-in-db* • **us-channel** *n m* **threshold snr-profiles** *threshold1-in-db threshold2-in-db* **Example:** `Router(config-controller)# us-channel 0 threshold snr-profiles 25 15`	(Optional) Specifies the MER (SNR) threshold in dB.
Step 7	Use one of the following commands: • **us-channel** *n* **threshold corr-fec** *corr-fec* • **us-channel** *n m* **threshold corr-fec** *corr-fec* **Example:** `Router(config-controller)# us-channel 0 threshold corr-fec 20`	(Optional) Specifies the allowable number of correctable FEC errors for the upstream.

	Command or Action	Purpose
Step 8	Use one of the following commands: • **us-channel** *n* **threshold uncorr-fec** *uncorr-fec* • **us-channel** *n m* **threshold uncorr-fec** *uncorr-fec* **Example:** `Router(config-controller)# us-channel 0 threshold uncorr-fec 10`	(Optional) Specifies the allowable number of uncorrectable FEC errors for the upstream.
Step 9	**us-channel** *n* **threshold hysteresis** *hysteresis-in-db* **Example:** `Router(config-controller)# us-channel 0 threshold hysteresis 10`	(Optional) Specifies the hysteresis value to be used in conjunction with the dynamic modulation upgrade thresholds.
Step 10	**end** **Example:** `Router(config-controller)# end`	Exits controller interface configuration mode and returns to privileged EXEC mode.

What to do next

Tip See the Dynamic Upstream Modulation (MER [SNR]-Based), on page 349 for a complete description of the Dynamic Upstream Modulation feature.

Verifying Frequency Hopping

You can verify frequency hopping on the CMTS by using the command-line interface (CLI).

Verifying Frequency Hopping Using CLI Commands

To verify frequency hopping using CLI commands, use the following procedure:

Procedure

Step 1 Verify that the interface being tested is up, using the **show interfaces cable** command in privileged EXEC mode. The first line of the output shows whether both the interface and line protocol are up.

Example:

```
Router# show interfaces cable 9/0/0

Hardware is CMTS MD interface, address is c414.3c16.cf8f (bia c414.3c16.cf8f)
MTU 1500 bytes, BW 26000 Kbit/sec, DLY 1000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation MCNS, loopback not set
```

Step 2 Verify that the upstream being tested is up, using the **show interfaces cable upstream** command. The first line shows whether the upstream is up.

Example:

```
Router# show interfaces cable 9/0/0 upstream 0

MAC domain upstream impairment report: 0x0
Cable9/0/0: Upstream 0 is up
Description: UC9/0/0:U0
Received 5 broadcasts, 0 multicasts, 26 unicasts
0 discards, 0 errors, 0 unknown protocol
31 packets input
Codewords: 348 good 0 corrected 0 uncorrectable
```

Step 3 Use the **show cable hop upstream-cable** command to display the frequency that the upstream is currently using:

Example:

```
Router# show cable hop upstream-cable 9/0/0

Upstream    Port       Poll Missed Min    Missed Hop   Hop     Corr    Uncorr
Port        Status     Rate Poll   Poll   Poll   Thres Period  FEC     FEC
                       (ms) Count  Sample Pcnt   Pcnt  (sec)   Errors  Errors
Cable6/0/U5 16.816 Mhz 1000 0      10         0%   20%  25     0       0
```

Step 4 Use the **show cable hop upstream-cable history** command to display the frequency change, modulation change, and channel width change action history of the upstreams:

Example:

```
Router# show cable hop upstream-cable 9/0/0 history

F = Frequency Hop, M = Modulation Change, C = Channel Width Change
Upstream     Action             Chg    Chg    Action
Port         Time          Code From   To     Reason
Ca7/0/0/U0   Sep 17 17:00:24 C  1.6    3.2    Configuration changed
             Sep 14 19:38:55 F  41.117 26.358 Interface state changed
             Sep 14 19:38:55 F   0.000 41.117 Interface state changed
             Sep 14 19:38:24 M  21     221    Configuration changed
```

Step 5 Use the **show cable hop upstream-cable threshold** command to display the user-defined thresholds and current CNR, MER (SNR), correctable FEC percentage, uncorrectable FEC percentage, and missed station maintenance percentage values of the upstreams:

Example:

```
Router# show cable hop upstream-cable 6/0/0 threshold

Upstream   SNR(dB)         CNR(dB)         CorrFEC%  UncorrFEC%       MissedSM%
Port       Val Thre1 Thre2 Val Thre1 Thre2 Pcnt Thre Pcnt Thre        Pcnt Thre

Ca6/0/0/U0 27   25   15    39   35   25    0    3    0    1          75   75
Ca6/0/0/U1 31   25   15    51   35   25    0    3    0    1          90   75
Ca6/0/0/U2 --   35   25    --   35   25    0    3    0    1          0    75
Ca6/0/0/U3 --   35   25    --   35   25    0    3    0    1          0    75
```

Step 6 Use the **test cable hop** command to force the desired upstream to perform a frequency hop. A few seconds after giving the command, a console message should appear informing you of the hop. Repeat the command as needed to verify that the upstream hops through all the frequencies that have been assigned to the upstream's spectrum group.

Example:

```
Router# test cable hop cable 6/0 upstream 5

2w0d: %UBR7200-5-USFREQCHG: Interface Cable6/0 Port U5, frequency changed to 15.760 MHz

Router# test cable hop cable 6/0 upstream 5

2w0d: %UBR7200-5-USFREQCHG: Interface Cable6/0 Port U5, frequency changed to 26.832 MHz
```

Step 7 Use the **test cable channel-width** command to force the desired upstream to perform a channel-width change. A few seconds after giving the test command, use the show cable hop command to verify the channel-width change.

Example:

```
Router# test cable channel-width cable 7/0/0 upstream 0

Channel width changed to 1600000 Hz for Cable7/0/0 U0

Router# *Sep 17 17:06:46.882: %UBR10000-5-USCWCHG: Interface Cable7/0/0 U0, channel width
changed to 1600 kHz SLOT 7/0: Sep 17 17:06:46.898: %UBR10000-5-USCWCHG: Interface Cable7/0/0
 U0, channel width changed to 1600 kHz

Router# Sep 17 17:06:46.898: %Interface Cable7/0/0 U0 With channel width 1600 kHz, the
minislot size is now changed to 4 ticks.

Router# show cable hop cable 7/0/0 upstream 0 history


F = Frequency Hop, M = Modulation Change, C = Channel Width Change
Upstream     Action             Chg    Chg    Action
Port         Time          Code From   To     Reason
Ca7/0/0/U0   Sep 17 17:06:46 C  3.2    1.6    Test command enforced
             Sep 17 17:06:02 M  222    221    SNR 36>=28 CFEC 0<=3 UnCFEC 0<=1
             Sep 17 17:06:00 M  221    222    Test command enforced
             Sep 17 17:03:21 M  222    221    SNR 36>=28 CFEC 0<=3 UnCFEC 0<=1
             Sep 17 17:03:19 M  221    222    Test command enforced
             Sep 17 17:01:44 F  26.358 19.742 Test command enforced
             Sep 17 17:01:17 F  21.528 26.358 Test command enforced
             Sep 17 17:00:24 C  1.6    3.2    Configuration changed
             Sep 14 19:38:55 F  41.117 21.528 Interface state changed
             Sep 14 19:38:55 F   0.000 41.117 Interface state changed
             Sep 14 19:38:24 M  21     221    Configuration changed
Router#
```

Step 8 Use the **test cable freq-hop** command to force the desired upstream to perform a dynamic frequency change. A few seconds after giving the test command, use the **show cable hop** command to verify the frequency change.

Example:

```
Router# test cable freq-hop cable 7/0/0 upstream 0
```

```
SLOT 7/0: Sep 17 17:01:44.650: %UBR10000-5-USFREQCHG: Interface Cable7/0/0 U0, changed to
Freq 19.742 MHz

Router# show cable hop cable 7/0/0 upstream 0 history

F = Frequency Hop, M = Modulation Change, C = Channel Width Change
Upstream     Action             Chg    Chg    Action
Port         Time          Code From   To     Reason
Ca7/0/0/U0   Sep 17 17:01:44 F  26.358 19.742 Test command enforced
             Sep 17 17:00:24 C  1.6    3.2    Configuration changed
             Sep 14 19:38:55 F  41.117 26.358 Interface state changed
             Sep 14 19:38:55 F   0.000 41.117 Interface state changed
             Sep 14 19:38:24 M  21     221    Configuration changed
```

Step 9 Use the **test cable modulation-change** command to force the desired upstream to perform a dynamic modulation change. A few seconds after giving the test command, use the **show cable hop** command to verify the modulation change.

Example:

```
Router# test cable modulation-change cable 7/0/0 upstream 0

SLOT 7/0: Sep 17 17:03:19.038: %UBR10000-5-USMODCHANGE: Interface Cable7/0/0 U0, dynamic
modulation changed to QPSK
SLOT 7/0: Sep 17 17:03:19.038: %UBR10000-6-PREAMLENADJUST: request burst's preamble length
 in mod profile 222 is adjusted to 38 bits.
SLOT 7/0: Sep 17 17:03:19.038: %UBR10000-6-PREAMLENADJUST: initial burst's preamble length
 in mod profile 222 is adjusted to 100 bits.
SLOT 7/0: Sep 17 17:03:19.038: %UBR10000-6-PREAMLENADJUST: station burst's preamble length
 in mod profile 222 is adjusted to 100 bits.

Router# show cable hop cable 7/0/0 upstream 0 history

F = Frequency Hop, M = Modulation Change, C = Channel Width Change
Upstream     Action             Chg    Chg    Action
Port         Time          Code From   To     Reason
Ca7/0/0/U0   Sep 17 17:03:19 M  221    222    Test command enforced
             Sep 17 17:01:44 F  26.358 19.742 Test command enforced
             Sep 17 17:01:17 F  21.528 26.358 Test command enforced
             Sep 17 17:00:24 C  1.6    3.2    Configuration changed
             Sep 14 19:38:55 F  41.117 21.528 Interface state changed
             Sep 14 19:38:55 F   0.000 41.117 Interface state changed
             Sep 14 19:38:24 M  21     221    Configuration changed
```

Troubleshooting Spectrum Group Characteristics

To troubleshoot the configuration, make sure that you entered a valid spectrum group number, time, frequency, and input power level. Also, when defining your spectrum, use the following guidelines:

- Avoid frequencies with known ingress problems, such as amateur radio bands or short-wave bands.
- Avoid a hostile spectrum below 20 MHz.
- Allow extra bands for frequency hopping.
- Place upstream ports in the same combiner group in a shared spectrum group.
- Use the receive power level setting to perform slight equalization adjustments.

Intelligent and Advanced Spectrum Management Configuration Tasks

The following sections describe the configuration tasks that are needed to configure a Cisco uBR7200 series or Cisco uBR10012 universal broadband router for the intelligent and advanced spectrum management features that are available with the Cisco cable interface line cards.

Configuring and Assigning Spectrum Groups

You must create and configure a spectrum group before you can use the intelligent and advanced spectrum management features. These procedures are the same as those used for guided and scheduled spectrum management, which are given in the following sections:

- http://www.cisco.com/c/en/us/td/docs/cable/cbr/configuration/guide/b_cbr_layer2_docsis/spectrum_management.html#task_1044164
- http://www.cisco.com/c/en/us/td/docs/cable/cbr/configuration/guide/b_cbr_layer2_docsis/spectrum_management.html#task_4BF4590BA65D4CF1851A4DA0C8A9ADB2

After the spectrum groups have been configured and assigned to upstreams, the Cisco IOS software automatically uses the advanced frequency hopping algorithms on the cable interface line cards that support it.

Note For efficient use of the intelligent and advanced spectrum management features, we recommend configuring only frequency bands, and not fixed frequencies, when creating spectrum groups. A spectrum group must contain a frequency band that is wide enough for the cable interface to find at least two center frequencies at the configured channel width, before frequency hopping can occur.

Configuring Dynamic Upstream Modulation (CNR-Based)

Configuring the CNR-based version of the Dynamic Upstream Modulation feature is similar to configuring the MER (SNR)-version of this feature:

1. Create a primary modulation profile. This typically is a more bandwidth-efficient but a less robust profile.
2. Create a secondary modulation profile. This typically is a less bandwidth-efficient but a more robust profile.

   Tip When creating the modulation profiles, we recommend that you use the predefined modulation profiles, as opposed to manually specifying each burst parameter for each modulation profile.
3. Assign the profiles to the desired cable interfaces and upstreams.

After the modulation profiles have been created and assigned to upstreams, the Cisco IOS software automatically uses the advanced CNR-based version of the Dynamic Upstream Modulation feature on the cable interface line cards that support it.

Restriction

- The Dynamic Upstream Modulation feature is supported only for DOCSIS 1.0 or DOCSIS 1.1 TDMA-only modulation profiles. It is not supported for DOCSIS 2.0 mixed-mode or A-TDMA-only mode modulation profiles.
- Three Step Dynamic Modulation is not supported on the CNR-based version of dynamic upstream modulation.
- The CNR-based Dynamic Upstream Modulation feature does not support A-TDMA modulation profiles. However, A-TDMA is supported in the MER (SNR)-based Dynamic Upstream Modulation feature.

To assign the primary and secondary profiles to an upstream, use the following procedure.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable modulation-profile** *profile* {**mix** | **qam-16** | **qpsk** | **robust-mix**} **Example:** `Router(config)# cable modulation-profile 3 mix`	Creates the primary modulation profile for use on a DOCSIS 1.0 or DOCSIS 1.1 TDMA upstream. Typically, the primary profile is either **qam-16** or **mix**. **Note** Repeat this command to create the secondary profile for use on a DOCSIS 1.0 or DOCSIS 1.1 TDMA upstream. Typically, the secondary profile is either **robust-mix** or **qpsk**. **Note** You can also create custom modulation profiles with the **cable modulation-profile** command by configuring the values for the individual burst parameters. These parameters, however, should not be modified unless you are thoroughly familiar with how changing each parameter affects the DOCSIS MAC layer. We recommend using the preconfigured default modulation profiles for most cable plants.

	Command or Action	Purpose
Step 4	**controller upstream-cable** *slot/subslot/port* **Example:** `Router(config)# controller upstream-cable 9/0/15`	Enters controller configuration mode.
Step 5	**cable upstream** *n* **modulation-profile** *primary-profile-number secondary-profile-number* **Example:** `Router(config-controller)# cable upstream 0 modulation-profile 3 4`	Assigns a primary modulation profile, and an optional secondary modulation profile, to the specified upstream port.
Step 6	**end** **Example:** `Router(config-controller)# end`	Exits controller interface configuration mode and returns to privileged EXEC mode.

Configuring Proactive Channel Management

The cable interface line cards that support the advanced spectrum management features can be configured with the following parameters to fine-tune the operation of proactive channel management on the upstreams of the cards:

- Priority of the corrective actions to be taken when noise on an upstream exceeds the threshold for its modulation profile.
- CNR (CNiR) and MER (SNR) threshold and FEC values for the upstream and its two modulation profiles.
- Allowable range of channel widths that can be used if frequency hopping or modulation switching cannot avoid the upstream problems.

These parameters all have default settings, so you do not need to perform this procedure unless you want to change these parameters to better match the characteristics of your physical plant.

To configure the parameters, use the following procedure.

Configuring Proactive Channel Management

You can configure two logical channels on a single physical port of the Cisco CMTS router. When you configure logical channels, the upstream related commands are categorized into two groups: physical port level and logical channel level.

Physical Port Level

Physical port level commands use the format of **cable upstream** *n*, where *n* denotes the physical port number.

Logical Channel Level

Logical channel level commands use the format of **cable upstream** *n m* , where *n* denotes the physical port number, and *m* denotes the logical channel index number of 0 or 1.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**controller upstream-cable** *slot/subslot/port* **Example:** `Router(config)# controller upstream-cable 9/0/0`	Enters controller configuration mode.
Step 4	**us-channel** *n* **hop modulation frequency channel-width** **Example:** `Router(config-controller)# us-channel 0 hop modulation frequency channel-width`	Specifies the priority of the three types of corrective actions (**modulation**, **frequency**, and **channel-width**) to be taken when the noise for the upstream exceeds the threshold specified for the current modulation profile. The default priority is **frequency**, **modulation**, and **channel-width**. **Note** The **channel-width** option must always appear after the **frequency** option.
Step 5	**cable upstream** *n* **threshold cnr-profiles** *threshold1-in-db threshold2-in-db* **Example:** `Router(config-controller)# cable upstream 2 threshold cnr-profiles 23 14`	(Optional) Specifies the CNR (CNiR) threshold and FEC values for the upstream and its two modulation profiles. **Note** To bypass both the primary and secondary CNR (CNiR) thresholds, set the first parameter (*threshold1-in-db*) to 0. This disallows the second parameter (*threshold2-in-db*), enabling you to bypass both the CNR (CNiR) thresholds.
Step 6	Use one of the following commands: • **cable upstream n upstream threshold snr-profiles** *threshold1-in-db threshold2-in-db* • • **cable upstream** *n m* **upstream threshold snr-profiles** *threshold1-in-db threshold2-in-db*	(Optional) Specifies the MER (SNR) threshold and FEC values for the upstream and its two modulation profiles. **Note** You can bypass the primary MER (SNR) threshold (*threshold1-in-db*) by setting it to 0. However, you must enter the second parameter (*threshold2-in-db*).

	Command or Action	Purpose
	• **Example:** `Router(config-controller)# cable upstream 2 threshold snr-profiles 23 14`	
Step 7	**cable upstream** *n* **threshold hysteresis** *hysteresis-in-db* **Example:** `Router(config-controller)# cable upstream 2 threshold hysteresis 3`	(Optional) Specifies the hysteresis value to be used in conjunction with the dynamic modulation upgrade thresholds. **Note** You can bypass the **hysteresis** threshold by setting the value to 0.
Step 8	Use one of the following commands: • **cable upstream** *n* **threshold corr-fec** *corrfec-threshold* • **cable upstream** *n m* **threshold corr-fec** *corrfec-threshold* **Example:** `Router(config-controller)# cable upstream 5 threshold corr-fec 5`	(Optional) Specifies the CNR (CNiR) threshold and FEC values for the upstream and its two modulation profiles. **Note** You can bypass the **corr-fec** threshold by setting the value to 0.
Step 9	Use one of the following commands: • **cable upstream** *n* **threshold uncorr-fec** *uncorrfec-threshold* • **cable upstream** *n m* **threshold uncorr-fec** *uncorrfec-threshold* **Example:** `Router(config-controller)# cable upstream 5 threshold uncorr-fec 1`	(Optional) Specifies the CNR (CNiR) threshold and FEC values for the upstream and its two modulation profiles. **Note** You can bypass the un**corr-fec** threshold by setting the value to 0. **Note** For normal plant use, we recommend that the uncorrectable FEC threshold remain at its default of 1 percent to avoid an unacceptable number of errors on the channel.
Step 10	**cable upstream** *n* **channel-width** *first-choice-width* [*last-choice-width*] **Example:** `Router(config-controller)# cable upstream 0 channel-width 800000 800000`	
Step 11	**end** **Example:** `Router(config-controller)# end`	Exits controller interface configuration mode and returns to privileged EXEC mode.

Verifying the Spectrum Management Configuration

Follow the steps given below to verify the spectrum management configuration.

Procedure

Step 1 To check the value of the settings you have entered, use the **show running-config** command in privileged EXEC mode:

Example:

```
Router# show running-config
```

Step 2 To display the configuration for each modulation profile, use the **show cable modulation-profile** command in privileged EXEC mode:

Example:

```
Router# show cable modulation-profile
```

To display the configuration for a specific modulation profile, add the profile number to the **show cable modulation-profile** command in privileged EXEC mode:

Example:

```
Router# show cable modulation-profile 6
```

Step 3 To display the status and configuration of each upstream, use the **show controllers cable upstream** command in privileged EXEC mode. The following example displays information for upstreams 0 on a cable line card:

Example:

```
Router# show controller cable 8/1/14 upstream 0

 Cable8/1/14 Upstream 0 is up
  Frequency 19.504 MHz, Channel Width 3.200 MHz, Symbol Rate 2.560 Msps
  Modulations (64-QAM) - A-short 64-QAM, A-long 64-QAM, A-ugs 64-QAM
  Mapped to shared connector 18 and receiver 56
  Spectrum Group 8
  MC3Gx60  CNR measurement : 30 dB
  US phy MER(SNR)_estimate for good packets - 32.5530 dB
  Nominal Input Power Level 0 dBmV, Tx Timing Offset 1547
  Ranging Backoff Start 3, Ranging Backoff End 6
  US timing offset adjustment type 0, value 0
  Ranging Insertion Interval automatic (60 ms)
  US throttling off
  Tx Backoff Start 3, Tx Backoff End 5
  Modulation Profile Group 221
  Concatenation is enabled
  Fragmentation is enabled
  part_id=0x3142, rev_id=0xC0, rev2_id=0x00
  nb_agc_thr=0x0000, nb_agc_nom=0x0000
  Range Load Reg Size=0x58
  Request Load Reg Size=0x0E
  Minislot Size in number of Timebase Ticks is = 2
  Minislot Size in Symbols = 32
  Bandwidth Requests = 0xEE3AF
  Piggyback Requests = 0x6A24F
  Invalid BW Requests= 0x76
  Minislots Requested= 0xC33362
```

```
Minislots Granted  = 0x158609
Minislot Size in Bytes = 24
Map Advance (Dynamic) : 2581 usecs
Map Count Internal = 330309891
No MAP buffer= 0x0   No Remote MAP buffer= 0x0
Map Counts: Controller 8/1/0 = 1321230158
UCD Counts:
  Controller 8/1/0:0 = 336057
  Controller 8/1/0:1 = 336057
  Controller 8/1/0:2 = 336057
  Controller 8/1/0:3 = 336057

UCD procedures on lch 0
UCD ucd-succeeds(5) ucd-shut(0) init-state-err(0)
UCD init-tss-err(0) init-timeout(0) init-start-err(0)
UCD ucd-ccc-time(0) ucd-timeout(0) ucd-tss-err(0)
UCD ucd-state-err(0) ucd-process(0) ucd-retries(0)
UCD stale-tss(0)
ATDMA mode enabled
PHY: us errors 0  us recoveries 0 (enp 0)
MAC PHY TSS: tss error start 0  tss error end 0
MAC PHY Status: bcm3140 status 0  lookout status 0
PHY: TSS late 0  discontinuous 0
PHY: TSS mis-match 0  not-aligned 0
PHY: TSS missed snapshots from phy 0
MAP/UCD Replication Instructions:
  Controller 8/1/0 index = 477, bitmap = 0x000F
Dynamic Services Stats:
DSA: 0  REQs  0 RSPs  0 ACKs
0 Successful DSAs  0 DSA Failures
DSC: 0  REQs  0 RSPs  0 ACKs
0 Successful DSCs  0 DSC Failures
DSD:  0 REQs  0 RSPs
0 Successful DSDs  0 DSD Failures
Dropped MAC messages: (none)
```

Step 4 To display the hop period and hop threshold values for each upstream, use the **show cable hop** command in privileged EXEC mode:

Example:

```
Router# show cable hop

 Upstream    Port       Poll Missed Min    Missed Hop   Hop     Corr    Uncorr
 Port        Status     Rate Poll   Poll   Poll   Thres Period  FEC     FEC
                        (ms) Count  Sample Pcnt   Pcnt  (sec)   Errors  Errors
Cable3/0/U0 20.800 Mhz 105  0      20         0%   25%  45     1       4
Cable3/0/U1 20.800 Mhz 105  0      48         0%   25%  45     2       19
Cable3/0/U2 23.120 Mhz 105  0      45         0%   25%  45     0       5
Cable3/0/U3 22.832 Mhz 105  0      26         0%   25%  45     0       6
Cable3/0/U4 22.896 Mhz 105  0      43         0%   25%  45     0       7
Cable3/0/U5 23.040 Mhz 105  0      54         0%   25%  45     1       3
Cable4/0/U0 22.896 Mhz 117  0      26         0%   25%  45     0       2
Cable4/0/U1 23.168 Mhz 117  0      87         0%   25%  45     4       2
Cable4/0/U2 22.896 Mhz 117  0      23         0%   25%  45     1       0
Cable4/0/U3 20.800 Mhz 117  0      54         0%   25%  45     0       0
Cable4/0/U4 22.928 Mhz 117  0      22         0%   25%  45     0       1
Cable4/0/U5 22.960 Mhz 117  0      0       -----   25%  45     0       0
```

Step 5 To display changes from one state to another, at any time and for any reason, for frequency, modulation, and channel width, use the **history** option of the **show cable hop** command.

Example:

```
Router# show cable hop c8/1/1 u0 history

F = Frequency Hop, M = Modulation Change, C = Channel Width Change
Upstream   Action             Chg    Chg    Action
Port       Time          Code From   To     Reason
C8/1/1 U0  Feb 20 12:21:29 M  142    141    SNR 28>=28 CFEC 0<=3 UnCFEC 0<=1
           Feb 20 12:09:08 F   0.000 24.000 Configuration changed
```

Step 6 To display thresholds for MER (SNR), CNR (CNiR), and FEC, use the **threshold** option of the **show cable hop** command.

Example:

```
Router# show cable hop c8/1/1 u0 threshold

Upstream   SNR(dB)          CNR(dB)          CorrFEC%  UncorrFEC%       MissedSM%
Port       Val Thre1 Thre2 Val Thre1 Thre2 Pcnt Thre Pcnt Thre        Pcnt Thre
C8/1/1 u0  33   23   14    60   25   15    0    1    0    2           0    50
```

Step 7 To display the assignment of each spectrum group, use the **show cable spectrum-group** command in privileged EXEC mode:

Example:

```
Router# show cable spectrum-group

Group  Frequency      Upstream      Weekly Scheduled          Power   Shared
No.    Band           Port          Availability              Level   Spectrum
       (MHz)                        From Time:    To Time:    (dBmV)
1      42.967 [3.20]  UC2/0/4:U0                               -1     No
1      83.400 [3.20]  UC2/0/4:U1                               -1     No
1      80.200 [3.20]  UC2/0/4:U2                               -1     No
1      42.922 [3.20]  UC2/0/4:U3                               -1     No
1      17.677 [3.20]  UC2/0/5:U0                               -1     54
1      10.603 [3.20]  UC2/0/5:U1                               -1     54
```

In the above example,

- No—Fiber node is not configured
- 54—ID of the fiber node

Step 8 To display the current CNR (CNiR) value for a particular cable modem, use the **show cable modem cnr** command in privileged EXEC mode:

Example:

```
Router# show cable modem 5.100.1.94 cnr

MAC Address    IP Address      I/F         MAC         Prim  snr/cnr
                                           State       Sid   (dB)
0018.689c.17b8 5.100.1.94      C7/0/0/U1   online      428   36.12
```

Monitoring Spectrum Management

You can either use Cisco CLI commands or SNMP to monitor spectrum management activity on the Cisco CMTS.

See the following sections for more information:

Using CLI Commands

The following commands provide information on the spectrum condition of an upstream:

Command	Purpose
`Router# show cable hop[cable x/y] [upstream usport]`	Displays the hop period and hop threshold values, as well as the FEC error counters, for all upstreams in the router, all upstreams on one cable interface line card, or a single upstream.
`Router# show cable hop [cable x/y[z]] [upstream n][thresholds]`	Displays the configured and current value of MER (SNR) in dB, CNR (CNiR) in dB, CorrFEC in percentage, UncorrFEC in percentage, and missed station maintenance in percentage for a specified upstream.
`Router# show cable hop history`	1. With the **show cable hop history** command for entire CMTS, the most recent change of each action is displayed. 2. With the **show cable hop history** command for a MAC domain, the most recent three changes of each action are displayed. 3. With the **show cable hop history** command for a specific upstream, the last ten changes of each action are displayed. Changes are sorted by time with the most recent at the top.
`Router# show cable hop [cable x/y[z]] [upstream n][summary`	Displays hourly, daily, weekly, 30 days running average, and average since the system was brought up for each specified upstream.
`Router# show cable hop [cable x/y[z]] [upstream n] [history]`	Displays changes from one state to another, at any time and for any reason, for frequency, modulation, and channel width.
`Router# show cable modem [ip-address | interface | mac-address] [options]`	Displays information, including MER (SNR) values, for the registered and unregistered cable modems.
`Router# show cable modulation-profile [num] [initial | long | reqdata | request | short | station]`	Displays the configuration for all modulation profiles, for a particular modulation profile, or for a specific burst type for a particular modulation profile.
`Router# show cable spectrum-group[groupnum] [detail]`	Displays information about the spectrum groups that have been configured.

Command	Purpose
`Router# show controllers cable x/y upstream n [ip-address | mac-address] start-freq end-freq res-freq`	Displays the upstream status, including the current frequency, channel width, modulation rate, and spectrum groups.
`Router# show controllers cable x/y upstream n spectrum [ip-address | mac-address] start-freq end-freq res-freq`	Displays the noise levels for a particular cable modem or displays the background noise for an entire upstream.

Note The **show cable flap-list** command displays the flap list of the CMTS router, which provides additional information about whether cable modems on an upstream are experiencing problems, and if so, what type of problems are occurring. For more information about the cable modem flapping and how to monitor the cable modem flap list, see the Flap List Troubleshooting for the Cisco CMTS Routers .

Using SNMP

You can use SNMP to monitor the spectrum management activity. The SNMP manager can be a graphically-based SNMP manager such as CiscoView or the Cable Broadband Troubleshooter (Release 3.0 or later).

The CISCO-CABLE-SPECTRUM-MIB has been enhanced to provide this SNMP support using the following MIB attributes:

ccsSNRRequestTable

The table below lists the attributes in the ccsSNRRequestTable table, which contains the CNR (CNiR) measurements that are made for individual cable modems on an upstream.

Table 52: ccsSNRRequestTable Attributes

Attribute	Type	Description
ccsSNRRequestIndex	Integer32	Arbitrary index to uniquely identify each table entry.
ccsSNRRequestMacAddr	MacAddress	MAC address of the remote online cable modem being reported on.
ccsSNRRequestSNR	Integer32	MER (SNR) value, in dB, that has been measured. This value is 0 when the Operation State is “running.”
ccsSNRRequestOperation	CCSRequestOperation	Sets the current operation: start, pending, running, or cancel.
ccsSNRRequestOperState	CCSRequestOperState	Reports on the current operation state: idle, pending, running, noError, canceled, notOnLine, invalidMac, timeOut, fftBusy, fftFailed, others.
ccsSNRRequestStartTime	TimeStamp	Contains the time when the MER (SNR) measurement operation starts.

Attribute	Type	Description
ccsSNRRequestStoppedTime	TimeStamp	Contains the time when the MER (SNR) measurement stops.
ccsSNRRequestStatus	RowStatus	Controls the modification, creation, and deletion of table entries.

ccsSpectrumRequestTable

The table below lists the attributes for each entry in the ccsSpectrumRequestTable table, which is used to obtain the spectrum profile for a particular cable modem or to obtain the background MER (SNR) for an entire upstream.

Table 53: ccsSpectrumRequestTable Attributes

Attribute	Type	Description
ccsSpectrumRequestIndex	Integer32	Arbitrary index to uniquely identify each table entry.
ccsSpectrumRequestIfIndex	InterfaceIndexOrZero	Interface identifying the upstream.
ccsSpectrumRequestMacAddr	MacAddress	MAC address to specify an MER (SNR) value for a particular cable modem, or 0000.0000.0000 to indicate background noise for the entire spectrum.
ccsSpectrumRequestUpperFreq	CCSFrequency	Upper frequency for the frequency range to be monitored (5000 to 42000 KHz, with a default of 42000 KHz).
ccsSpectrumRequestLowFreq	CCSFrequency	Lower frequency (in KHz) for the frequency range to be monitored (5000 to 42000 KHz, with a default of 5000 KHz).
ccsSpectrumRequestResolution	Integer32	Requested resolution to determine how the frequency range should be sampled (12 to 37000 KHz, with a default of 60 KHz).
ccsSpectrumRequestStartTime	TimeStamp	Time when the spectrum measurement began.
ccsSpectrumRequestStoppedTime	TimeStamp	Time when the spectrum measurement finished.
ccsSpectrumRequestOperation	CCSRequestOperation	Starts a new spectrum management request or cancels the current one.
ccsSpectrumRequestOperState	CCSRequestOperState	Provides the operational state of the current spectrum management request.
ccsSpectrumRequestStatus	RowStatus	Controls the modification, creation, and deletion of table entries.

ccsSpectrumDataTable

The table below lists the attributes in each entry of the ccsSpectrumDataTable table, which contains the results for a spectrum request.

Table 54: ccsSpectrumDataTable Attributes

Attribute	Type	Description
ccsSpectrumDataFreq	CCSMeasuredFrequency	Frequency in KHz for which this power measurement was made.
ccsSpectrumDataPower	INTEGER	Measured received power for the given frequency (–50 to 50 dBmV).

Note The ccsSpectrumRequestTable and ccsSpectrumDataTable tables provide the same information as that provided by the **show controllers cable upstream spectrum** command.

ccsUpSpecMgmtTable

The table below lists the attributes in the ccsUpSpecMgmtTable table, which provides an entry describing each frequency hop.

Table 55: ccsUpSpecMgmtEntry Attributes

Attribute	Type	Description
ccsUpSpecMgmtHopPriority	INTEGER	Specifies the priority of frequency, modulation profile, and channel width in determining corrective action for excessive noise on the upstream (default is frequency, modulation profile, and channel width).
ccsUpSpecMgmtSnrThres1	Integer32	Specifies the upper MER (SNR) threshold for modulation profile 1 (5 to 35 dB, default of 25).
ccsUpSpecMgmtSnrThres2	Integer32	Specifies the upper MER (SNR) threshold for modulation profile 2 (5 to 35 dB, default of 13, and must be lower than that specified for ccsUpSpecMgmtSnrThres1).
ccsUpSpecMgmtFecCorrectThres1	Integer32	Specifies the FEC correctable error threshold for modulation profile 1 (1 to 20 percent)
ccsUpSpecMgmtFecCorrectThres2	Integer32	Deprecated and no longer used.

Attribute	Type	Description
ccsUpSpecMgmtFecUnCorrecThres1	Integer32	Specifies the FEC uncorrectable error threshold for modulation profile 1 (1 to 20 percent).
ccsUpSpecMgmtFecUnCorrecThres2	Integer32	Deprecated and no longer used.
ccsUpSpecMgmtSnrPollPeriod	Integer32	Deprecated and no longer used.
ccsUpSpecMgmtHopCondition	INTEGER	Reports the condition that triggers a frequency hop (MER [SNR] value or percentage of modems going offline).
ccsUpSpecMgmtFromCenterFreq	CCSFrequency	Provides the center frequency (in KHz) before the latest frequency hop.
ccsUpSpecMgmtToCenterFreq	CCSFrequency	Provides the current center frequency (in KHz) after the latest frequency hop.
ccsUpSpecMgmtFromBandWidth	CCSFrequency	Provides the channel width (in KHz) before the latest frequency hop.
ccsUpSpecMgmtToBandWidth	CCSFrequency	Provides the current channel width (in KHz) after the latest frequency hop.
ccsUpSpecMgmtFromModProfile	Integer32	Provides the modulation profile number before the latest frequency hop.
ccsUpSpecMgmtToModProfile	Integer32	Provides the current modulation profile number after the latest frequency hop.
ccsUpSpecMgmtSNR	Integer32	Provides the current MER (SNR) value (in dB) for the upstream.
ccsUpSpecMgmtCnrThres1	Integer32	Specifies the upper CNR (CNiR) threshold for modulation profile 1 (5 to 35 dB, default of 25).
ccsUpSpecMgmtCnrThres2	Integer32	Specifies the upper CNR (CNiR) threshold for modulation profile 2 (5 to 35 dB, default of 13, and must be lower than that specified for ccsUpSpecMgmtCnrThres1).
ccsUpSpecMgmtCNR	Integer32	Provides the current CNR (CNiR) value (in dB) for the upstream.
ccsUpSpecMgmtMissedMaintMsgThres	Integer32	Provides the frequency hop threshold, as a percentage of station maintenance messages that are lost for a spectrum group.

Attribute	Type	Description
ccsUpSpecMgmtHopPeriod	Integer32	Provide the minimum time, in seconds, between frequency hops.

ccsHoppingNotification

The table below describes the attributes contained in the notification that is sent after each frequency hop.

Table 56: ccsHoppingNotification Attributes

Attribute	Type	Description
ccsUpSpecMgmtHopCondition	INTEGER	Reports the condition that triggers a frequency hop (MER [SNR] value or percentage of modems going offline).
ccsUpSpecMgmtFromCenterFreq	CCSFrequency	Provides the center frequency (in KHz) before the latest frequency hop.
ccsUpSpecMgmtToCenterFreq	CCSFrequency	Provides the current center frequency (in KHz) after the latest frequency hop.
ccsUpSpecMgmtFromBandWidth	CCSFrequency	Provides the channel width (in KHz) before the latest frequency hop.
ccsUpSpecMgmtToBandWidth	CCSFrequency	Provides the current channel width (in KHz) after the latest frequency hop.
ccsUpSpecMgmtFromModProfile	Integer32	Provides the modulation profile number before the latest frequency hop.
ccsUpSpecMgmtToModProfile	Integer32	Provides the current modulation profile number after the latest frequency hop.

Configuration Examples

This section provides the following configuration examples:

Spectrum Group and Combiner Group Examples

The following examples help you to determine whether spectrum group and combiner groups are configured and activated.

Example: Verifying Spectrum Group Creation

To verify that a spectrum group has been created, enter the **show cable spectrum-group** command:

```
Router# show cable spectrum-group
spectrum-group 1
spectrum-group 2
spectrum-group 3
```

Example: Time-Scheduled Spectrum Group

If your cable plant has an upstream noise characteristic on a weekly cycle, use time-scheduled spectrum allocation.

```
Router(config)# cable spectrum-group 1 time Mon 08:00:00 frequency 21600000
```

Deletion is performed using the **delete** keyword:

```
Router(config)# cable spectrum-group 1 time Mon 18:00:00 delete frequency 21600000
```

Example: Verifying Spectrum Group Configuration

To verify if spectrum groups have been configured and activated, enter the **show cable spectrum-group** command. This command displays each spectrum group, the frequencies assigned to it, the upstream port to which it has been assigned, whether a schedule exists for it, the currently measured power level, and whether it is a shared spectrum group.

```
Router# show cable spectrum-group

22:07:46: %SYS-5-CONFIG_I: Configured from console by console
Group  Frequency      Upstream     Weekly Scheduled            Power   Shared
No.    Band           Port         Availability                Level   Spectrum
       (Mhz)                       From Time:    To Time:      (dBmV)
1       5.000-15.000                                           0       Yes
1      12.000                                                  0       Yes
1      22.000         Cable6/0 U5                             7       Yes
2      29.000         Cable6/0 U4                             6       No
2      26.000                                                 0       No
3      35.000-41.000                                          0       No
3      16.000-19.000  Cable6/0 U3                             5       No
5*      5.000-10.000               Thu 21:50:00  Thu 21:45:00  0       Yes
```

Example: Determining the Upstream Ports Assigned to a Combiner Group

Following is a sample topology for a CMTS with combiner groups designated A through J. Combiner groups C and E have multiple upstream ports that should be configured in a shared spectrum group. The other upstreams should be configured in a nonshared spectrum group.

In this example, ten combiner groups are served with frequency hop tables from three spectrum groups:

```
Cable3/0
DS +-----+ Upconverter +----- laser group 1
U0 +----- combiner group A
U1 +----- combiner group B
U2 +------combiner group C
U3 +------combiner group C
U4 +----- combiner group D
U5 +------combiner group E
Cable4/0
DS +-----+ Upconverter +----- laser group 2
U0 +------combiner group E
U1 +----- combiner group F
U2 +----- combiner group G
U3 +----- combiner group H
U4 +----- combiner group I
U5 +----- combiner group J
```

The *laser group* term refers to the set of fiber nodes that share the same downstream signal. An optical splitter is often used to create individual feeds per node.

In the downstream direction, two 6-MHz channel slots are assigned. All fiber nodes in combiner groups A through E should have a channel slot containing the downstream signal from Cable3/0. Combiner groups A through E are said to belong to laser group 1.

All fiber nodes in combiner groups E through J should have a channel slot containing the downstream signal from Cable4/0. Combiner groups E through J are said to belong to laser group 2.

Because combiner group E belongs to two laser groups, there should be two different downstream channel slots for Cable3/0 and Cable4/0.

Example: Combiner Group

The following example enables spectrum management for all upstream ports, where all combiner groups use the frequency band from 20 to 26 MHz:

```
CMTS01(config)# cable spectrum-group 1 band 20000000 26000000
CMTS01(config)# cable spectrum-group 2 shared
CMTS01(config)# cable spectrum-group 2 band 20000000 26000000
CMTS01(config)# cable spectrum-group 3 shared
CMTS01(config)# cable spectrum-group 3 band 20000000 26000000
CMTS01(config)# controller upstream-Cable 9/0/0
CMTS01(config-controller)# cable spectrum-group 1
CMTS01(config-controller)# cable upstream 2 spectrum-group 2
CMTS01(config-controller)# cable upstream 3 spectrum-group 2
CMTS01(config-controller)# cable upstream 5 spectrum-group 3
CMTS01(config-controller)# exit
CMTS01(config)# controller upstream-Cable 9/0/1
CMTS01(config-controller)# cable spectrum-group 1
CMTS01(config-controller)# cable upstream 0 spectrum-group 3
```

A description of the spectrum groups 1 through 3 follows:

- Spectrum group 1—This group is nonshared. Upstream RF domains exist for each member upstream port.

```
Upstream Port   RF Domain
Cable3/0 U0     combiner group A
Cable3/0 U1     combiner group B
Cable3/0 U4     combiner group D
Cable4/0 U1     combiner group F
Cable4/0 U2     combiner group G
Cable4/0 U3     combiner group H
Cable4/0 U4     combiner group I
Cable4/0 U5     combiner group J
```

- Spectrum group 2—This group is shared. A single upstream RF domain exists.

```
Upstream Port   RF Domain
Cable3/0 U2     combiner group C
Cable3/0 U3     combiner group C
```

- Spectrum group 3—This group is shared. A single upstream RF domain exists.

```
Upstream Port   RF Domain
Cable3/0 U5     combiner group E
Cable4/0 U0     combiner group E
```

For the 20- to 26-MHz band of each RF domain, the spectrum is channelized according to the channel width settings of each member port. For example, if the ports U2 and U3 of Cable3/0 are set to 3.2 MHz and 1.6 MHz channel widths, respectively, then spectrum group 2 uses the following channelization:

```
> Channel Width Start Stop Center
>           (Mhz) (Mhz) (Mhz) (Mhz)
> 1        3.2    20.0 23.2 21.6
> 2*       1.6    20.0 21.6 20.8
> 3*       1.6    21.6 23.2 22.4
> 4        1.6    23.2 24.8 24.0
```

Note Channels 2 and 3 are not available when channel 1 is in use.

Because the group is shared, ports U2 and U3 will be assigned channels 1 and 4, respectively, to prevent overlap.

Note There are no alternate frequency assignments for either port, and bandwidth is wasted from 24.8 to 26.0 MHz. To create alternate channels, increase the upper boundary from 26.0 to 28.0 MHz.

```
> Channel  Width  Start Stop  Center
>           (Mhz)  (Mhz) (Mhz) (Mhz)
>  1        3.2    20.0  23.2  21.6
>  2        3.2    23.2  26.4  24.8
>  3        1.6    20.0  21.6  20.8
>  4        1.6    21.6  23.2  22.4
>  5        1.6    23.2  24.8  24.0
>  6        1.6    24.8  26.4  25.6
>  7        1.6    26.4  28.0  27.4
```

Try to reduce the spectrum allocation when it is used with small channel widths. Otherwise, there will be a large number of upstream channel slots, and the frequency hopping may require several minutes to find a clean slot.

Example: Other Spectrum Management Configurations

To configure differing spectrum groups, refer to the following examples:

- Use the following example to configure spectrum group 3 with an upstream band of 12,000,000 to 18,000,000 Hz and default power level of 0 dBmV:

  ```
  Router(config)# cable spectrum-group 3 band 12000000 18000000
  ```

- Use the following example to add the upstream band 20,000,000 to 24,000,000 Hz to the list of valid bands with a change in the power level of 13 dBmV for spectrum group 3:

  ```
  Router(config)# cable spectrum-group 3 band 20000000 24000000 13
  ```

- Use the following example to configure a continuous band between 5,000,004 and 40,000,000 Hz for scheduled spectrum group 4 with a default power level of 0 dBmV. The band is available to the spectrum group starting at 12:00 p.m. local time each Monday:

```
Router(config)# cable spectrum-group 4 time Monday 12:00:00 band 5000004 40000000
```

- Use the following example to add the upstream frequency 9,500,000 Hz to the list of valid frequencies and change the nominal power level to 5 dBmV. The spectrum manager adjusts frequencies and power levels on this group at 2:00 a.m. local time each day:

```
Router(config)# cable spectrum-group 3 time 02:00:00 frequency 9500000 5
```

- Use the following example to configure the minimum period before which a frequency hop can occur in seconds:

```
Router(config)# cable spectrum-group 3 hop period 800
```

- Use the following example to configure the threshold value (expressed as a percentage) of the number of "offline" modems identified before the router initiates an automatic frequency hop:

```
Router(config)# cable spectrum-group 3 hop threshold 40
```

- Use the following example to configure a particular spectrum group as a shared RF spectrum group. Specifying a given spectrum group as "shared" tells the router that you want to be sure that upstream frequencies assigned to upstream ports are not assigned to additional upstream ports:

```
Router(config)# cable spectrum-group 3 shared
```

- Use the following example to remove a specified spectrum group from your configuration:

```
Router(config)# no cable spectrum-group 3
```

Dynamic Upstream Modulation Examples

The following examples describe how to display modulation profile information with the **show cable modulation-profile** command and to define a modulation profile with the **cable modulation-profile** command.

Verifying Your Settings

Procedure

Step 1 To check the value of the settings you have entered, enter the **show running-config** command in privileged EXEC mode:

Example:

```
Router# show running-config
```

To review changes you make to the configuration, use the **show startup-config** command in privileged EXEC mode to display the information stored in NVRAM.

Step 2 To display modulation profile group information, use the **show cable modulation-profile** command in privileged EXEC mode:

Example:

```
Router# show cable modulation-profile[profile][iuc-code]
```

This command uses the following syntax:

- *profile*—(Optional) Profile number. Valid values are from 1 to 8.
- *iuc-code*—(Optional) Internal usage code.

  Valid options are:

  - **initial**—Initial ranging burst
  - **long**—Long grant burst
  - **request**—Request burst
  - **short**—Short grant burst
  - **station**—Station ranging burst

Example: Modulation Profiles

The Cisco CMTS has one preconfigured modulation profile resident in memory, which defines a typical profile for QPSK modulation. To use the Dynamic Upstream Modulation feature, a second profile must be created that is unique from the first profile, and typically provides a higher, more robust modulation scheme.

The following example is a modulation profile for QAM-16, in which the initial, request, and station maintenance messages are sent as QPSK, and the short and long data packets are sent as QAM-16. The QAM-16 modulation is more bandwidth-efficient than QPSK, but QPSK is more robust than QAM-16.

Note The upstream request and station maintenance messages use less time on the cable network when configured in QPSK for symbol rates of 640K, 1280K, and 2560K symbols/sec. Thus, these messages are actually more efficient when used in QPSK mode and they ensure a more reliable modem connection. The upstream initial maintenance message takes exactly the same amount of time on the cable network, no matter how it is configured. Modems connect more quickly and experience fewer cycles of power adjustment during initial maintenance if the system is set for QPSK.

```
Router# configure terminal
Router(config)# cable modulation-profile 2 request 0 16 1 8 qpsk scrambler 152 no-diff 64
fixed uw16
Router(config)# cable modulation-profile 2 initial 5 34 0 48 qpsk scrambler 152 no-diff 128
 fixed uw16
Router(config)# cable modulation-profile 2 station 5 34 0 48 qpsk scrambler 152 no-diff 128
 fixed uw16
Router(config)# cable modulation-profile 2 short 6 75 6 8 16qam scrambler 152 no-diff 72
fixed uw16
Router(config)# cable modulation-profile 2 long 8 220 0 8 16qam scrambler 152 no-diff 160
fixed uw16
```

In the following example, all message types are carried with QAM-16 modulation. Although QAM-16 modulation offers a consistent modulation scheme for all five types of messages, the added length of the

QAM-16 preamble offsets the increased bandwidth efficiency of the MAC data message for the station maintenance messages and bandwidth request messages.

```
Router# configure terminal
Router(config)# cable modulation-profile 2 request 0 16 1 8 16qam scrambler 152 no-diff 128
 fixed uw16
Router(config)# cable modulation-profile 2 initial 5 34 0 48 16qam scrambler 152 no-diff
256 fixed uw16
Router(config)# cable modulation-profile 2 station 5 34 0 48 16qam scrambler 152 no-diff
256 fixed uw16
Router(config)# cable modulation-profile 2 short 5 75 6 8 16qam scrambler 152 no-diff 144
fixed uw16
Router(config)# cable modulation-profile 2 long 8 220 0 8 16qam scrambler 152 no-diff 160
fixed uw16
```

Note When using DOCSIS concatenation with a 16-QAM or mixed symbol rate, configure the CMTS for Unique Word 16 ("uw16") in the preamble for both short and long data burst profiles.

Add the **cable upstream** *port-number* **modulation-profile** *primary profile-number secondary profile-number* command to the appropriate interfaces. In this example, modulation profile 2 is for QAM-16 modulation and profile 1 is for QPSK modulation.

```
Router# configure terminal
Router(config)# controller upstream-Cable 6/0/0
Router(config-controller)# cable upstream 0 modulation-profile 2 1
```

Example: Input Power Level

In the following example, the modem transmit power at 24.8 MHz is adjusted upstream by 1 dBmV and the modem transmit power at 28.0 MHz is adjusted upstream by 2 dBmV.

```
CMTS01(config)# cable spectrum-group 1 frequency 21600000
CMTS01(config)# cable spectrum-group 1 frequency 24800000 1
CMTS01(config)# cable spectrum-group 1 frequency 28000000 2
```

Advanced Spectrum Management Configuration Examples

This section provides the following typical configurations:

Example: Advanced Spectrum Management for the Cisco cBR Series Routers

This section provides an excerpt from a typical configuration example for a Cisco cBR Series router using a cable interface line card. This configuration does the following:

- Configures four spectrum groups with a hop period of 30 seconds.
- Creates a QPSK modulation profile and assigns it to four upstreams on the Cisco cable interface line card in slot 6/1/0.
- Assigns a spectrum group to each of the four upstreams.
- Configures each upstream for the default CNR (CNiR) and FEC thresholds.

```
cable modulation-profile 21 qpsk
interface Cable6/1/0
cable bundle 1
cable downstream annex B
cable downstream modulation 256qam
cable downstream interleave-depth 32
! upstream 0
cable upstream 0 spectrum-group 1
cable upstream 0 modulation-profile 21
cable upstream 0 threshold cnr-profiles 16 0
cable upstream 0 threshold Corr-Fec 3
cable upstream 0 threshold Uncorr-Fec 1
no cable upstream 0 shutdown ! upstream 1
cable upstream 1 spectrum-group 2
cable upstream 1 modulation-profile 21
cable upstream 1 threshold cnr-profiles 16 0
cable upstream 1 threshold Corr-Fec 3
cable upstream 1 threshold Uncorr-Fec 1
no cable upstream 1 shutdown ! upstream 2
cable upstream 2 spectrum-group 3
cable upstream 2 modulation-profile 21
cable upstream 2 threshold cnr-profiles 16 0
cable upstream 2 threshold Corr-Fec 3
cable upstream 2 threshold Uncorr-Fec 1
no cable upstream 2 shutdown ! upstream 3
cable upstream 3 spectrum-group 4
cable upstream 3 modulation-profile 21
cable upstream 3 threshold cnr-profiles 16 0
cable upstream 3 threshold Corr-Fec 3
cable upstream 3 threshold Uncorr-Fec 1
no cable upstream 3 shutdown
```

Additional References

The following sections provide references related to Spectrum Management and Advanced Spectrum Management for the Cisco CMTS routers.

Related Documents

Related Topic	Document Title
CMTS Command Reference	Cisco Broadband Cable Command Reference Guide.

Standards and RFCs

Standards	Title
SP-RFIv1.1-I09-020830	Data-over-Cable Service Interface Specifications Radio Frequency Interface Specification, version 1.1
SP-RFIv2.0-I03-021218	Data-over-Cable Service Interface Specifications Radio Frequency Interface Specification, version 2.0
SP-OSSIv2.0-I03-021218	Data-over-Cable Service Interface Specifications Operations Support System Interface Specification, version 2.0

Standards	Title
SP-BPI+-I09-020830	Data-over-Cable Service Interface Specifications Baseline Privacy Plus Interface Specification, version 2.0

MIBs

MIBs	MIBs Link
CISCO-CABLE-SPECTRUM-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

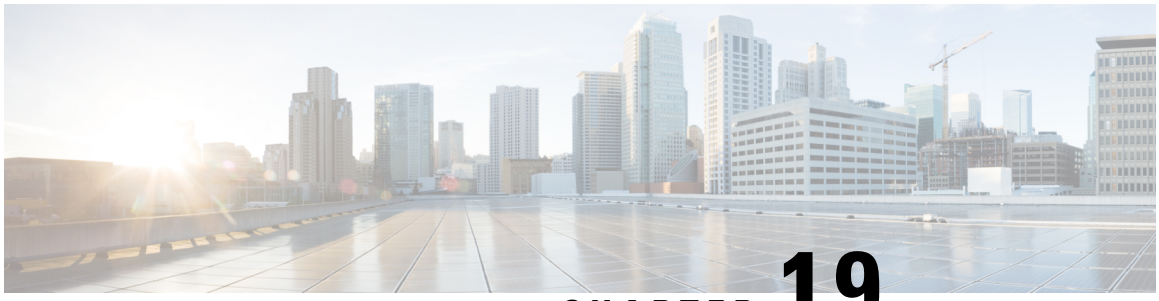
Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Spectrum Management and Advanced Spectrum Management

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Feature Name	Releases	Feature Information
Spectrum Management and Advanced Spectrum Management	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Routers.



CHAPTER 19

Upstream Scheduler Mode

This document describes how to configure optional upstream (US) scheduler modes.

With this feature, you can select Unsolicited Grant Services (UGS), Real Time Polling Service (rtPS) or Non-Real Time Polling Service (nrtPS) scheduling types, as well as packet-based or Time Division Multiplex (TDM) based scheduling. Low latency queuing (LLQ) emulates a packet-mode-like operation over the TDM infrastructure of DOCSIS. As such, the feature provides the typical trade-off between packets and TDM. With LLQ, you have more flexibility in defining service parameters for UGS, rtPS or nrtPS, but with no guarantee (other than statistical distribution) regarding parameters such as delay and jitter.

- Finding Feature Information, on page 387
- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 388
- Restrictions for Upstream Scheduler Mode, on page 388
- Information About Upstream Scheduler Mode for the Cisco CMTS Routers, on page 389
- How to Configure Upstream Scheduler Modes, on page 389
- Additional References, on page 390
- Feature Information for Upstream Scheduler Mode, on page 391

Finding Feature Information

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 57: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for Upstream Scheduler Mode

- To ensure proper operation, Interface-based Admission Control must be enabled. When the LLQ option is enabled, it is possible for the upstream path to be filled with so many calls that it becomes unusable, making voice quality unacceptable. Interface-based admission control must be used to limit the number of calls to ensure acceptable voice quality, as well as to ensure traffic other than voice traffic.
- Even if Interface-based admission control is not enabled, the default (DOCSIS)scheduling mode blocks traffic after a certain number of calls.
- UGS with Activity Detection (UGS-AD) is not supported by the LLQ scheduler mode but remains supported by the default DOCSIS scheduler mode.

Information About Upstream Scheduler Mode for the Cisco CMTS Routers

With UGS, a service flow is created that enables a cable modem to transmit fixed-size bursts of data at a guaranteed rate and with a guaranteed level of jitter by providing periodic transmission opportunities to the cable modem for fixed-sized frames. This kind of service flow is particularly suitable for VoIP applications.

With rtPS, a service flow is created that provides a periodic opportunity for a cable modem to request permission to transmit data by polling a single cable modem for a bandwidth request, rather than all the cable modems. This satisfies applications that have a requirement for real-time data transmission, and enables the cable modem to transmit data bursts of varying length. This kind of service flow is particularly suitable for MPEG VoIP.

The rtPS requests, by default, are internally treated as priority 7—the highest priority for all Best Effort traffic. This high priority reduces the latency of rtPS traffic under congestion.

With nrtPS, a service flow is created that provides a periodic opportunity for a cable modem to request permission to transmit data by polling a single cable modem for a bandwidth request, rather than all the cable modems. The data bursts may be of varying length. This kind of service flow is particularly suitable for non-interactive services such as file transfers.

How to Configure Upstream Scheduler Modes

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	Use one the following commands: • **interface cable** *slot/subslot/port* • **interface cable** *slot/port* **Example:** `Router(config)# interface cable 7/0/1`	Enters interface configuration mode for the specified cable interface.
Step 4	**cable upstream** *n* **scheduling type ugs mode** [**llq** | **docsis**] **Example:**	Enables LLQ-type (packet-based) scheduling for UGS services.

	Command or Action	Purpose
	`Router(config-if)# cable upstream 4 scheduling type ugs mode llq`	**Note** Any combination of **ugs**, **rtps**, **nrtps**, **llq**, and **docsis** is allowed. The only default value is **docsis** .
Step 5	**cable upstream** *n* **scheduling type rtps mode** [**llq** | **docsis**] **Example:** `Router(config-if)# cable upstream 4 scheduling type rtps mode docsis`	Enables standard DOCSIS (TDM-based) scheduling for rtPS services. **Note** Any combination of **ugs**, **rtps**, **nrtps**, **llq**, and **docsis** is allowed. The only default value is **docsis** .
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

What to do next

To confirm whether the scheduler is operating in DOCSIS mode, use the **show interface cable mac-scheduler** command.

```
Router# show interface cable 7/0/1 mac-scheduler 0
     DOCSIS 1.1 MAC scheduler for Cable7/0/1/U0 : rate 30720000
     wfq:None
     us_balance:OFF
     fairness:OFF
     Queue[Rng Polls] flows 0
     Queue[CIR Grants] flows 0
     Queue[BE(07) Grants] flows 0
     Queue[BE(06) Grants] flows 0
     Queue[BE(05) Grants] flows 0
     Queue[BE(04) Grants] flows 0
     Queue[BE(03) Grants] flows 0
     Queue[BE(02) Grants] flows 0
     Queue[BE(01) Grants] flows 0
     Queue[BE(00) Grants] flows 0
     Req Slots 2601578997, Req/Data Slots 4484512
     Init Mtn Slots 38265829, Stn Mtn Slots 78753
     Short Grant Slots 0, Long Grant Slots 0
     Adv Phy Short Grant Slots 412, Adv Phy Long Grant Slots 5519087
     Adv Phy UGS Grant Slots 0
     Avg upstream channel utilization : 1%
     Avg percent contention slots : 98%
     Avg percent initial ranging slots : 1%
     Avg percent minislots lost on late MAPs : 0%

     MAP TSS: lch_state 9, init_retries 0
              late_initial_maps 0, late_ucd_maps 0
              mac-phy tss errors 0, missed ccc 0
```

Additional References

The following sections provide references related to the Cisco CMTS routers.

Related Documents

Related Topic	Document Title
Cisco CMTS command reference	*Cisco CMTS Cable Command Reference* http://www.cisco.com/c/en/us/td/docs/cable/cmts/cmd_ref/b_cmts_cable_cmd_ref.

Standards

Standard	Title
DOCSIS	Data-Over-Cable Service Interface Specifications, DOCSIS 2.0, Radio Frequency Interface Specification, CM-SP-RFIv2.0-I08-050408

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

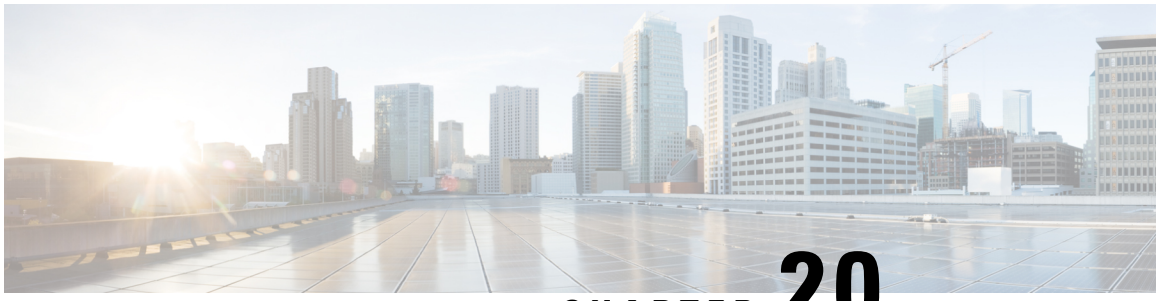
Feature Information for Upstream Scheduler Mode

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 58: Feature Information for Upstream Scheduler Mode

Feature Name	Releases	Feature Information
Upstream Scheduler Mode	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Routers.



CHAPTER 20

Generic Routing Encapsulation

This document describes the Generic Routing Encapsulation (GRE) feature. This feature is a tunneling protocol that enables the encapsulation of a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link to Cisco routers at remote points over an IP internetwork.

- Finding Feature Information, on page 393
- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 394
- Restrictions for Implementing Tunnels, on page 394
- Restrictions for GRE IPv6 Tunnels, on page 395
- Information About Implementing Tunnels, on page 395
- Information About IPv6 over IPv4 GRE Tunnels, on page 397
- Information About GRE IPv6 Tunnels, on page 399
- How to Implement Tunnels, on page 400
- Configuration Examples for Implementing Tunnels, on page 408
- How to Configure IPv6 over IPv4 GRE Tunnels, on page 410
- Configuration Examples for IPv6 over IPv4 GRE Tunnels, on page 412
- How to Configure GRE IPv6 Tunnels, on page 413
- Configuration Examples for GRE IPv6 Tunnels, on page 414
- Additional References, on page 415
- Feature Information for Generic Routing Encapsulation , on page 416

Finding Feature Information

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 59: Hardware Compatibility Matrix for the Cisco cBR Series Routers

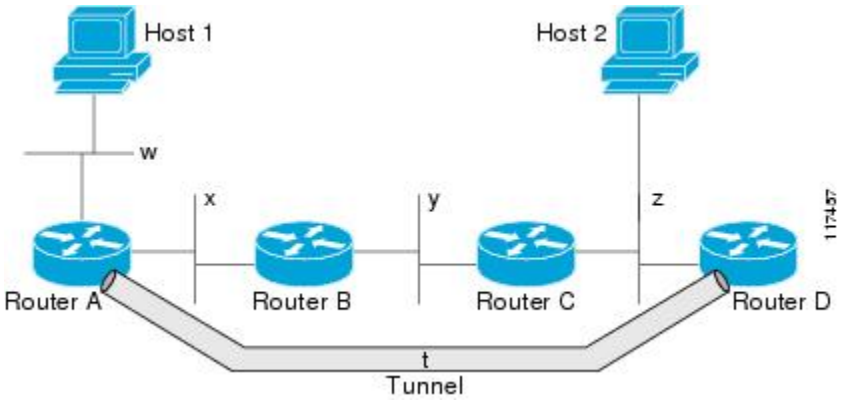
Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for Implementing Tunnels

- It is important to allow the tunnel protocol to pass through a firewall and access control list (ACL) check.
- Multiple point-to-point tunnels can saturate the physical link with routing information if the bandwidth is not configured correctly on a tunnel interface.
- A tunnel looks like a single hop link, and routing protocols may prefer a tunnel over a multihop physical path. The tunnel, despite looking like a single hop link, may traverse a slower path than a multihop link. A tunnel is as robust and fast, or as unreliable and slow, as the links that it actually traverses. Routing protocols that make their decisions based only on hop counts will often prefer a tunnel over a set of physical links. A tunnel might appear to be a one-hop, point-to-point link and have the lowest-cost path, but the tunnel may actually cost more in terms of latency when compared to an alternative physical

topology. For example, in the topology shown in the figure below, packets from Host 1 will appear to travel across networks w, t, and z to get to Host 2 instead of taking the path w, x, y, and z because the tunnel hop count appears shorter. In fact, the packets going through the tunnel will still be traveling across Router A, B, and C, but they must also travel to Router D before coming back to Router C.

Figure 18: Tunnel Precautions: Hop Counts



- A tunnel may have a recursive routing problem if routing is not configured accurately. The best path to a tunnel destination is via the tunnel itself; therefore recursive routing causes the tunnel interface to flap. To avoid recursive routing problems, keep the control-plane routing separate from the tunnel routing by using the following methods:
  - Use a different autonomous system number or tag.
  - Use a different routing protocol.
  - Ensure that static routes are used to override the first hop (watch for routing loops).

  The following error is displayed when there is recursive routing to a tunnel destination:

```
%TUN-RECURDOWN Interface Tunnel 0
temporarily disabled due to recursive routing
```

Restrictions for GRE IPv6 Tunnels

- GRE tunnel keepalive packets are not supported.
- Multipoint GRE (mGRE) IPv6 tunneling is not supported.
- There is limited support for tunnel transport in virtual routing and forwarding (VRF). The limited support in VRF is applicable to IPv6 point-to-point GRE without tunnel protection.

Information About Implementing Tunnels

Tunneling Versus Encapsulation

To understand how tunnels work, you must be able to distinguish between concepts of encapsulation and tunneling. Encapsulation is the process of adding headers to data at each layer of a particular protocol stack.

The Open Systems Interconnection (OSI) reference model describes the functions of a network. To send a data packet from one host (for example, a PC) to another on a network, encapsulation is used to add a header in front of the data packet at each layer of the protocol stack in descending order. The header must contain a data field that indicates the type of data encapsulated at the layer immediately above the current layer. As the packet ascends the protocol stack on the receiving side of the network, each encapsulation header is removed in reverse order.

Tunneling encapsulates data packets from one protocol within a different protocol and transports the packets on a foreign network. Unlike encapsulation, tunneling allows a lower-layer protocol and a same-layer protocol to be carried through the tunnel. A tunnel interface is a virtual (or logical) interface. Tunneling consists of three main components:

- Passenger protocol—The protocol that you are encapsulating. For example, IPv4 and IPv6 protocols.
- Carrier protocol—The protocol that encapsulates. For example, generic routing encapsulation (GRE) and Multiprotocol Label Switching (MPLS).
- Transport protocol--The protocol that carries the encapsulated protocol. The main transport protocol is IP.

Tunnel ToS

Tunnel type of service (ToS) allows you to tunnel network traffic and group all packets in the same ToS byte value. The ToS byte values and Time-to-Live (TTL) hop-count value can be set in the encapsulating IP header of tunnel packets for an IP tunnel interface on a router. Tunnel ToS feature is supported for Cisco Express Forwarding (formerly known as CEF), fast switching, and process switching.

The ToS and TTL byte values are defined in RFC 791. RFC 2474, and RFC 2780 obsolete the use of the ToS byte as defined in RFC 791. RFC 791 specifies that bits 6 and 7 of the ToS byte (the first two least significant bits) are reserved for future use and should be set to 0.

Path MTU Discovery

Path MTU Discovery (PMTUD) can be enabled on a GRE or IP-in-IP tunnel interface. When PMTUD (RFC 1191) is enabled on a tunnel interface, the router performs PMTUD processing for the GRE (or IP-in-IP) tunnel IP packets. The router always performs PMTUD processing on the original data IP packets that enter the tunnel. When PMTUD is enabled, packet fragmentation is not permitted for packets that traverse the tunnel because the Don’t Fragment (DF) bit is set on all the packets. If a packet that enters the tunnel encounters a link with a smaller MTU, the packet is dropped and an Internet Control Message Protocol (ICMP) message is sent back to the sender of the packet. This message indicates that fragmentation was required (but not permitted) and provides the MTU of the link that caused the packet to be dropped.

Note PMTUD on a tunnel interface requires that the tunnel endpoint be able to receive ICMP messages generated by routers in the path of the tunnel. Ensure that ICMP messages can be received before using PMTUD over firewall connections.

Use the **tunnel path-mtu-discovery** command to enable PMTUD for the tunnel packets and use the **show interfaces tunnel** command to verify the tunnel PMTUD parameters. PMTUD works only on GRE and IP-in-IP tunnel interfaces.

QoS Options for Tunnels

A tunnel interface supports various quality of service (QoS) features as a physical interface. QoS provides a way to ensure that mission-critical traffic has an acceptable level of performance. QoS options for tunnels include support for applying generic traffic shaping (GTS) directly on the tunnel interface and support for class-based shaping using the modular QoS CLI (MQC). Tunnel interfaces also support class-based policing, but they do not support committed access rate (CAR).

GRE tunnels allow the router to copy the IP precedence bit values of the ToS byte to the tunnel or the GRE IP header that encapsulates the inner packet. Intermediate routers between the tunnel endpoints can use the IP precedence values to classify packets for QoS features such as policy routing, weighted fair queueing (WFQ), and weighted random early detection (WRED).

When packets are encapsulated by tunnel or encryption headers, QoS features are unable to examine the original packet headers and correctly classify the packets. Packets that travel across the same tunnel have the same tunnel headers, so the packets are treated identically if the physical interface is congested. Tunnel packets can, however, be classified before tunneling and encryption can occur when a user applies the QoS preclassify feature on the tunnel interface or on the crypto map.

Note Class-based WFQ (CBWFQ) inside class-based shaping is not supported on a multipoint interface.

For examples of how to implement some QoS features on a tunnel interface, see the section"Configuring QoS Options on Tunnel Interfaces Examples, on page 409" on page 32.

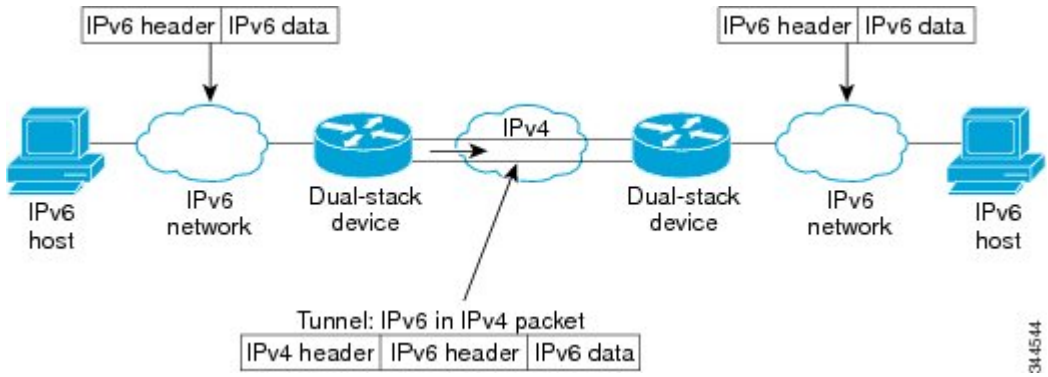
Information About IPv6 over IPv4 GRE Tunnels

Overlay Tunnels for IPv6

Overlay tunneling encapsulates IPv6 packets in IPv4 packets for delivery across an IPv4 infrastructure (a core network or the figure below). By using overlay tunnels, you can communicate with isolated IPv6 networks without upgrading the IPv4 infrastructure between them. Overlay tunnels can be configured between border devices or between a border device and a host; however, both tunnel endpoints must support both the IPv4 and IPv6 protocol stacks. IPv6 supports the following types of overlay tunneling mechanisms:

- Manual
- Generic routing encapsulation (GRE)
- IPv4-compatible
- 6to4
- Intrasite Automatic Tunnel Addressing Protocol (ISATAP)

Figure 19: Overlay Tunnels



Note Overlay tunnels reduce the maximum transmission unit (MTU) of an interface by 20 octets (assuming that the basic IPv4 packet header does not contain optional fields). A network that uses overlay tunnels is difficult to troubleshoot. Therefore, overlay tunnels that connect isolated IPv6 networks should not be considered a final IPv6 network architecture. The use of overlay tunnels should be considered as a transition technique toward a network that supports both the IPv4 and IPv6 protocol stacks or just the IPv6 protocol stack.

Use the table below to help you determine which type of tunnel that you want to configure to carry IPv6 packets over an IPv4 network.

Table 60: Suggested Usage of Tunnel Types to Carry IPv6 Packets over an IPv4 Network

Tunneling Type	Suggested Usage	Usage Notes
Manual	Simple point-to-point tunnels that can be used within a site or between sites.	Can carry IPv6 packets only.
GRE- and IPv4-compatible	Simple point-to-point tunnels that can be used within a site or between sites.	Can carry IPv6, Connectionless Network Service (CLNS), and many other types of packets.
IPv4- compatible	Point-to-multipoint tunnels.	Uses the ::/96 prefix. We do not recommend using this tunnel type.
6to4	Point-to-multipoint tunnels that can be used to connect isolated IPv6 sites.	Sites use addresses from the 2002::/16 prefix.
6RD	IPv6 service is provided to customers over an IPv4 network by using encapsulation of IPv6 in IPv4.	Prefixes can be from the SP's own address block.
ISATAP	Point-to-multipoint tunnels that can be used to connect systems within a site.	Sites can use any IPv6 unicast addresses.

Individual tunnel types are discussed in detail in this document. We recommend that you review and understand the information about the specific tunnel type that you want to implement. When you are familiar with the

type of tunnel you need, see the table below for a summary of the tunnel configuration parameters that you may find useful.

Table 61: Tunnel Configuration Parameters by Tunneling Type

Tunneling Type	Tunnel Configuration Parameter			
Tunnel Mode	**Tunnel Source**	**Tunnel Destination**	**Interface Prefix or Address**	
Manual	ipv6ip	An IPv4 address, or a reference to an interface on which IPv4 is configured.	An IPv4 address.	An IPv6 address.
GRE/IPv4	gre ip		An IPv4 address.	An IPv6 address.
IPv4- compatible	ipv6ip auto-tunnel		Not required. These are all point-to-multipoint tunneling types. The IPv4 destination address is calculated, on a per-packet basis, from the IPv6 destination.	Not required. The interface address is generated as ::*tunnel-source*/96.
6to4	ipv6ip 6to4			An IPv6 address. The prefix must embed the tunnel source IPv4 address.
6RD	ipv6ip 6rd			An IPv6 address.
ISATAP	ipv6ip isatap			An IPv6 prefix in modified eui-64 format. The IPv6 address is generated from the prefix and the tunnel source IPv4 address.

GRE IPv4 Tunnel Support for IPv6 Traffic

IPv6 traffic can be carried over IPv4 GRE tunnels using the standard GRE tunneling technique that is designed to provide the services to implement any standard point-to-point encapsulation scheme. As in IPv6 manually configured tunnels, GRE tunnels are links between two points, with a separate tunnel for each link. The tunnels are not tied to a specific passenger or transport protocol but, in this case, carry IPv6 as the passenger protocol with the GRE as the carrier protocol and IPv4 or IPv6 as the transport protocol.

The primary use of GRE tunnels is for stable connections that require regular secure communication between two edge devices or between an edge device and an end system. The edge devices and the end systems must be dual-stack implementations.

Information About GRE IPv6 Tunnels

Overview of GRE IPv6 Tunnels

The GRE IPv6 Tunnels feature enables the delivery of packets from other protocols through an IPv6 network and allows the routing of IPv6 packets between private networks across public networks with globally routed IPv6 addresses.

For point-to-point GRE tunnels, each tunnel interface requires a tunnel source IPv6 address and a tunnel destination IPv6 address when being configured. All packets are encapsulated with an outer IPv6 header and a GRE header.

How to Implement Tunnels

Determining the Tunnel Type

Before configuring a tunnel, you must determine the type of tunnel you want to create.

Procedure

Step 1 Determine the passenger protocol. A passenger protocol is the protocol that you are encapsulating.

Step 2 Determine the **tunnel mode** command keyword, if appropriate.

The table below shows how to determine the appropriate keyword to be used with the **tunnel mode** command.

Table 62: Determining the tunnel mode Command Keyword

Keyword	Purpose
dvmrp	Use the **dvmrp** keyword to specify that the Distance Vector Multicast Routing Protocol encapsulation will be used.
gre ip	Use the **gre** and **ip** keywords to specify that GRE encapsulation over IP will be used.
gre ipv6	Use the **gre** and **ipv6** keywords to specify that GRE encapsulation over IPv6 will be used.
ipip [**decapsulate-any**]	Use the **ipip** keyword to specify that IP-in-IP encapsulation will be used. The optional **decapsulate-any** keyword terminates any number of IP-in-IP tunnels at one tunnel interface. Note that this tunnel will not carry any outbound traffic; however, any number of remote tunnel endpoints can use a tunnel configured as their destination.
ipv6	Use the **ipv6** keyword to specify that generic packet tunneling in IPv6 will be used.
ipv6ip	Use the **ipv6ip** keyword to specify that IPv6 will be used as the passenger protocol and IPv4 as both the carrier (encapsulation) and transport protocol. When additional keywords are not used, manual IPv6 tunnels are configured. Additional keywords can be used to specify IPv4-compatible, 6to4, or ISATAP tunnels.
mpls	Use the **mpls** keyword to specify that MPLS will be used for configuring traffic engineering (TE) tunnels.

Configuring an IPv4 GRE Tunnel

Perform this task to configure a GRE tunnel. A tunnel interface is used to pass protocol traffic across a network that does not normally support the protocol. To build a tunnel, you must define a tunnel interface on each of the two routers, and the tunnel interfaces must reference each other. At each router, the tunnel interface must be configured with a Layer 3 address. The tunnel endpoints, tunnel source, and tunnel destination must be defined, and the type of tunnel must be selected. Optional steps can be performed to customize the tunnel.

Remember to configure the router at each end of the tunnel. If only one side of a tunnel is configured, the tunnel interface may still come up and stay up (unless keepalive is configured), but packets going into the tunnel will be dropped.

GRE Tunnel Keepalive

Keepalive packets can be configured to be sent over IP-encapsulated GRE tunnels. You can specify the rate at which keepalives are sent and the number of times that a device will continue to send keepalive packets without a response before the interface becomes inactive. GRE keepalive packets may be sent from both sides of a tunnel or from just one side.

Before you begin

Ensure that the physical interface to be used as the tunnel source in this task is up and configured with the appropriate IP address. For hardware technical descriptions and information about installing interfaces, see the hardware installation and configuration publication for your product.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 0`	Specifies the interface type and number, and enters interface configuration mode. • To configure a tunnel, use **tunnel** for the *type* argument.
Step 4	**bandwidth** *kb/s* **Example:** `Router(config-if)# bandwidth 1000`	Sets the current bandwidth value for an interface and communicates it to higher-level protocols. • Specifies the tunnel bandwidth to be used to transmit packets. • Use the *kb/s* argument to set the bandwidth, in kilobits per second (kb/s).

	Command or Action	Purpose
		Note This is only a routing parameter; it does not affect the physical interface. The default bandwidth setting on a tunnel interface is 9.6 kb/s. You should set the bandwidth on a tunnel to an appropriate value.
Step 5	**keepalive** [*period* [*retries*]] **Example:** `Router(config-if)# keepalive 3 7`	(Optional) Specifies the number of times the device will continue to send keepalive packets without response before bringing the tunnel interface protocol down. • GRE keepalive packets may be configured either on only one side of the tunnel or on both. • If GRE keepalive is configured on both sides of the tunnel, the *period* and *retries* arguments can be different at each side of the link. **Note** This command is supported only on GRE point-to-point tunnels. **Note** The GRE tunnel keepalive feature should not be configured on a VRF tunnel. This combination of features is not supported.
Step 6	**tunnel source** {*ip-address* | *interface-type interface-number*} **Example:** `Router(config-if)# tunnel source TenGigabitEthernet 4/1/0`	Configures the tunnel source. **Note** The tunnel source IP address and destination IP addresses must be defined on two separate devices.
Step 7	**tunnel destination** {*hostname* | *ip-address*} **Example:** `Router(config-if)# tunnel destination 10.0.2.1`	Configures the tunnel destination. **Note** The tunnel source and destination IP addresses must be defined on two separate devices.
Step 8	**tunnel key** *key-number* **Example:** `Router(config-if)# tunnel key 1000`	(Optional) Enables an ID key for a tunnel interface. **Note** This command is supported only on GRE tunnel interfaces. We do not recommend relying on this key for security purposes.

	Command or Action	Purpose
Step 9	**tunnel mode gre** { **ip** | **multipoint**} **Example:** `Device(config-if)# tunnel mode gre ip`	Specifies the encapsulation protocol to be used in the tunnel.
Step 10	**ip mtu** *bytes* **Example:** `Device(config-if)# ip mtu 1400`	(Optional) Sets the MTU size of IP packets sent on an interface. • If an IP packet exceeds the MTU set for the interface, the Cisco software will fragment it unless the DF bit is set. • All devices on a physical medium must have the same protocol MTU in order to operate. • For IPv6 packets, use the **ipv6 mtu** command. **Note** If the **tunnel path-mtu-discovery** command is enabled do not configure this command.
Step 11	**ip tcp mss** *mss-value* **Example:** `Device(config-if)# ip tcp mss 250`	(Optional) Specifies the maximum segment size (MSS) for TCP connections that originate or terminate on a router.
Step 12	**tunnel path-mtu-discovery** [**age-timer** {*aging-mins* | **infinite**}] **Example:** `Device(config-if)# tunnel path-mtu-discovery`	(Optional) Enables PMTUD on a GRE or IP-in-IP tunnel interface. • When PMTUD is enabled on a tunnel interface, PMTUD will operate for GRE IP tunnel packets to minimize fragmentation in the path between the tunnel endpoints.
Step 13	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

What to Do Next

Proceed to the “Verifying Tunnel Configuration and Operation” section.

Configuring 6to4 Tunnels

Before you begin

With 6to4 tunnels, the tunnel destination is determined by the border-router IPv4 address, which is concatenated to the prefix 2002::/16 in the format 2002:*border-router-IPv4-address* ::/48. The border router at each end of a 6to4 tunnel must support both the IPv4 and IPv6 protocol stacks.

Note The configuration of only one IPv4-compatible tunnel and one 6to4 IPv6 tunnel is supported on a router. If you choose to configure both of these tunnel types on the same router, Cisco recommends that they not share the same tunnel source.

A 6to4 tunnel and an IPv4-compatible tunnel cannot share the same interface because both of them are NBMA "point-to-multipoint" access links, and only the tunnel source can be used to reorder the packets from a multiplexed packet stream into a single packet stream for an incoming interface. When a packet with an IPv4 protocol type of 41 arrives on an interface, the packet is mapped to an IPv6 tunnel interface on the basis of the IPv4 address. However, if both the 6to4 tunnel and the IPv4-compatible tunnel share the same source interface, the router cannot determine the IPv6 tunnel interface to which it should assign the incoming packet.

Manually configured IPv6 tunnels can share the same source interface because a manual tunnel is a "point-to-point" link, and both IPv4 source and the IPv4 destination of the tunnel are defined.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Router(config)# interface tunnel 0`	Specifies a tunnel interface and number and enters interface configuration mode.
Step 4	**ipv6 address** *ipv6-prefix/prefix-length* [**eui-64**] **Example:** `Router(config-if)# ipv6 address 2002:c0a8:6301:1::1/64`	Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface. • The 32 bits following the initial 2002::/16 prefix correspond to an IPv4 address assigned to the tunnel source.

	Command or Action	Purpose
		Note See the "Configuring Basic Connectivity for IPv6" module for more information on configuring IPv6 addresses.
Step 5	**tunnel source** {*ip-address* | *interface-type interface-number*} **Example:** `Router(config-if)# tunnel source TenGigabitEthernet 4/1/0`	Specifies the source IPv4 address or the source interface type and number for the tunnel interface. **Note** The interface type and number specified in the **tunnel source** command must be configured with an IPv4 address.
Step 6	**tunnel mode ipv6ip 6to4** **Example:** `Router(config-if)# tunnel mode ipv6ip 6to4`	Specifies an IPv6 overlay tunnel using a 6to4 address.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**ipv6 route** *ipv6-prefix* / *prefix-length* **tunnel** *tunnel-number* **Example:** `Router(config)# ipv6 route 2002::/16 tunnel 0`	Configures a static route to the specified tunnel interface. **Note** When configuring a 6to4 overlay tunnel, you must configure a static route for the IPv6 6to4 prefix 2002::/16 to the 6to4 tunnel interface. • The tunnel number specified in the **ipv6 route** command must be the same tunnel number specified in the **interface tunnel** command.
Step 9	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

What to Do Next

Proceed to the “Verifying Tunnel Configuration and Operation” section.

Verifying Tunnel Configuration and Operation

The **show** and **ping** commands in the steps below can be used in any sequence. The following commands can be used for GRE tunnels, IPv6 manually configured tunnels, and IPv6 over IPv4 GRE tunnels.

Procedure

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show interfaces tunnel** *number* [**accounting**]

Two routers are configured to be endpoints of a tunnel. Device A has TenGigabit Ethernet interface 4/1/0 configured as the source for tunnel interface 0 with an IPv4 address of 10.0.0.1 and an IPv6 prefix of 2001:0DB8:1111:2222::1/64. Device B has TenGigabit Ethernet interface 4/1/0 configured as the source for tunnel interface 1 with an IPv4 address of 10.0.0.2 and an IPv6 prefix of 2001:0DB8:1111:2222::2/64.

To verify that the tunnel source and destination addresses are configured, use the **show interfaces tunnel** command on Device A.

Example:

```
Device A# show interfaces tunnel 0

Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.0.0.1 (TenGigabitEthernet4/1/0), destination 10.0.0.2, fastswitch TTL
255
  Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
  Tunnel TTL 255
  Checksumming of packets disabled,  fast tunneling enabled
  Last input 00:00:14, output 00:00:04, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     4 packets input, 352 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     8 packets output, 704 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Step 3 **ping** [*protocol*] *destination*

To check that the local endpoint is configured and working, use the **ping** command on Device A.

Example:

```
DeviceA# ping 2001:0DB8:1111:2222::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:0DB8:1111:2222::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/20 ms
```

Step 4 **show ip route** [*address* [*mask*]]

To check that a route exists to the remote endpoint address, use the **show ip route** command.

Example:

```
DeviceA# show ip route 10.0.0.2

Routing entry for 10.0.0.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Routing Descriptor Blocks:
  * directly connected, via TenGigabitEthernet4/1/0
      Route metric is 0, traffic share count is 1
```

Step 5 **ping** [*protocol*] *destination*

To check that the remote endpoint address is reachable, use the **ping** command on Device A.

Note The remote endpoint address may not be reachable using the **ping** command because of filtering, but the tunnel traffic may still reach its destination.

Example:

```
DeviceA# ping 10.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/21/28 ms
```

To check that the remote IPv6 tunnel endpoint is reachable, use the **ping** command again on Device A. The note regarding filtering earlier in step also applies to this example.

Example:

```
DeviceA# ping 2001:0DB8:1111:2222::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/20 ms
```

These steps may be repeated at the other endpoint of the tunnel.

Configuration Examples for Implementing Tunnels

Example: Configuring a GRE IPv4 Tunnel

The following example shows a simple configuration of GRE tunneling. Note that TenGigabit Ethernet interface 4/1/0 is the tunnel source for Router A and the tunnel destination for Router B. TenGigabit Ethernet interface 4/1/1 is the tunnel source for Router B and the tunnel destination for Router A.

Router A

```
interface Tunnel 0
 ip address 10.1.1.2 255.255.255.0
 tunnel source TenGigabitEthernet 4/1/0
 tunnel destination 192.168.3.2
 tunnel mode gre ip
!
interface TenGigabitEthernet 4/1/0
 ip address 192.168.4.2 255.255.255.0
```

Router B

```
interface Tunnel 0
 ip address 10.1.1.1 255.255.255.0
 tunnel source TenGigabitEthernet 4/1/1
 tunnel destination 192.168.4.2
 tunnel mode gre ip
!
interface TenGigabitEthernet 4/1/1
 ip address 192.168.3.2 255.255.255.0
```

The following example configures a GRE tunnel running both IS-IS and IPv6 traffic between Router A and Router B:

Router A

```
ipv6 unicast-routing
clns routing
!
interface Tunnel 0
 no ip address
 ipv6 address 2001:0DB8:1111:2222::1/64
 ipv6 router isis
 tunnel source TenGigabitEthernet 4/1/0
 tunnel destination 10.0.0.2
 tunnel mode gre ip
!
interface TenGigabitEthernet 4/1/0
 ip address 10.0.0.1 255.255.255.0
!
router isis
 network 49.0000.0000.000a.00
```

Router B

```
ipv6 unicast-routing
clns routing
!
interface Tunnel 0
 no ip address
 ipv6 address 2001:0DB8:1111:2222::2/64
 ipv6 router isis
 tunnel source TenGigabitEthernet 4/1/0
 tunnel destination 10.0.0.1
 tunnel mode gre ip
!
interface TenGigabitEthernet 4/1/0
 ip address 10.0.0.2 255.255.255.0
!
router isis
 network 49.0000.0000.000b.00
 address-family ipv6
 redistribute static
 exit-address-family
```

Configuring QoS Options on Tunnel Interfaces Examples

The following sample configuration applies GTS directly on the tunnel interface. In this example, the configuration shapes the tunnel interface to an overall output rate of 500 kb/s.

```
interface Tunnel 0
 ip address 10.1.2.1 255.255.255.0
 traffic-shape rate 500000 125000 125000 1000
 tunnel source 10.1.1.1
 tunnel destination 10.2.2.2
```

The following sample configuration shows how to apply the same shaping policy to the tunnel interface with the MQC commands:

```
policy-map tunnel
 class class-default
 shape average 500000 125000 125000
!
interface Tunnel 0
 ip address 10.1.2.1 255.255.255.0
 service-policy output tunnel
 tunnel source 10.1.35.1
 tunnel destination 10.1.35.2
```

Policing Example

When an interface becomes congested and packets start to queue, you can apply a queueing method to packets that are waiting to be transmitted. Logical interfaces--tunnel interfaces in this example--do not inherently support a state of congestion and do not support the direct application of a service policy that applies a queueing method. Instead, you must apply a hierarchical policy. Create a "child" or lower-level policy that configures a queueing mechanism, such as low-latency queueing, with the **priority** command and CBWFQ with the **bandwidth** command.

```
policy-map child
```

```
 class voice
 priority 512
```

Create a "parent" or top-level policy that applies class-based shaping. Apply the child policy as a command under the parent policy because admission control for the child class is done according to the shaping rate for the parent class.

```
policy-map tunnel
 class class-default
 shape average 2000000
 service-policy child
```

Apply the parent policy to the tunnel interface.

```
interface tunnel 0
 service-policy tunnel
```

In the following example, a tunnel interface is configured with a service policy that applies queueing without shaping. A log message is displayed noting that this configuration is not supported.

```
Router(config)# interface tunnel1
Router(config-if)# service-policy output child
Class Based Weighted Fair Queueing not supported on this interface
```

How to Configure IPv6 over IPv4 GRE Tunnels

Configuring GRE on IPv6 Tunnels

GRE tunnels can be configured to run over an IPv6 network layer and to transport IPv4 and IPv6 packets in IPv6 tunnels.

Before you begin

When GRE IPv6 tunnels are configured, IPv6 addresses are assigned to the tunnel source and the tunnel destination. The tunnel interface can have either IPv4 addresses or IPv6 addresses assigned (this is not shown in the task). The host or device at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface tunnel** *tunnel-number* **Example:** `Device(config)# interface tunnel 0`	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	Enter one of the following commands: • **ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*} • **ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**] **Example:** `Device(config-if)# ipv6 address 3ffe:b00:c18:1::3/127`	Specifies the IPv6 network assigned to the interface and enables IPv6 processing on the interface. • If you specify the **eui-64** keyword, the software configures an IPv6 address for an interface and enables IPv6 processing on the interface using an EUI-64 interface ID in the low-order 64 bits of the address.
Step 5	**tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*} **Example:** `Device(config-if)# tunnel source Tengigabitethernet 4/1/0`	Specifies the source IPv4 address, IPv6 address, or the source interface type and number for the tunnel interface. • If an interface is specified, the interface must be configured with an IPv4 address.
Step 6	**tunnel destination** {*hostname* | *ip-address* | *ipv6-address*} **Example:** `Device(config-if)# tunnel destination 2001:DB8:1111:2222::1/64`	Specifies the destination IPv4 address, IPv6 address, or hostname for the tunnel interface.
Step 7	**tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **gre multipoint** | **gre ipv6** | **ipip** [**decapsulate-any**] | **iptalk** | **ipv6** | **mpls** | **nos**} **Example:** `Device(config-if)# tunnel mode gre ipv6`	Specifies a GRE IPv6 tunnel. **Note** The **tunnel mode gre ipv6** command specifies GRE as the encapsulation protocol for the tunnel.
Step 8	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuration Examples for IPv6 over IPv4 GRE Tunnels

Example: GRE Tunnel Running IS-IS and IPv6 Traffic

The following example configures a GRE tunnel running both IS-IS and IPv6 traffic between Router A and Router B:

Router A Configuration

```
ipv6 unicast-routing
clns routing
!
interface tunnel 0
 no ip address
 ipv6 address 3ffe:b00:c18:1::3/127
 ipv6 router isis
 tunnel source TenGigabitEthernet 4/1/0
 tunnel destination 2001:DB8:1111:2222::1/64
 tunnel mode gre ipv6
!
interface TenGigabitEthernet4/1/0
 ip address 10.0.0.1 255.255.255.0
!
router isis
 net 49.0000.0000.000a.00
```

Router B Configuration

```
ipv6 unicast-routing
clns routing
!
interface tunnel 0
 no ip address
 ipv6 address 3ffe:b00:c18:1::2/127
 ipv6 router isis
 tunnel source TenGigabitEthernet 4/1/0
 tunnel destination 2001:DB8:1111:2222::2/64
 tunnel mode gre ipv6
!
interface TenGigabitEthernet4/1/0
 ip address 10.0.0.2 255.255.255.0
!
router isis
 net 49.0000.0000.000b.00
 address-family ipv6
 redistribute static
 exit-address-family
```

Example: Tunnel Destination Address for IPv6 Tunnel

```
Router(config)#interface Tunnel0
Router(config-if)#ipv6 address 2001:1:1::1/48
Router(config-if)#tunnel source TenGigabitEthernet 4/1/0
Router(config-if)#tunnel destination 10.0.0.2
```

```
Router(config-if)#tunnel mode gre ipv6
Router(config-if)#exit
!
Router(config)#interface TenGigabitEthernet4/1/0
Router(config-if)#ip address 10.0.0.1 255.255.255.0
Router(config-if)#exit
!
Router(config)#ipv6 unicast-routing
Router(config)#router isis
Router(config)#net 49.0000.0000.000a.00
```

How to Configure GRE IPv6 Tunnels

Configuring GRE IPv6 Tunnels

Perform this task to configure a GRE tunnel on an IPv6 network. GRE tunnels can be configured to run over an IPv6 network layer and transport IPv6 and IPv4 packets through IPv6 tunnels.

Note You must enable IPv6 or configure IPv6 MTU size more than 1500 on a tunnel's exit interface to avoid receiving warning messages.

Before you begin

When GRE IPv6 tunnels are configured, IPv6 addresses are assigned to the tunnel source and the tunnel destination. The tunnel interface can have either IPv4 or IPv6 addresses (this is not shown in the task below). The host or device at each end of the configured tunnel must support both IPv4 and IPv6 protocol stacks.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Device(config)# interface tunnel 0`	Specifies a tunnel interface and number and enters interface configuration mode.
Step 4	**tunnel source** {*ipv6-address* | *interface-type interface-number*} **Example:**	Specifies the source IPv6 address or the source interface type and number for the tunnel interface.

	Command or Action	Purpose
	`Device(config-if)# tunnel source ethernet 0`	• If an interface type and number are specified, the interface must be configured with an IPv6 address. **Note** Only the syntax used in this context is displayed. For more details, see the IPv6 Command Reference.
Step 5	**tunnel destination** *ipv6-address* **Example:** `Device(config-if)# tunnel destination 2001:0DB8:0C18:2::300`	Specifies the destination IPv6 address for the tunnel interface. **Note** Only the syntax used in this context is displayed. For more details, see the IPv6 Command Reference.
Step 6	**tunnel mode gre ipv6** **Example:** `Device(config-if)# tunnel mode gre ipv6`	Specifies a GRE IPv6 tunnel. **Note** The **tunnel mode gre ipv6** command specifies GRE as the encapsulation protocol for the tunnel interface. Only the syntax used in this context is displayed. For more details, see the IPv6 Command Reference.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for GRE IPv6 Tunnels

Example: Configuring GRE IPv6 Tunnels

The following example shows how to configure a GRE tunnel over an IPv6 transport. In this example, Ethernet0/0 has an IPv6 address, and this is the source address used by the tunnel interface. The destination IPv6 address of the tunnel is specified directly. In this example, the tunnel carries both IPv4 and IS-IS traffic.

```
interface Tunnel0
 ip address 10.1.1.1 255.255.255.0
 ip router isis
 tunnel source Ethernet0/0
 tunnel destination 2001:DB8:1111:2222::1
 tunnel mode gre ipv6
!
interface Ethernet0/0
 no ip address
 ipv6 address 2001:DB8:1111:1111::1/64
!
```

```
router isis
 net 49.0001.0000.0000.000a.00
```

Additional References

The following sections provide references related to the GRE feature.

Related Documents

Related Topic	Document Title
CMTS Command Reference	Cisco CMTS Cable Command Reference, at the following URL: http://www.cisco.com/c/en/us/td/docs/cable/cmts/cmd_ref/b_cmts_cable_cmd_ref.html
Configuring GRE Tunnel over Cable	Configuring GRE Tunnel over Cable, at the following URL: http://www.cisco.com/en/US/tech/tk86/tk89/technologies_configuration_example09186a008011520d.sht

Standards

Standard	Title
SP-RFIv1.1-I09-020830	Data-over-Cable Service Interface Specifications Radio Frequency Interface Specification, version 1.1 (http://www.cablemodem.com)

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://tools.cisco.com/ITDIT/MIBS/servlet/index

RFCs

RFC	Title
RFC 1701	Generic Routing Encapsulation (GRE)
RFC 1702	Generic Routing Encapsulation over IPv4 networks
RFC 1853	IP in IP Tunneling
RFC 2003	IP Encapsulation within IP
RFC 2784	Generic Routing Ecapsulation (GRE)

RFC	Title
RFC 2890	Key and Sequence Number Extensions to GRE

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/cisco/web/support/index.html

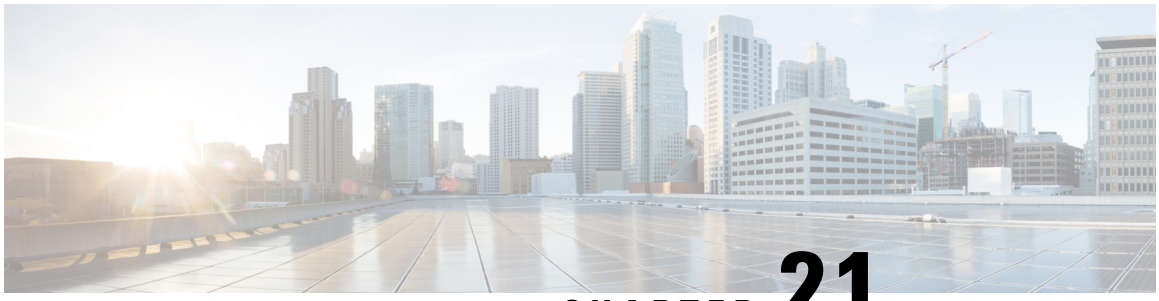
Feature Information for Generic Routing Encapsulation

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 63: Feature Information for Generic Routing Encapsulation

Feature Name	Releases	Feature Information
Generic Routing Encapsulation	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Routers.



CHAPTER 21

Transparent LAN Service over Cable

This document describes the Transparent LAN Service (TLS) over Cable feature, which enhances existing Wide Area Network (WAN) support to provide more flexible Managed Access for multiple Internet service provider (ISP) support over a hybrid fiber-coaxial (HFC) cable network. This feature allows service providers to create a Layer 2 tunnel by mapping an upstream service identifier (SID) to an IEEE 802.1Q Virtual Local Area Network (VLAN).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 417
- Prerequisites for Transparent LAN Service over Cable, on page 418
- Restrictions for Transparent LAN Service over Cable, on page 418
- Information About Transparent LAN Service over Cable, on page 419
- How to Configure the Transparent LAN Service over Cable, on page 421
- Configuration Examples for Transparent LAN Service over Cable, on page 424
- Verifying the Transparent LAN Service over Cable Configuration, on page 425
- Additional References, on page 426
- Feature Information for Transparent LAN Service over Cable , on page 427

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 64: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Transparent LAN Service over Cable

- You must know the hardware (MAC) addresses of the cable modems that are to be mapped to IEEE 802.1Q VLANs.
- You must create a bridge group for each separate customer on the Layer 2 bridge aggregator, so that traffic from all of the Customer Premises Equipment (CPE) devices for the customer is grouped together into the same 802.1Q tunnel.

Restrictions for Transparent LAN Service over Cable

- Configuring 802.1Q for a particular cable modem removes any previous cable modem configuration on the router.
- We strongly recommend that TLS over Cable only be used when Baseline Privacy Interface (BPI) is enabled in the environment. If BPI is not enabled when using the TLS feature, traffic can flow between multiple virtual private networks (VPNs), and become vulnerable to denial-of-service attacks or snooping. We also recommend that remote networks be isolated with a gateway or firewall router when BPI is not enabled.

When the TLS feature is used with Layer 2 VPNs, the participating cable modems *must* have the Baseline Privacy Interface security feature (BPI) enabled. Otherwise, the Cisco CMTS drops such Layer 2 traffic in the upstream or downstream.

- Packets are mapped to their Layer 2 tunnel only on the basis of Layer 2 information (the cable modem's MAC address and primary SID). Layer 3 services, such as access lists, IP address source-verify, and IP QoS, are not supported as packets are sent through the tunnel.
- All traffic from a cable modem is mapped to the same Layer 2 tunnel. It is not possible to differentiate traffic from different customer premises equipment (CPE) devices behind the cable modem.
- CPE learning is not available when using the Transparent LAN Service over Cable feature. When a cable modem is mapped to a Layer 2 tunnel, the **show interface cable modem** command shows that the IP addresses for its CPE devices are "unavailable."
- DOCSIS QoS is supported across the Layer 2 tunnel only on the primary SID. Traffic using secondary services uses the same Layer 2 tunnel as the primary SID.
- The Spanning Tree Protocol (STP) cannot be used with devices (cable modems, their CPE devices, and the endpoint CPE devices) that are using this feature. In particular, Spanning Tree Protocol cannot be used between the VLAN bridge aggregator and the endpoint customer devices.
- The following restrictions apply to Layer 2 tunnels over an Ethernet IEEE 802.1Q VLAN interface:
  - IEEE 802.1Q tunnels are supported only on Ten Gigabit Ethernet interfaces.
  - The Cisco CMTS router supports a maximum of 4095 VLAN IDs, but the switches acting as the bridge aggregator might support a lower number of VLAN IDs. If this is the case, the Cisco CMTS should be configured only for the maximum number of VLANs that are supported by the bridge aggregator switches.

Information About Transparent LAN Service over Cable

This section contains the following:

Feature Overview

The Transparent LAN Service over Cable feature enables service providers to provide Layer 2 tunnels for traffic to and from cable modems. This allows customers to create their own virtual local area network (VLAN) using any number of cable modems in multiple sites.

On the Cisco CMTS, you map each cable modem (on the basis of its MAC address) to the appropriate VLAN. The CMTS then creates an internal database of this one-to-one mapping of cable modems to VLANs, and uses it to encapsulate packets for the appropriate VLAN.

The CMTS encapsulates the CPE traffic from mapped cable modems using the following method:

- IEEE 802.1Q Mapping—The cable modem's MAC address is mapped to an IEEE 802.1Q VLAN on a specific Ten Gigabit Ethernet interface, so that all traffic from the cable modem is tagged with the specified VLAN ID.

Traffic to and from this group of cable modems is bridged into a single logical network (the VLAN) by the bridge aggregator, creating a secure Virtual Private Network (VPN) for that particular group of cable modems. Traffic in one VLAN cannot be sent into another VLAN, unless specifically done so by an external router.

The switch acting as the Layer 2 Bridge Aggregator uses the VLAN tagging to forward the traffic to the appropriate destination. This frees up service providers from needing to know the addressing, routing, and topological details of the customer's network.

Transparent LAN Service and Layer 2 Virtual Private Networks

In addition, service providers can provide a Layer 2 VPN with only minimal configuration changes on the provider's routers. The service subscriber does not need to make any changes to their private network or cable modems, nor does the service provider have to provide any special DOCSIS configuration files to enable this feature.

For the TLS feature with Layer 2 VPNs:

- When the TLS feature is used with Layer 2 VPNs, the participating cable modems must have the Baseline Privacy Interface security feature (BPI) enabled. Otherwise, the Cisco CMTS drops such Layer 2 traffic in the upstream or downstream.
- Information about Customer Premises Equipment (CPE) does not display in the output of the **show cable modem** command.

IEEE 802.1Q Mapping

This section describes the mapping of cable modems to an IEEE 802.1Q VLAN, as it is available in the Transparent LAN Service over Cable feature:

Overview

The Transparent LAN Service over Cable feature enables service providers to provide Layer 2 tunnels over an Ethernet network, using IEEE 802.1Q standard tags. This allows customers to create their own virtual network using any number of cable modems in different sites.

On the Cisco CMTS, you map each cable modem (on the basis of its MAC address) to the appropriate VLAN. The CMTS then creates an internal database of this one-to-one mapping of cable modems to VLANs, and uses it to encapsulate packets for the appropriate VLAN.

The CMTS encapsulates the CPE traffic from mapped cable modems using VLAN tags, as defined in IEEE 802.1Q-1993, IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks . The switch acting as the Layer 2 Bridge Aggregator uses the VLAN tagging to forward the packets to the appropriate destination.

Traffic to and from this group of cable modems is bridged into a single logical network by the bridge aggregator, creating a secure Virtual Private Network (VPN) for that particular group of cable modems. Traffic in one VLAN cannot be sent into another VLAN, unless specifically done so by an external router.

Details of IEEE 802.1Q Mapping

To implement the Transparent LAN Service over Cable feature using IEEE 802.1Q VLANs, a service provider must perform the following configuration steps:

1. Identify the cable modems and their MAC addresses that should be mapped to the IEEE 802.1Q VLANs.
2. Create the required VLANs on the router that is acting as the bridge aggregator.
3. Enable Layer 2 mapping on the Cisco CMTS, and then map each cable modem on that Cisco CMTS to the appropriate VLAN.

After the Transparent LAN Service over Cable feature has been enabled and configured to use IEEE 802.1Q mappings, the Cisco CMTS immediately begins mapping traffic between the associated cable modems and VLANs. For efficient mapping, the Cisco CMTS maintains an internal database that links each cable modem's primary service flow ID (SFID) and service ID (SID) to the appropriate VLAN and Ethernet interface. This ensures that all service flows from the cable modem are routed properly.

When the Cisco CMTS receives a packet on an upstream, it looks up its SID to see if it is mapped to a VLAN. If so, and if the packet's source MAC address is not the cable modem's MAC address, the Cisco CMTS inserts the appropriate IEEE 802.1Q VLAN tag into the packet's header and forwards the packet to the appropriate Ethernet interface. If the packet is not being mapped, or if the packet originated from the cable modem, the Cisco CMTS routes the packet using the normal Layer 3 processes.

When the Cisco CMTS receives a packet from a WAN interface that is encapsulated with an IEEE 802.1Q VLAN tag, it looks up the packet's SID to see if it belongs to a cable modem being mapped. If so, the Cisco CMTS strips off the VLAN tag, adds the proper DOCSIS header, and transmits the packet on the appropriate downstream interface. If the packet is not being mapped, the Cisco CMTS continues with the normal Layer 3 processing.

Benefits

The Transparent LAN Service over Cable feature provides the following benefits to cable service providers and their partners and customers:

- Provides Layer 2 level mapping, which is transparent to Layer 3 protocols and services. This means that service providers do not need to know the details of their customers' network topologies, routing protocols, or IP addressing.
- Allows service providers to maximize the use of their existing Ethernet WAN networks. Multiple customers can be combined on the same outgoing interface, while still ensuring that each customer's network is kept private while it is transmitted over the tunnel.
- Provides a highly flexible and scalable solution for multiple customers. The service provider needs to create only one bridge group for each VPN, and then only one VLAN mapping for each cable modem should participate in that VPN tunnel.
- Customers retain full control over their private networks, while service providers retain full control over cable modems and the rest of the cable and WAN networks. Only the CPE traffic from the cable modems is mapped into the L2VPN tunnel, while traffic originating at the cable modem continues to be processed as normal by the service provider's network.
- Allows service providers to mix tunneled and non-tunneled cable modems on the same DOCSIS cable network.
- Allows customers to create a single, secure virtual network with Ethernet Layer 2 connectivity for multiple sites.
- Allows multiple tunnels from different customers and endpoints to be aggregated into a single bridge, so as to maximize the use of bandwidth and other network resources.
- Supports the tunneling of multiple Layer 3, non-IP protocols, and not just IP Layer 3 services, as is the case with Layer 3 solutions, such as Multiprotocol Label Switching (MPLS) VPNs.
- All DOCSIS services, including BPI+ encryption and authentication, continue to be supported for all cable modems.

How to Configure the Transparent LAN Service over Cable

This section contains the following:

Configuring IEEE 802.1Q VLAN Mapping

This section describes how to enable Layer 2 mapping on the Cisco CMTS, and then to map particular cable modems to an IEEE 802.1Q VLAN.

Enabling and Configuring Layer 2 Tunneling for IEEE 802.1Q Mapping

This section describes how to enable Layer 2 mapping on the Cisco CMTS, and then to map particular cable modems to IEEE 802.1Q VLANs on a Ten Gigabit Ethernet interface.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable l2-vpn-service xconnect nsi dot1q** **Example:** `Router(config)# cable l2-vpn-service xconnect nsi dot1q`	Enables Layer 2 tunneling for IEEE 802.1Q VLAN mapping. **Note** It is not required to configure VLAN trunking on the Cisco CMTS. Though VLAN trunking is supported, be aware of additional impact of VLAN trunking on the Cisco CMTS.
Step 4	**cable dot1q-vc-map** *mac-address ethernet-interface vlan-id* [*cust-name*] **Example:** `Router(config)# cable dot1q-vc-map 0000.0C04.0506 TenGigabitEthernet4/1/0 10`	Maps the specified MAC address of a cable modem to the indicated VLAN and Ten Gigabit Ethernet interface. **Note** Repeat this command for each cable modem that is to be mapped to an IEEE 802.1Q VLAN.
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Creating the IEEE 802.1Q VLAN Bridge Group

This section describes the minimum configuration needed to configure a Cisco router, which is acting as an IEEE 802.1Q VLAN bridge aggregator, so that it can terminate the VLANs being used with the Transparent LAN Service over Cable feature.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal` **Example:**	Enters global configuration mode.
Step 3	**interface TenGigabitEthernet** *slot/subslot/port* **Example:** `Router(config)# interface TenGigabitEthernet4/1/0`	Enters interface configuration mode for the Ten Gigabit Ethernet interface.
Step 4	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.10.10.85 255.255.255.0`	Configures the interface with the specified IP address and subnet mask.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration and returns to global configuration mode.
Step 6	**interface TenGigabitEthernet** *slot/subslot/port.y* **Example:** `Router(config)# interface TenGigabitEthernet4/1/0.10`	Creates a subinterface on the Ten Gigabit Ethernet interface. **Note** To simplify network management, set the subinterface number to the same value as the VLAN ID that will use this subinterface (which in this case is 10). **Note** The steps to create a subinterface is not essential for dot1q tagging of frames, but it is recommended.

	Command or Action	Purpose
Step 7	**bridge group** *number* **Example:** `Router(config-if)# bridge group 20`	Configures this subinterface to belong to the specified bridge group. **Note** Repeat steps Step 5 through Step 7 for each subinterface to be created and bridged.
Step 8	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Transparent LAN Service over Cable

This section lists sample configurations for the Transparent LAN Service over Cable feature on a CMTS router and on a Cisco router acting as a bridge aggregator:

Example: Configuring IEEE 802.1Q VLAN Mapping

The following partial configuration shows a typical configuration that shows a number of cable modems being mapped to two different IEEE 802.1Q VLANs.

```
cable l2-vpn-service xconnect nsi dot1q
! Customer 1
cable dot1q-vc-map 000C.0e03.69f9 TenGigabitEthernet 4/1/0 10 Customer1
cable dot1q-vc-map 0010.7bea.9c95 TenGigabitEthernet 4/1/0 11 Customer1
cable dot1q-vc-map 0010.7bed.81c2 TenGigabitEthernet 4/1/0 12 Customer1
cable dot1q-vc-map 0010.7bed.9b1a TenGigabitEthernet 4/1/0 13 Customer1
! Customer 2
cable dot1q-vc-map 0002.fdfa.137d TenGigabitEthernet 4/1/0 20 Customer2
cable dot1q-vc-map 0006.28f9.9d19 TenGigabitEthernet 4/1/0 21 Customer2
cable dot1q-vc-map 000C.7b6b.58c1 TenGigabitEthernet 4/1/0 22 Customer2
cable dot1q-vc-map 000C.7bed.9dbb TenGigabitEthernet 4/1/0 23 Customer2
cable dot1q-vc-map 000C.7b43.aa7f TenGigabitEthernet 4/1/0 24 Customer2
cable dot1q-vc-map 0050.7302.3d83 TenGigabitEthernet 4/1/0 25 Customer2
...
```

Example: Configuring IEEE 802.1Q Bridge Aggregator

The following example shows a router being used as a bridge aggregator to transmit VLANs across the same Ten Gigabit Ethernet interface, using IEEE 802.1Q tagging.

```
!
interface TenGigabitEthernet4/1/0
 ip address 10.10.10.31 255.255.255.0
 duplex full
 speed auto
!
interface TenGigabitEthernet4/1/0.10
 description Customer1-site10
 encapsulation dot1Q 10
```

```
 bridge-group 200
interface TenGigabitEthernet4/1/0.11
 description Customer1-site11
 encapsulation dot1Q 11
 bridge-group 200
interface TenGigabitEthernet4/1/0.12
 description Customer1-site12
 encapsulation dot1Q 12
 bridge-group 200
interface TenGigabitEthernet4/1/0.13
 description Customer1-site13
 encapsulation dot1Q 13
 bridge-group 200
!-----------------------------------
interface TenGigabitEthernet4/1/0.20
 description Customer2-site20
 encapsulation dot1Q 20
 bridge-group 201
interface TenGigabitEthernet4/1/0.21
 description Customer2-site21
 encapsulation dot1Q 21
 bridge-group 201
interface TenGigabitEthernet4/1/0.22
 description Customer2-site22
 encapsulation dot1Q 22
 bridge-group 201
interface TenGigabitEthernet4/1/0.23
 description Customer2-site23
 encapsulation dot1Q 23
 bridge-group 201
interface TenGigabitEthernet4/1/0.24
 description Customer2-site24
 encapsulation dot1Q 24
 bridge-group 201
interface TenGigabitEthernet4/1/0.25
 description Customer2-site25
 encapsulation dot1Q 25
 bridge-group 201
!
bridge 200 protocol ieee
bridge 201 protocol ieee
...
```

Verifying the Transparent LAN Service over Cable Configuration

- **show cable l2-vpn xconnect dot1q-vc-map** —Displays the mapping information of the cable modems to IEEE 802.1Q VLANs.

  Following is a sample output of the command:

  ```
  Router# show cable l2-vpn xconnect dot1q-vc-map

  MAC Address    Ethernet Interface      VLAN ID   Cable Intf  SID  Customer Name/VPNID
  38c8.5cac.4a62 TenGigabitEthernet4/1/2     56    Cable3/0/0  4    Customer2
  38c8.5cfe.f6fa TenGigabitEthernet4/1/2     34    Cable3/0/0  3    Customer1
  602a.d083.2e1c TenGigabitEthernet4/1/4     43    Cable3/0/0  5    Customer3
  ```

- **show cable l2-vpn xconnect dot1q-vc-map customer** *name*—Displays the mapping information of the cable modems to IEEE 802.1Q VLANs for the specified customer name.

Following is a sample output of the command.

```
Router# show cable l2-vpn xconnect dot1q-vc-map customer Customer1

MAC Address    Ethernet Interface      VLAN ID   Cable Intf  SID  Customer Name/VPNID
38c8.5cfe.f6fa TenGigabitEthernet4/1/2     34    Cable3/0/0  3    Customer1
```

- **show cable l2-vpn xconnect dot1q-vc-map** *mac-address*—Displays the mapping information of the cable modems to IEEE 802.1Q VLANs for the specified MAC address.

  Following is a sample output of the command:

```
Router# show cable l2-vpn xconnect dot1q-vc-map 38c8.5cac.4a62

MAC Address    Ethernet Interface      VLAN ID   Cable Intf  SID  Customer Name/VPNID
38c8.5cac.4a62 TenGigabitEthernet4/1/2     56    Cable3/0/0  4    Customer2
```

- **show cable l2-vpn xconnect dot1q-vc-map** *mac-address* **verbose**—Displays additional information about the Layer 2 mapping, including the number of packets and bytes received on the upstream and downstream.

  Following is a sample output of the command:

```
Router# show cable l2-vpn xconnect dot1q-vc-map 38c8.5cac.4a62 verbose

MAC Address                     : 38c8.5cac.4a62
Customer Name                   : Customer2
Prim Sid                        : 4
Cable Interface                 : Cable3/0/0
Ethernet Interface              : TenGigabitEthernet4/1/2
DOT1Q VLAN ID                   : 56
Total US pkts                   : 1
Total US bytes                  : 342
Total DS pkts                   : 4
Total DS bytes                  : 512
```

Additional References

Standards

Standards	Title
SP-RFIv1.1-I08-020301	Data-over-Cable Service Interface Specifications Radio Frequency Interface Specification
IEEE 802.1Q, 1998 Edition	IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks

RFCs

RFCs[2]	Title
RFC 1163	A Border Gateway Protocol

RFCs[2]	Title
RFC 1164	Application of the Border Gateway Protocol in the Internet
RFC 2233	DOCSIS OSSI Objects Support
RFC 2283	Multiprotocol Extensions for BGP-4
RFC 2665	DOCSIS Ethernet MIB Objects Support
RFC 2669	Cable Device MIB

[2] Not all supported RFCs are listed.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/cisco/web/support/index.html

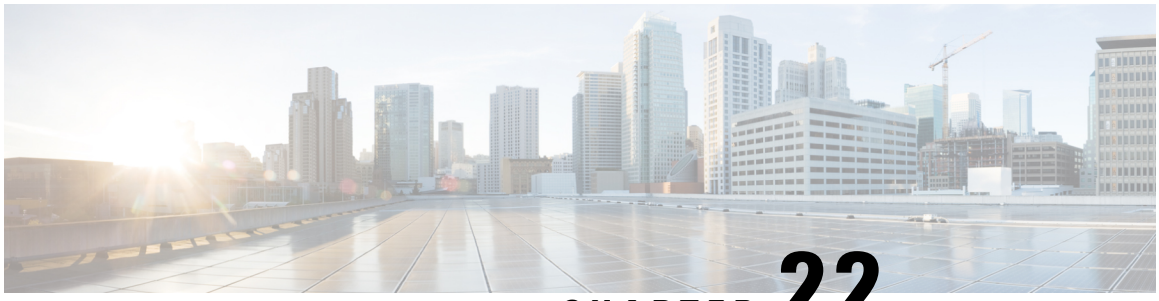
Feature Information for Transparent LAN Service over Cable

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 65: Feature Information for Upstream Channel Bonding

Feature Name	Releases	Feature Information
Transparent LAN Service over Cable	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router.



CHAPTER 22

Downgrading Channel Bonding in Battery Backup Mode

Cisco CMTS supports downgrading the channel bonding for cable modems and media terminal adapters (MTAs) in battery backup mode.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 429
- Prerequisites for Downgrading Channel Bonding in Battery Backup Mode, on page 430
- Restrictions for Downgrading Channel Bonding in Battery Backup Mode, on page 430
- Information About Downgrading Channel Bonding in Battery Backup Mode, on page 431
- How to Configure Downgrading Channel Bonding in Battery Backup Mode, on page 431
- Verifying the Configuration for Channel Bonding Downgrade in Battery Backup Mode, on page 434
- Additional References, on page 437
- Feature Information for Downgrading Channel Bonding in Battery Backup Mode, on page 438

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 66: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Downgrading Channel Bonding in Battery Backup Mode

- The cable modem must be DOCSIS3.0-compliant with battery backup capability.
- At last one free Downstream Resilient Bonding Group (RBG) must be available.

Note For information about how to reserve RBG and verify reserved RBG, refer to Downstream Resiliency Bonding Group , on page 279

Restrictions for Downgrading Channel Bonding in Battery Backup Mode

- If the cable modem does not support the CM-STATUS events 9 and 10, channel bonding is not downgraded for the cable modem in battery backup mode.

Note We recommend that you configure separate dynamic bonding groups for each primary channel in a MAC domain.

- If the cable modem has an active voice call, channel bonding is not downgraded for the cable modem in battery backup mode.
- If the cable modem is working on the protect line card, channel bonding is not downgraded if its primary channel is not included in the dynamic bonding group.
- If the line card switches over when the cable modem is entering or exiting the battery backup mode, the cable modem may go offline.

Information About Downgrading Channel Bonding in Battery Backup Mode

When this feature is enabled and the cable modem enters the battery backup mode, channel bonding is downgraded to one downstream and one upstream channels (battery backup 1x1 mode). This feature reduces the power usage when the cable modem is running on battery backup. When the cable modem returns to the AC power mode, the channel bonding is returned to its original configuration. You can configure this feature globally and for each MAC domain.

Note We recommend that you enable this feature globally and for each MAC domain.

The cable modem uses the following CM-STATUS events to indicate its power status to the Cisco CMTS:

- 9—Indicates that the cable modem is operating in battery backup mode.
- 10—Indicates that the cable modem has returned to AC power mode.

When this feature is disabled, cable modem cannot downgrade the channel bonding even if it is running on battery backup.

How to Configure Downgrading Channel Bonding in Battery Backup Mode

This section contains the following:

Configuring Channel Bonding Downgrade in Battery Backup Mode Globally

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable reduction-mode mta-battery enable** **Example:** `Router(config)# cable reduction-mode mta-battery enable`	Enables the channel bonding downgrade for cable modems in battery backup mode.
Step 4	**cable reduction-mode mta-battery dampen-time** *seconds* **Example:** `Router(config)# cable reduction-mode mta-battery dampen-time 40`	(Optional) Configures the dampen time, in seconds, to defer the cable modems from entering or exiting the channel bonding downgrade 1x1 mode.
Step 5	**cable reduction-mode mta-battery ranging-init-technique** *us-ranging-init-technique* **Example:** `Router(config)# cable reduction-mode mta-battery ranging-init-technique 3`	(Optional) Configures the init-ranging technique.
Step 6	**cable reduction-mode mta-battery dynamic-channel-percent** *percent* **Example:** `Router(config)# cable reduction-mode mta-battery dynamic-channel-percent 10`	(Optional) Configures the maximum and first try percentage of dynamic channel bandwidth in battery backup mode. **Note** Ensure to leave enough bandwidth for primary channel so that it can allocate dynamic channel bandwidth when it joins a newly created dynamic bonding group.
Step 7	**exit** **Example:** `Router(config)# exit`	Returns to the privileged EXEC mode.

Configuring Channel Bonding Downgrade in Battery Backup Mode for MAC Domain

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface wideband-cable** *slot/subslot/port:wideband-channel* **Example:** `Router(config)# interface wideband-cable 1/0/0:7`	Configures a wideband cable interface.
Step 4	**cable ds-resiliency** **Example:** `Router(config-if)# cable ds-resiliency`	Reserves a resiliency bonding group or WB interface for usage on a line card, on a per controller basis.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Returns to the global configuration mode.
Step 6	**interface cable** *slot/subslot/port* **Example:** `Router(config)# interface cable 9/0/0`	Specifies the cable interface on the router and enters the interface configuration mode.
Step 7	**cable reduction-mode mta-battery enable** **Example:** `Router(config-if)# cable reduction-mode mta-battery enable`	Enables the channel bonding downgrade for cable modems in battery backup mode for each MAC domain.
Step 8	**cable cm-status enable 9** **Example:** `Router(config-if)# cable cm-status enable 9`	Enables the CM-STATUS event 9 for the MAC domain. The value 9 indicates that the cable modem is operating in battery backup mode.
Step 9	**cable cm-status enable 10** **Example:** `Router(config-if)# cable cm-status enable 10`	Enables the CM-STATUS event 10 for the MAC domain. The value 10 indicates that the cable modem has returned to AC power mode.

	Command or Action	Purpose
Step 10	**end** **Example:** `Router(config-if)# end`	Returns to the privileged EXEC mode.

Verifying the Configuration for Channel Bonding Downgrade in Battery Backup Mode

- **show cable modem**—Displays information if the cable modem is running in battery backup mode.

  Following is a sample output of the command:

```
Router# show cable modem

 D
MAC Address    IP Address      I/F           MAC               Prim  RxPwr  Timing Num
 I                                           State             Sid   (dBmv) Offset CPE
 P
f45f.d4a1.b75a ---             C6/1/0/UB     p-online(pt)      846  !-3.50  1475   0
 N
c427.9551.3489 30.154.1.12     C6/1/0/UB     w-online(pt)      930   -0.50  1579   2
 Y
f45f.d4a1.b762 30.55.223.253   C6/1/0/UB     w-online          1770  0.00   1503   0
 Y
0016.925e.661a 30.55.230.136   C6/1/0/U0     online(pt)        825   -0.50  1467   1
 N
4458.2945.458a 30.0.7.72       C6/1/0/UB     w-online          3916  0.00   1511   2
 Y
4458.2945.401e ---             C6/1/0/UB     w-online(pt)      847   -0.50  1473   1
 N
4458.2945.20c6 ---             C6/1/0/UB     w-online(pt)(bm)  895   0.00   1481   0
 N
```

- **show cable modem reduction-mode mta-battery**—Displays the channel bonding downgrade information for cable modems in battery backup mode.

  Following is a sample output of the command:

```
Router# show cable modem reduction-mode mta-battery

                              Orig BG                        Curr BG
I/F     MAC Address    ID      I/F       RFs     ID          I/F        Upstream
------- -------------- ------------------------------------------------------------
C7/0/0  0025.2eaf.843e 897  Wi7/0/0:0     4      252         Wi7/0/0:1      US0
C7/0/0  0025.2eaf.8356 897  Wi7/0/0:0     4      252         Wi7/0/0:1      US0
C7/0/0  0015.d176.5199 897  Wi7/0/0:0     4      252         Wi7/0/0:1      US0
```

  Following is a sample output of the command for a cable modem when the MAC address is specified:

```
Router# show cable modem 0025.2eaf.843e reduction-mode mta-battery

                              Orig BG                        Curr BG
I/F     MAC Address    ID      I/F       RFs     ID          I/F        Upstream
------- -------------- ------------------------------------------------------------
```

```
C7/0/0  0025.2eaf.843e 897  Wi7/0/0:0     4      252        Wi7/0/0:1      US0
```

Following is a sample output of the command for a cable modem when the IP address is specified:

```
Router# show cable modem 90.18.0.9 reduction-mode mta-battery

                              Orig BG                       Curr BG
I/F     MAC Address    ID      I/F       RFs     ID          I/F        Upstream
------- -------------- ------------------------------------------------------
C7/0/0  0025.2eaf.843e 897  Wi7/0/0:0     4      252        Wi7/0/0:1      US0
```

Following is a sample output of the command for a cable modem when the IPv6 address is specified:

```
Router# show cable modem 2001:18::9 reduction-mode mta-battery

                              Orig BG                       Curr BG
I/F     MAC Address    ID      I/F       RFs     ID          I/F        Upstream
------- -------------- ------------------------------------------------------
C7/0/0  0025.2eaf.843e 897  Wi7/0/0:0     4      252        Wi7/0/0:1      US0
```

- **show cable modem verbose**—Displays the detailed information for the cable modem.

  Following is a sample output of the command:

```
Router# show cable modem 54d4.6ffb.30fd verbose

MAC Address                      : 54d4.6ffb.30fd
IP Address                       : 40.4.58.14
IPv6 Address                     : 2001:40:4:58:741A:408D:7E4B:D7C8
Dual IP                          : Y
Prim Sid                         : 9
Host Interface                   : C7/0/0/UB
MD-DS-SG / MD-US-SG              : 1 / 1
MD-CM-SG                         : 0x3C0101
Primary Wideband Channel ID      : 897 (Wi7/0/0:0)
Primary Downstream               : In7/0/0:2 (RfId : 722)
Wideband Capable                 : Y
RCP Index                        : 3
RCP ID                           : 00 10 00 00 08
Downstream Channel DCID RF Channel : 99    7/0/0:2
Downstream Channel DCID RF Channel : 97    7/0/0:0
Downstream Channel DCID RF Channel : 98    7/0/0:1
Downstream Channel DCID RF Channel : 100   7/0/0:3
Multi-Transmit Channel Mode      : Y
Extended Upstream Transmit Power : 0dB
Upstream Channel                 : US0        US1
Ranging Status                   : sta        sta
Upstream SNR (dB)                : 36.12      32.55
Upstream Data SNR (dB)           : --         --
Received Power (dBmV)            : 0.00       0.00
Reported Transmit Power (dBmV)   : 25.25      26.00
Peak Transmit Power (dBmV)       : 54.00      54.00
Phy Max Power (dBmV)             : 54.00      54.00
Minimum Transmit Power (dBmV)    : 24.00      24.00
Timing Offset            (97.6 ns): 1226       1226
Initial Timing Offset            : 1229       973
Rng Timing Adj Moving Avg(0.381 ns): -1         0
Rng Timing Adj Lt Moving Avg     : -7         0
Rng Timing Adj Minimum           : -768       0
Rng Timing Adj Maximum           : 0          64768
Pre-EQ Good                      : 0          0
Pre-EQ Scaled                    : 0          0
Pre-EQ Impulse                   : 0          0
```

```
Pre-EQ Direct Loads                : 0          0
Good Codewords rx                  : 515        472
Corrected Codewords rx             : 0          0
Uncorrectable Codewords rx         : 0          0
Phy Operating Mode                 : atdma*     atdma*
sysDescr                           :
Downstream Power                   : 0.00 dBmV (SNR = ----- dB)
MAC Version                        : DOC3.0
QoS Provisioned Mode               : DOC1.1
Enable DOCSIS2.0 Mode              : Y
Modem Status                       : {Modem= w-online, Security=disabled}
Capabilities                       : {Frag=N, Concat=N, PHS=Y}
Security Capabilities              : {Priv=, EAE=Y, Key_len=}
L2VPN Capabilities                 : {L2VPN=N, eSAFE=N}
Sid/Said Limit                     : {Max US Sids=16, Max DS Saids=15}
Optional Filtering Support         : {802.1P=N, 802.1Q=N, DUT=N}
Transmit Equalizer Support         : {Taps/Symbol= 1, Num of Taps= 24}
Number of CPE IPs                  : 0(Max CPE IPs = 16)
CFG Max-CPE                        : 200
Flaps                              : 0()
Errors                             : 0 CRCs, 0 HCSes
Stn Mtn Failures                   : 0 aborts, 0 exhausted
Total US Flows                     : 1(1 active)
Total DS Flows                     : 1(1 active)
Total US Data                      : 7 packets, 2006 bytes
Total US Throughput                : 0 bits/sec, 0 packets/sec
Total DS Data                      : 5 packets, 1202 bytes
Total DS Throughput                : 0 bits/sec, 0 packets/sec
LB group ID assigned (index)       : 2151416065 (48131)
LB group ID in config file (index) : N/A (N/A)
LB policy ID                       : 0
LB policy ID in config file        : 0
LB priority                        : 0
Tag                                :
Required DS Attribute Mask         : 0x0
Forbidden DS Attribute Mask        : 0x0
Required US Attribute Mask         : 0x0
Forbidden US Attribute Mask        : 0x0
Service Type ID                    :
Service Type ID in config file     :
Active Classifiers                 : 2 (Max = NO LIMIT)
CM Upstream Filter Group           : 0
CM Downstream Filter Group         : 0
CPE Upstream Filter Group          : 0
CPE Downstream Filter Group        : 0
DSA/DSX messages                   : permit all
Voice Enabled                      : NO
CM Energy Management Capable       : Y
CM Enable Energy Management        : Y
CM Enter Energy Management         : No
Battery Mode                       : Yes
Battery Mode Status                : BATTERY_MODE / AC_POWER_MODE
DS Change Times                    : 0
Boolean Services                   : 2
Number of Multicast DSIDs Support  : 16
MDF Capability Mode                : 2
IGMP/MLD Version                   : MLDv2
FCType10 Forwarding Support        : Y
Features Bitmask                   : 0x0
Total Time Online                  : 2h12m   (2h12m    since last counter reset)
CM Initialization Reason           : NO_PRIM_SF_USCHAN
CFG Max IPv6 CPE Prefix            : 16 (-1 used)
```

Note *Battery Mode* indicates if the cable modem is in battery backup mode or AC power mode.

Battery Mode Status indicates the status of the cable modem:

- When the cable modem is in AC_POWER_MODE/BATTERY_MODE status, it is in stable state.
- When the cable modem is in AC_POWER_PENDING/BATTERY_PENDING status, it is in transfer state.
- When the cable modem is in AC_POWER_HOLD/BATTERY_HOLD status, it is updating status of the last event received until the dampen time expires.

- **show cable modem cm-status**—Displays the cable modem CM-STATUS event information.

  Following is a sample output of the command:

```
Router# show cable modem e448.c70c.9d80 cm-status

I/F       MAC Address    Event              TID   Count Error Dups  Time
C6/0/3/UB e448.c70c.9d80 Battery backup     14     1     0     0     Apr 2 22:17:29
          e448.c70c.9d80 A/C power          1      1     0     0     Apr 2 22:43:52
```

Additional References

Related Documents

Related Topic	Document Title
CMTS commands	Cisco CMTS Cable Command Reference

Standards and RFCs

Standard/RFC	Title
CM-SP- MULPIv3.1-I01-131029	Data-Over-Cable Service Interface Specifications, DOCSIS 3.1, MAC and Upper Layer Protocols Interface Specification

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

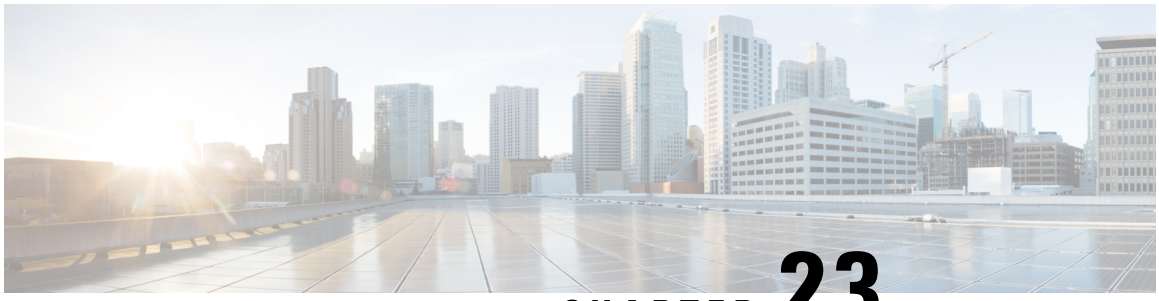
Feature Information for Downgrading Channel Bonding in Battery Backup Mode

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 67: Feature Information for Downgrading Channel Bonding in Battery Backup Mode

Feature Name	Releases	Feature Information
Battery Backup 1x1 Mode	Cisco IOS XE Everest 16.5.1	This feature was introduced in the Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 23

Upstream Bonding Support for D-PON

The DOCSIS Passive Optical Network (D-PON) architecture, also known as RF over Glass (RFoG), helps the cable operators enter fiber-to-home market space and utilize the DOCSIS infrastructure effectively.

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 439
- Prerequisites for Upstream Bonding Support for D-PON, on page 440
- Restrictions for Upstream Bonding Support for D-PON, on page 440
- Information About Upstream Bonding Support for D-PON, on page 441
- How to Configure Upstream Bonding Support for D-PON , on page 442
- Verifying the Upstream Bonding Support for D-PON, on page 444
- Additional References, on page 444
- Feature Information for Upstream Bonding Support for D-PON, on page 445

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 68: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Upstream Bonding Support for D-PON

- DOCSIS 3.0 cable modems (CMs)
- DOCSIS 2.0 capable set-top boxes (STBs)
- You should configure the **cable upstream ranging-init-technique 2** command to prevent the use of additional broadcast initial ranging opportunities for the non D-PON reference upstream channels.

Restrictions for Upstream Bonding Support for D-PON

- Multiple CMs cannot work at the same time. Each CM must get a separate time slot for upstream data transmission. Configuring D-PON ensures that the upstream scheduler allows only a single CM to transmit at a particular time.
- cBR8 does not support Upstream Bonding for D-PON on Remote-PHY.
- It is possible to enable or disable D-PON for individual Mac domain.
- When D-PON is enabled in a MAC domain, you must manually shut down and enable the MAC domain using **shutdown** and **no shutdown** command.
- All frequencies in a MAC domain must have the same configuration for:

  - minislot size
  - channel-width
  - modulation profile

- Supports only ATDMA DOCSIS mode.
- The following features are not supported in MAC domains configured for D-PON:
  - Load balancing
  - Spectrum management
  - Upstream configuration (to change upstream configuration, you should shut down the MAC domain)
  - S-CDMA logical channels
  - Lower modulations profiles (D-PON uses only 16 QAM and 64 QAM modulation profiles)
  - Channel-width other than 3.2 MHz and 6.4 MHz
  - In-service Software Upgrade (ISSU)
  - Mixing of D-PON and HFC on the same MAC domain
  - Software licensing

Information About Upstream Bonding Support for D-PON

D-PON is a type of networking that allows the CMTS to transmit RF signals over the optical network. This technology enables the cable operators to use the RF technologies in a Hybrid Fiber-Coaxial (HFC) network.

The downstream data from the CMTS is combined with other RF signals at the hub and is sent to the transmitter. The signal from the transmitter is intended for multiple PONs. Each PON serves 32 homes from a single fiber.

The upstream data from the Optical Network Terminal (ONT) is combined at the splitter and sent to the hub, which is then routed to the optical receiver (RX). Upstream data from multiple optical receivers are combined and sent to the CMTS.

The upstream data to the CMTS contains signals from multiple PONs. Each upstream optical receiver is dedicated to a PON that can have multiple DOCSIS upstream sources (multiple modems and DSG terminals).

A PON can be configured in different ways, depending on the coaxial network. These configurations broadly fall under the following categories:

- Internal Control Configuration—In this configuration, the internal CM of the ONT controls the laser. Upstream signals go to the Ethernet interface of the CM allowing the ONT to control upstream timing. The device that uses this type of configuration is the Digital Audio Visual Council (DAVIC) set top terminal (STT).
- External Control Configuration—In this configuration, the presence of the RF at the ONT upstream input activates the laser. The device that uses this type of configuration is the DOCSIS Set-top Gateway (DSG) device.

- Dual Control Configuration—In this configuration, the home that contains an ONT with and internal CM and other DOCSIS upstream sources, like DSG devices. The RF presence detector or the internal CM control line detects the upstream signal and activates the upstream laser.

D-PON on Upstream Scheduling

In D-PON implementation, the native upstream scheduling software of the CMTS controls the timing of the upstream data transmission. Only one PON Receiver Domain (PRD) is allowed to transmit upstream data at any given point of time, irrespective of the upstream frequency allocation. The reason for this is that two lasers from the ONT of a PRD cannot work simultaneously as it leads to an Optical Beat Interference (OBI). Moreover, the use of a frequency modulation (FM) for and upstream signal transmission results in PHY errors when multiple ONTs within a PON transmits simultaneously.

Initial maintenance regions are scheduled on all upstream channels to prevent the DOCSIS 3.0 CMs in a D-PON environment from failing initial ranging on any upstream channel. When a ranging request is received within a MAC domain configured for D-PON, the CM receives an upstream channel override to the D-PON reference channel (US0).

In this implementation of D-PON, a DOCSIS device within a PRD is given a timeslot to transmit the upstream data, irrespective of the upstream frequency. Therefore, there is no benefit in having more than one upstream in a MAC domain without using the upstream channel bonding feature.

The D-PON feature supports the following service types:

- Best effort (BE) 3.0 using up to four frequencies concurrently
- BE 2.0 using only a single frequency
- Unsolicited grant service (UGS) using only a single frequency
- Real-time polling service (RTPS) using only a single frequency
- Non-real-time polling service (nRTPS) using only a single frequency

How to Configure Upstream Bonding Support for D-PON

This section describes how to enable D-PON for a MAC domain on the Cisco cBR router. The bonding group must include a reference channel which is default to the upstream channel 0.

Note The USCB can support a maximum of 4 US channels in a RFOG MAC domain in the following combination:

- US0
- US0, US1
- US0, US1, US2
- US0, US1, US2, US3

All US channels must be configured to have the same number of minislots, channel width (only 3.2 and 6.4 MHz supported), DOCSIS mode (only ATDMA supported) and modulation profile.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** *slot/subslot/cable-interface-index* **Example:** `Router(config)# interface cable 8/0/0`	Enters interface configuration mode for the specified cable interface.
Step 4	**cable upstream dpon** **Example:** `Router(config-if)# cable upstream dpon`	Enables D-PON for a MAC domain.
Step 5	**shutdown** **Example:** `Router(config-if)# shutdown`	Shuts down the interface.
Step 6	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface.
Step 7	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to the privileged EXEC mode.

DOCSIS 3.0 Cable Modems Upstream Bonding Enters Partial Bonding

Sometimes, upstream bonding enters partial bonding while reverting to the original CLC after switchover.

In the following example, the switchover is performed between CLC in slot 1 and 0. After reverting to CLC in slot 1, C1/0/3/UB turns to C1/0/3/p. The following output highlights the change:

```
Router# show cable modem
Load for five secs: 1%/0%; one minute: 2%; five minutes: 2% Time source is NTP, 13:00:25.570
 UTC Thu Feb 18 2016

                                                                                  D
MAC Address    IP Address      I/F           MAC              Prim  RxPwr  Timing Num I
                                             State            Sid   (dBmv) Offset CPE P
4844.8789.e0fc 10.78.100.6     C1/0/0/UB     w-online         6     -0.50  1787   0   N
c8fb.2639.33d0 10.78.100.7     C1/0/1/UB     w-online         1     -1.00  1793   0   N
4844.8789.e10e 10.78.100.10    C1/0/2/UB     w-online         1     -0.50  1784   0   N
602a.d0a2.9b3e 10.78.100.11    C1/0/3/p   w-online          4      1.50   1789   0   N
0025.2ecf.f922 10.78.100.8     C1/0/4/UB     w-online         1     -1.00  1788   0   N
```

This partial bonding can be recovered using the **shutdown** and **no shutdown** commands on the related upstream port under the controller. The following is a sample configuration:

```
Router(config)# controller Upstream-Cable 1/0/3
Router(config-controller)# us-channel 0 shutdown
Router(config-controller)# no us-channel 0 shutdown
```

Verifying the Upstream Bonding Support for D-PON

To verify the upstream scheduler output for a MAC domain configured with D-PON, use the **show interface cable mac-scheduler** command.

Note The D-PON reference channel (US-1) MAP serves as a template for producing other MAPs within the MAC domain. Therefore, some of the statistics related to upstream scheduling is not relevant for other channels, except for the D-PON reference channel.

```
Router# show interface cable 8/0/0 mac-scheduler 1
     DOCSIS 1.1 MAC scheduler for Cable8/0/0/U1 : rate 30720000
     wfq:None
     us_balance:OFF
     dpon_mode:ON
     fairness:OFF
     Queue[Rng Polls] flows 0
     Queue[CIR Grants] flows 0
     Queue[BE(07) Grants] flows 0
     Queue[BE(06) Grants] flows 0
     Queue[BE(05) Grants] flows 0
     Queue[BE(04) Grants] flows 0
     Queue[BE(03) Grants] flows 0
     Queue[BE(02) Grants] flows 0
     Queue[BE(01) Grants] flows 0
     Queue[BE(00) Grants] flows 0
     Req Slots 1824595508, Req/Data Slots 10640906
     Init Mtn Slots 89924653, Stn Mtn Slots 989543
     Short Grant Slots 0, Long Grant Slots 0
     Adv Phy Short Grant Slots 538, Adv Phy Long Grant Slots 219831
     Adv Phy UGS Grant Slots 0
     Avg upstream channel utilization : 0%
     Avg percent contention slots : 98%
     Avg percent initial ranging slots : 1%
     Avg percent minislots lost on late MAPs : 0%
     MAP TSS: lch_state 9, init_retries 0
     late_initial_maps 0, late_ucd_maps 0
     mac-phy tss errors 0, missed ccc 0
```

Additional References

The following sections provide references related to the Upstream Bonding Support for D-PON feature.

Related Documents

Related Topic	Document Title
Prisma D-PON	Cisco Prisma D-PON

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

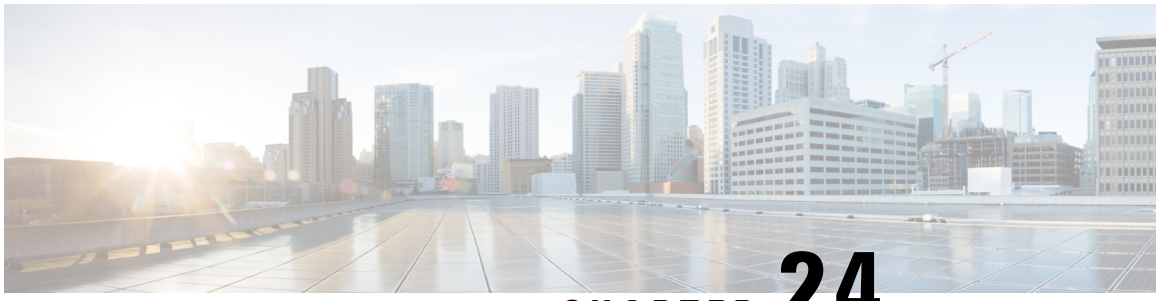
Feature Information for Upstream Bonding Support for D-PON

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 69: Feature Information for Upstream Bonding Support for D-PON

Feature Name	Releases	Feature Information
Upstream Bonding Support for D-PON	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Routers.



CHAPTER 24

Energy Management Mode

Data-over-Cable Service Interface Specifications (DOCSIS) cable modems (CM) and CMTS support a low power energy mode referred to as the Energy Management 1x1 (EM) mode. During idle times, when the data rate demand of a user is met by the available capacity on a single upstream and downstream channel pair to which it is assigned, the CM switches to the Energy Management 1x1 mode. When the CM requires a higher data rate than that can be reliably provided on the single channel pair, the CMTS instructs the CM to return to the larger transmit and receive channel set.

Contents

- Information About Energy Management Mode, on page 447
- Prerequisites for Energy Management Mode, on page 451
- Restrictions for the Energy Management Mode, on page 451
- How to Configure the Energy Management Mode, on page 454
- Verifying the Energy Management Mode, on page 455
- Feature Information for Energy Management Mode, on page 458

Information About Energy Management Mode

The following sections provide more information about the Energy Management mode.

Dynamic Downstream Bonding Group

To support the Energy Management 1x1 (EM) mode feature, CMTS selects an upstream and a downstream channel pair for CM. The downstream and upstream channel assigned to the CMs should be available. If CMTS selects a channel that is not available, the downstream bonding channel might fail.

To simplify the process, CMTS chooses the 1x1 bonding group for the CM according to the following rules:

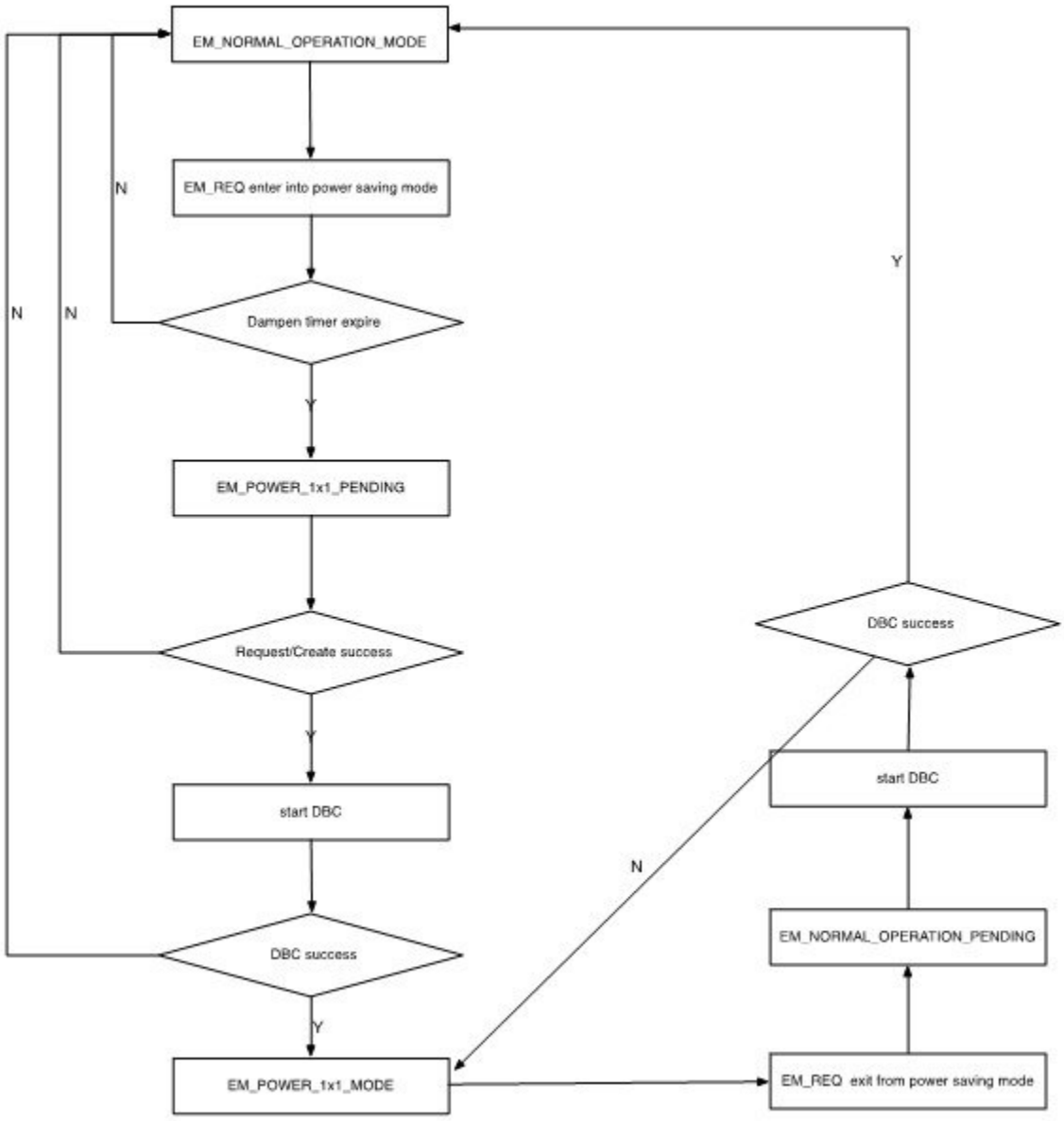
- For upstream, it chooses the highest actual amiable bandwidth channel from the upstream channels currently used by the CM.
- For downstream, CMTS chooses the current primary downstream channel used by the CM.
- If the CM is online with channel bonding 1xN or Nx1, and requests to enter into the EM mode, CMTS does not change the upstream and the downstream channel if the original channel bonding is 1 and the Quality of Service (QoS) parameter is not updated.

- CMTS checks the existing dynamic bonding groups (DBG), for an exact match in the target channel.
  - If found, CMTS uses this bonding group to instruct the CM to enter into EM mode.
  - If there is no available DBG and there is an unused DBG, CMTS adds the primary channel into the unused DBG and instructs the CM to enter the EM mode.
  - If there is no available DBG and no unused DBG, CMTS logs a warning to notify you that a new DGB should be configured.

Flow Chart of the CM Power State

The following figure shows the flow chart of the CM power state:

Figure 20: Flow Chart for the Power State of the Cable Modem



Interaction with the Battery Mode

Energy management mode is similar to battery mode as they both enter the 1x1 mode to save power. But, both have a different purpose to enter the 1x1 mode, so there are some differences in their behavior. The purpose of EM mode is to save power when the traffic is low, and it has minimum impact on the normal service. The purpose of the battery mode is try to guarantee the voice service, especially the 911 call service, for which it may drop other services, if necessary.

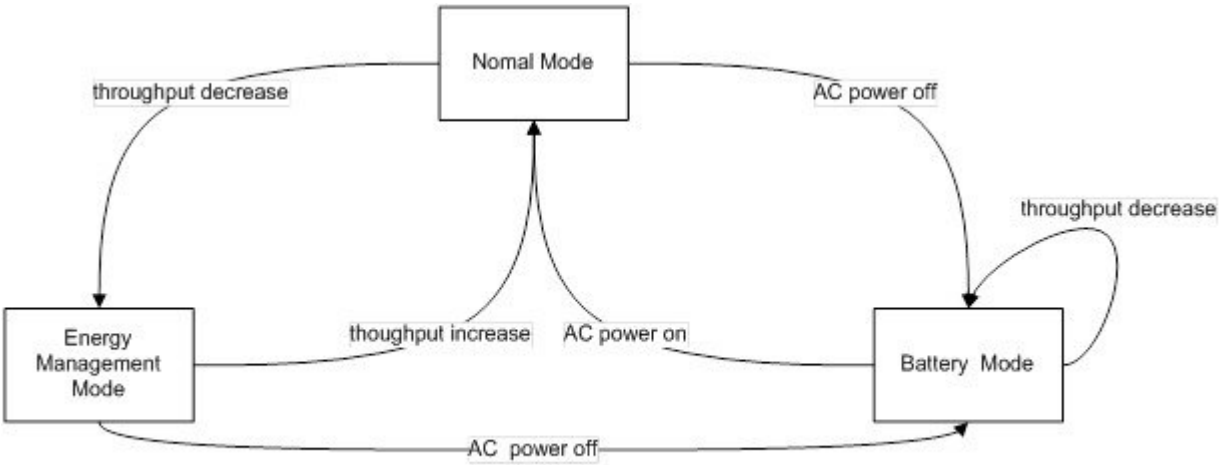
The table below describes the behavior difference between energy management mode and battery mode.

	Energy Management (EM) Mode	Battery Mode (BM)
QoS	Minimum reserved rate service up to 200k.	No minimum reserved rate.
Multicast	• IGMP request is guaranteed. • If the CM has joined the Internet Group Management Protocol (IGMP) groups, CMTS rejects the EM request to enter the EM mode and keeps the CM in the IGMP groups. • If the CM is in the EM mode and IGMP join request is received, CMTS instructs the CM to leave the EM mode.	• If the CM has joined IGMP groups and CMTS receives a CM-STATUS event with code 9, CMTS lets the CM leave the IGMP group. • If the CM is in the BM, CMTS rejects the IGMP join request from the CM.

BM has higher priority than the EM mode. If a CM is already in EM mode and a power off occurs, CM enters into the BM. After the power is restored, the CM returns to the normal mode, and if the traffic is lower than the threshold, it re-enters the EM mode. The CM does not directly transfer from the BM to the EM mode.

The interaction between the battery mode and the energy management mode is illustrated in the figure below:

Figure 21: Interaction Between the BM and the EM Modes



1. When the CM is in normal mode and CMTS receives a request to enter the EM mode, CMTS instructs the CM to enter the EM mode with downstream bonding channel (DBC).
2. When the CM is in EM mode and CMTS receives a request to leave the EM mode, CMTS instructs the CM to leave the EM mode to normal mode with DBC.
3. When the CM is in normal mode and CMTS receives a message: CM operating on battery backup, CMTS instructs the CM to enter the BM mode with DBC.
4. When the CM is in BM mode and CMTS receives a message: CM operating on AC power, CMTS instructs the CM to leave the BM mode to normal mode with DBC.

5. When the CM is in EM mode and CMTS receives a message: CM operating on battery backup, CMTS instructs the CM to enter the BM mode with service flow re-admin.
6. When the CM is in BM mode and CMTS receives a request to enter EM mode, CMTS waits until it receives the message: CM operating on AC power. It then instructs the CM to return to normal mode.

Handling Energy Management Request Load

When many CMs send EM requests at the same time, such as at the beginning or end of work hours, the traffic soars or slumps in a very short period and causes heavy load for CMTS. A throttle mechanism is adopted to avoid such load surge for CMTS.

The line card EM process defines a variable that indicates the current transactions handled by the process. When an energy management request is received and the maximum number of transactions is not met, CMTS handles this request and updates the counter of current transactions. When the maximum number of transactions is met, CMTS sends a temporary reject response. After a transaction is over or a CM goes offline, the counter of current transactions is updated.

Supervisor High Availability and Line Card Switchover

Energy Management feature supports supervisor high availability and line card switchover with limitations.

The active supervisor or line card syncs the EM mode data of a CM to the standby SUP or protected line card when the CM enters a stable EM status. When a CM enters or leaves the EM mode with an ongoing DBC process, the supervisor high availability or line card switchover causes the CM to enter into an offline or online status.

Prerequisites for Energy Management Mode

To enable the energy management mode, you must configure resiliency bonding group (RBG) and dynamic bonding group.

Restrictions for the Energy Management Mode

Restrictions for CMTS High Availability

- If there is no DBG available, CMTS cannot create a new DBG on the protected line card and CMs cannot enter the EM mode.
- Line card switchover is not supported when the CM enters the EM mode from the normal mode or exits the EM mode to the normal mode.
- To reduce the operation of the EM mode, the information about the EM status is not synced with the protected line card. Hence, the EM status is cleared after the line card high availability.

Restrictions for Dynamic Bonding Group

To support the EM feature, CMTS configures separate DBG for each primary channel in each MAC domain. For example, if a MAC domain has eight primary channels, it will create eight DBGs for the MAC domain. This ensures that the EM does not fail due to lack of DBGs.

Restrictions for Interaction of CMTS with Other Features

The following sections describe the restrictions for CMTS interaction with other features.

Voice

If a voice call is in progress, CMTS does not instruct the CM to enter into the EM mode.

When the CM is in the EM mode, and it receives a voice call, it adds a dynamic Unsolicited Grant Service (UGS) or Unsolicited Grant Service with activity detection (UGS-AD) service flow. During the voice call, the CM does not exit from the EM mode irrespective of the flow of traffic. Voice service is given the highest priority

Dynamic Bonding Change and Dynamic Channel Change and Related Applications

In D2.0 and D3.0 load-balance (static and dynamic), CM is not moved by load-balance when it is in EM mode.

For RF-adapt, CM is not relocated to an alternate logical channel by the RF-adapt when it is in EM mode.

Multicast

- When the CM is in a multicast group, CMTS would reject the EM request for both bonded and non bonded multicast cases.
- When the CM is in EM state and a multicast join request is received, CMTS discards this join request and forces the CM to exit the EM mode.
- When the CM is in EM state and a voice call is in progress, and a new multicast join request is received, CMTS discards this join request and does not force the CM to exit the EM mode since the voice call is in progress.
- There is a threshold for currently handled transactions. When there is multicast join request and the maximum transaction threshold has been reached, CMTS cannot instruct the CM to exit the EM mode. The multicast join is also be denied until the CM can exit the EM mode.
- When the CM is in EM mode and needs to join PacketCable Multimedia (PCMM) multicast, you should send a GateSet request twice, so that the gate can be setup successfully. The first GateSet request only forces the modem to exit the EM mode, but does not set up the gate

Committed Information Rate

If the QoS is defined by the Minimum Reserved Rate service flow QoS parameter in excess of 200 kbps, when the CM enters into the EM mode, CMTS only provides 200 kbps as the minimum reserved rate. If the minimum reserved rates is less than 200 kbps, CMTS schedules the minimum reserved rate according to the service flow configuration when the CM enters into the EM mode.

CMTS records the Minimum Reserved Rate service flow QoS parameter when the CM enters into the EM mode. When the CM exits the EM mode, CMTS uses the original parameter.

When the CM enters the EM mode, it selects one of the CM upstream channels. If the service flow is completely on that upstream channel, the service flow parameter is not changed. This behavior is because the service flow is not moved into the DBC operation, and the change of the service flow parameter has no benefit

Admission Control

When a request is received to exit the EM mode and recovery to the original wideband interface is restricted due to an admission control failure, CMTS forces the CM to go offline and re-register to prevent it from getting stuck in the EM mode. In such a case, CMTS logs a warning message.

Battery Mode

When CMTS receives the status of the CM as operating on battery power, CMTS instructs the CM to enter into the BM. If the CM rejects the instruction received, CMTS keeps the modem in normal status.

When the CM is in BM and CMTS receives the status of the CM as operating on A/C power, CMTS instructs the CM to exit the BM. If the CM rejects the instruction received, CMTS forces the CM to go offline to prevent it from getting stuck in the battery mode. In such a case, CMTS logs a warning message.

Attribute Mask

When selecting an upstream or a downstream channel pair for energy management mode, CMTS selects channels that meet the requirements of the attribute masks for the existing service flows for the corresponding CM.

In some cases, adherence to the service flow attribute-based assignment may not be possible when selecting an upstream and downstream channel pair for energy management mode of operation. To resolve this conflict, CMTS supports one or both of the following approaches:

1. CMTS may require strict adherence to the required and forbidden attribute masks and thus deny entry into the EM mode if these masks cannot be met by the available individual channels in the MD-CM-SG.
2. CMTS may allow the CM to enter the EM mode while not meeting all the criteria for the attribute masks. In this case, CMTS logs a warning event notifying that the attribute masks are not maintained.

For the following case, CMTS supports approach two:

When the CM is instructed to enter into the EM mode and the selected target upstream and downstream channels do not adhere to the service flow attribute mask. For this conflict CMTS instructs the modem to enter into the EM mode. CMTS also logs a warning message to notify this conflict.

Dynamic Service Addition

When the CM is in EM mode, DSA request can still be set up even if the requested attributes can not be met with a single channel. In order to not effect voice services, the CM is not forced to exit the EM mode.

Restrictions for Configuration Change and Interface Shutdown

1. **Shutdown of the upstream channel**—Shutdown of the upstream channel recalculates the MD-US-SG-ID and assigns a new MD-US-SG-ID. In this case, the CM is not offline and internal data structure of the CM instance is not updated. DBC operation checks the MD-US-SG-ID, so when the CM enters into the EM mode there is a mismatch between the MD-US-SD-ID on the CM and the new MD-US-SG-ID. Hence, DBC fails and the CM cannot get into the EM mode.

2. **Change in Upstream Service Group makes the CM in EM mode go offline**—The US-SG configuration change blocks the DBC behavior and the CM gets stuck in the EM mode. To avoid this scenario, when there is a change in the Upstream Service Group (US-SG), such as shutdown or no shutdown of the upstream channels, CMTS makes the CM go offline. The CM should re-register as a normal CM with the wideband channel bonding including multiple channels.
3. **Modify the original wideband interface**—When the CM is in EM mode, change in the original wideband interface channels on the CM makes the CM go offline and re-register as a normal CM.
4. **Disable or enable feature**— When you disable this feature, CMTS does not force CMs to exit from the EM mode unless CMs sends a request. CMTS does not accept EM requests after the EM feature is disabled from the CLI.

How to Configure the Energy Management Mode

This section describes how to configure the energy management feature on the Cisco cBR-8.

Contents

Enabling Energy Management Mode

To enable the energy management mode, complete the following procedure:

```
configure terminal
cable reduction-mode energy-management enable
```

Verifying the Energy Management Mode

- To verify if the CM is in EM mode, use the **show cable modem** command. If the cable modem is working in energy management mode, the MAC state is displayed with an "em" flag.

```
show cable modem

                                                                D
MAC Address    IP Address      I/F                 MAC                Prim  RxPwr
 Timing Num   I
                                                                          State
         Sid   (dBmv) Offset  CPE    P
7cb2.1b0f.ea72 40.4.58.4       C7/0/0/UB     w-online(em)               2      0.00
    1231     1         Y
54d4.6ffb.2f6b 40.4.58.24      C7/0/0/UB     w-online                   3     -0.50
   1241     0        Y
0025.2ed9.9a22 40.4.58.3       C7/0/0/UB     w-online                   4      0.50
   1240     0        Y
```

- To verify which CM is in EM mode and to get the original wideband and upstream channel information, use the **show cable modem reduction-mode energy-management-mode** command.

```
show cable modem reduction-mode energy-management-mode

                              Orig BG     Orig US                   Curr BG
I/F     MAC Address    ID      I/F        bitmap   RFs     ID          I/F
Upstream
------- -------------- ---------------- -------- ----------------------------------------
C7/0/0  0025.2eaf.843e 897  Wi7/0/0:0     0x3B     4      252        Wi7/0/0:1      US0
```

```
C7/0/0  0025.2eaf.8356 897  Wi7/0/0:0     0x3B     4      252        Wi7/0/0:1      US0

C7/0/0  0015.d176.5199 897  Wi7/0/0:0     0x3B     4      252        Wi7/0/0:1      US0
```

Enabling Energy Management Mode per MAC Domain

CMTS supports the EM feature when enabled both globally and per MAC domain. Use the following procedure to enable energy management feature per MAC domain.

To enable the EM mode per MAC domain, complete the following procedure:

```
configure terminal
interface cable slot/subslot/cable-interface-index

cable reduction-mode energy-management enable
```

Configuring Initialization Ranging Technique in Dynamic Bonding Channel

The default value for the technique in init-ranging is set to 1 and the valid range is 1-4.

To configure the technique in init-ranging, complete the following procedure:

```
configure terminal
cable reduction-mode energy-management ranging-init-techniquevalue
```

Configuring the Percentage for the Dynamic Channel Bandwidth

Make sure that you leave enough bandwith for the primary channel so that it can allocate dynamic channel bandwidth when it joins to a newly created DBG. The default percentage value is set to 5 and the valid range is 1-96.

To configure the percentage of dynamic channel bandwidth, complete the following procedure:

```
configure terminal
cable reduction-mode energy-management dynamic-channel-percent value
```

Configuring the Queue Size for Energy Management

The default value for the queue size is set to 150 and the valid range is 50-10000.

To set the queue size of the energy management requests, complete the following procedure:

```
configure terminal
cable reduction-mode energy-management process-queue-size value

-
```

Verifying the Energy Management Mode

This section describes how to verify the EM mode.

Contents

Viewing the Basic Statistics for Energy Management Receive Request

To view the basic statistics for all energy management receive request events for a specific CM, use the **show cable modem <cable if | mac_addr | ip_addr> reduction-mode energy-management-status** command.

```
show cable modem c8/0/0 reduction-mode energy-management-status
```

```
I/F    MAC Address    Event          TID Count Error Dups   Time
C8/0/0 54d4.6ffb.2e21 Enter EM mode   1   1     0     1    Jul 16 21:38:18
                      Exit EM mode    1   1     0     0    Jul 16 21:38:39
C8/0/0 602a.d07c.4ec6 Enter EM mode   1   1     0     0    Jul 16 21:40:57
                      Exit EM mode    1   1     0     0    Jul 16 21:41:17
```

To clear the basic receive statistics for all EM_REQ events for a specified CM, use the **clear cable modem <cable if | mac_addr | ip_addr> em-status** command.

Verifying the Configuration Parameters

To verify the configuration parameters used in the CM configuration file, use the **show cable modem <mac address> reduction-mode energy-management-param** command.

```
show cable modem 54d4.6ffb.2e21 reduction-mode energy-management-param
```

```
Energy Management feature enable : Y
DS entry bitrate threshold(bps)  : 100000
DS entry time threshold(s)       : 120
DS exit bitrate threshold(bps)   : 200000
DS exit time threshold(s)        : 2
US entry bitrate threshold(bps)  : 100000
US entry time threshold(s)       : 120
US exit bitrate threshold(bps)   : 200000
US exit time threshold(s)        : 2
cycle period(s)                  : 300
```

Viewing Information Regarding a Cable Modem

To view all the information regarding a CM, use the **show cable modem** *mac address* **verbose** command.

```
show cable modem 54d4.6ffb.30fd verbose
```

```
MAC Address                       : 54d4.6ffb.30fd
IP Address                        : 40.4.58.14
IPv6 Address                      : 2001:40:4:58:741A:408D:7E4B:D7C8
Dual IP                           : Y
Prim Sid                          : 9
Host Interface                    : C7/0/0/UB
MD-DS-SG / MD-US-SG               : 1 / 1
MD-CM-SG                          : 0x3C0101
Primary Wideband Channel ID       : 897 (Wi7/0/0:0)
Primary Downstream                : In7/0/0:2 (RfId : 722)
Wideband Capable                  : Y
RCP Index                         : 3
RCP ID                            : 00 10 00 00 08
Downstream Channel DCID RF Channel : 99    7/0/0:2
Downstream Channel DCID RF Channel : 97    7/0/0:0
```

```
Downstream Channel DCID RF Channel : 98    7/0/0:1
Downstream Channel DCID RF Channel : 100   7/0/0:3
Multi-Transmit Channel Mode        : Y
Extended Upstream Transmit Power   : 0dB
Upstream Channel                   : US0        US1
Ranging Status                     : sta        sta
Upstream SNR (dB)                  : 36.12      32.55
Upstream Data SNR (dB)             : --         --
Received Power (dBmV)              : 0.00       0.00
Reported Transmit Power (dBmV)     : 25.25      26.00
Peak Transmit Power (dBmV)         : 54.00      54.00
Phy Max Power (dBmV)               : 54.00      54.00
Minimum Transmit Power (dBmV)      : 24.00      24.00
Timing Offset             (97.6 ns): 1226       1226
Initial Timing Offset              : 1229       973
Rng Timing Adj Moving Avg(0.381 ns): -1         0
Rng Timing Adj Lt Moving Avg       : -7         0
Rng Timing Adj Minimum             : -768       0
Rng Timing Adj Maximum             : 0          64768
Pre-EQ Good                        : 0          0
Pre-EQ Scaled                      : 0          0
Pre-EQ Impulse                     : 0          0
Pre-EQ Direct Loads                : 0          0
Good Codewords rx                  : 515        472
Corrected Codewords rx             : 0          0
Uncorrectable Codewords rx         : 0          0
Phy Operating Mode                 : atdma*     atdma*
sysDescr                           :
Downstream Power                   : 0.00 dBmV (SNR = ----- dB)
MAC Version                        : DOC3.0
QoS Provisioned Mode               : DOC1.1
Enable DOCSIS2.0 Mode              : Y
Modem Status                       : {Modem= w-online(em), Security=disabled}
Capabilities                       : {Frag=N, Concat=N, PHS=Y}
Security Capabilities              : {Priv=, EAE=Y, Key_len=}
L2VPN Capabilities                 : {L2VPN=N, eSAFE=N}
Sid/Said Limit                     : {Max US Sids=16, Max DS Saids=15}
Optional Filtering Support         : {802.1P=N, 802.1Q=N, DUT=N}
Transmit Equalizer Support         : {Taps/Symbol= 1, Num of Taps= 24}
Number of CPE IPs                  : 0(Max CPE IPs = 16)
CFG Max-CPE                        : 200
Flaps                              : 0()
Errors                             : 0 CRCs, 0 HCSes
Stn Mtn Failures                   : 0 aborts, 0 exhausted
Total US Flows                     : 1(1 active)
Total DS Flows                     : 1(1 active)
Total US Data                      : 7 packets, 2006 bytes
Total US Throughput                : 0 bits/sec, 0 packets/sec
Total DS Data                      : 5 packets, 1202 bytes
Total DS Throughput                : 0 bits/sec, 0 packets/sec
LB group ID assigned (index)       : 2151416065 (48131)
LB group ID in config file (index) : N/A (N/A)
LB policy ID                       : 0
LB policy ID in config file        : 0
LB priority                        : 0
Tag                                :
Required DS Attribute Mask         : 0x0
Forbidden DS Attribute Mask        : 0x0
Required US Attribute Mask         : 0x0
Forbidden US Attribute Mask        : 0x0
Service Type ID                    :
Service Type ID in config file     :
Active Classifiers                 : 2 (Max = NO LIMIT)
CM Upstream Filter Group           : 0
```

```
CM Downstream Filter Group          : 0
CPE Upstream Filter Group           : 0
CPE Downstream Filter Group         : 0
DSA/DSX messages                    : permit all
Voice Enabled                       : NO
DS Change Times                     : 0
Boolean Services                    : 2
CM Energy Management Capable        : Y
CM Enable Energy Management         : Y
CM Enter Energy Management          : YES
Number of Multicast DSIDs Support   : 16
MDF Capability Mode                 : 2
IGMP/MLD Version                    : MLDv2
FCType10 Forwarding Support         : Y
Features Bitmask                    : 0x0
Total Time Online                   : 2h12m   (2h12m    since last counter reset)
CM Initialization Reason            : NO_PRIM_SF_USCHAN
CFG Max IPv6 CPE Prefix             : 16 (-1 used)
```

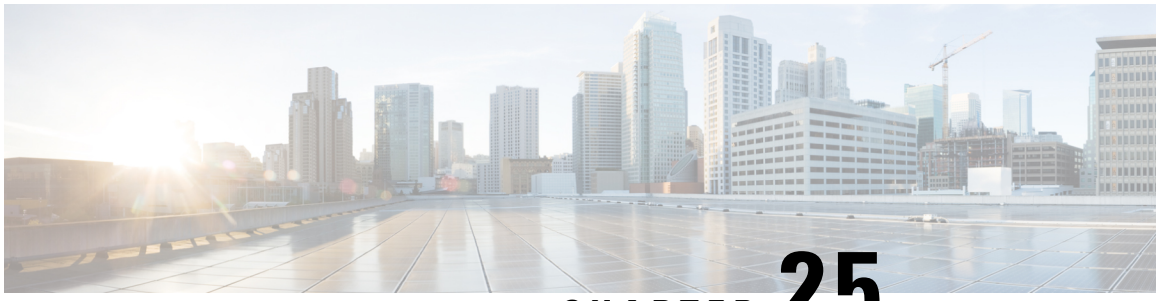
Feature Information for Energy Management Mode

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 70: Feature Information for Downgrading Channel Bonding in Battery Backup Mode

Feature Name	Releases	Feature Information
Energy Management Mode	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Routers.



CHAPTER 25

Cable Modem Steering

The cable modem steering feature helps to redirect or steer cable modems to multiple CMTS routers. A configurable string is used to bond the cable modem to the proper CMTS. Once the bonding is done, the CMTS can move the cable modem within itself for load balancing.

- Cable Modem Steering on the Cisco cBR Series Converged Broadband Routers, on page 459
- Prerequisites for Cable Modem Steering, on page 460
- Restrictions for Cable Modem Steering, on page 460
- Information About Cable Modem Steering, on page 460
- How to Configure Cable Modem Steering on the CMTS Router, on page 462
- Verifying and Troubleshooting Cable Modem Steering, on page 464
- Additional References, on page 466
- Feature Information for Cable Modem Steering , on page 467

Cable Modem Steering on the Cisco cBR Series Converged Broadband Routers

First Published: July 13, 2016

The cable modem steering feature helps to redirect or steer cable modems to multiple CMTS routers. A configurable string is used to bond the cable modem to the proper CMTS. Once the bonding is done, the CMTS can move the cable modem within itself for load balancing.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

Prerequisites for Cable Modem Steering

DOCSIS 3.0-defined type, length, values (TLVs) are required to aid channel selection. All TLVs encoded as general extension information in cable modem configuration files are backward compatible with DOCSIS 1.1 and DOCSIS 2.0 cable modems.

Note The hardware components introduced in a given Cisco IOS Release are supported in all subsequent releases unless otherwise specified.

Restrictions for Cable Modem Steering

The Cable Modem Steering feature has the following general restrictions:

- To restrict the cable modem on the exact downstream on the target CMTS, the redirection must be configured on the target CMTS.
- You must manually execute the **clear cable modem service-type** command to clear the cable modem service type identifier stored at the CMTS during registration.
- You must manually execute the **clear cable modem attribute-masks** command to clear the cable modem attribute masks stored at the CMTS during registration. These attribute masks are used to restrict usage of upstream channels during ranging time.
- Cable modem steering supports only upstream masks.
- Channel steering does not take place when the cable modem fails in initial ranging.
- The cable modem will take more time to come online when channel steering is deployed. The time taken is proportional to the number of modems, and the downstreams and upstreams that the cable modem can reach.
- A modem cannot be load balanced to any upstream channel with attributes that conflict with attribute masks of the modem.
- Do not configure to use UCC in DOCSIS LBG.

Information About Cable Modem Steering

Cable modem steering allows you to redirect or steer the cable modems to one or more CMTS routers using downstream frequency overrides. Once a cable modem registers on a downstream on the proper CMTS router, the CMTS router can move the cable modem to any location for load balancing.

The DOCSIS 3.0-compliant Service Type Identifier used as the configurable string in the cable modem configuration file is backward-compatible with DOCSIS 1.1 and DOCSIS 2.0 cable modems.

The CMTS router can also impose restrictions on the number of channels a cable modem can use. DOCSIS 3.0 defines several TLVs to aid the channel selection.

The following TLVs are used in cable modem steering:

- TLV 43.9 (Cable Modem Attribute Masks) limits the set of channels the CMTS router can assign to the cable modem by allowing or forbidding certain binary attributes. The cable modem attribute masks have four sub-TLVs and cable modem steering makes use of two sub-TLVs, which are listed below:
  - TLV 43.9.3—Cable Modem Upstream Required Attribute Mask (C.1.1.18.1.8.3 of CM-SP-MULPIv3.0-I07-080215). It is a 32-bit mask representing the set of binary upstream channel attributes required for the cable modem.
  - TLV 43.9.4—Cable Modem Upstream Forbidden Attribute Mask (C.1.1.18.1.8.4 of CM-SP-MULPIv3.0-I07-080215). It is a 32-bit mask representing the set of binary upstream channel attributes forbidden for the cable modem.
- TLV 43.11 is used for a redirection action based on the service type identifier field. The cable modem sends the TLV 43.11 in the REG-REQ MAC message. The DOCSIS 1.1 and DOCSIS 2.0 modems will also send this file ID when doing the registration.
- TLV43.1, defined as Policy ID in DOCSIS 2.0 and DOCSIS 3.0, is parsed and stored in the cable modem during registration. Before moving the cable modem during load balancing (LB), the CMTS router checks whether the cable modem has a preconfigured policy with the same Policy ID. If the policy does exist, the CMTS router disables LB for this cable modem and moves to the next cable modem. If the policy does not exist on the CMTS router, or the Policy ID is missing from the cable modem configuration file, LB prohibition is not performed.

The following TLVs are supported in cable modem steering:

- TLV 43.11 (Service type identifier) from section C.1.1.18.1.10 in CM-SP-MULPIv3.0-I07-080215.
- Cable modem attribute masks (TLV 43.9) from C.1.1.18.1.8.3 and C.1.1.18.1.8.4 of CM-SP-MULPIv3.0-I07-080215.
- TLV portion (43.1, Policy ID) of REQ-REQ
- TLV 18--- Ranging Hold-off Priority Field
- TLV 19---Channel Class ID

Upstream Channel Descriptor TLV for Ranging Hold-off

The Upstream Channel Descriptor (UCD) TLV for Ranging Hold-off feature, enables the Cisco cBR Series Converged Broadband router to hold off a cable modem from initial ranging based on TLV 18 and 19 specified in the upstream channel descriptor (UCD) messages. The router can hold off a cable modem from initial ranging only for 5 minutes. This default value cannot be changed. This feature is supported with DOCSIS 2.0 and later releases cable modems using upstream logical channels.

Ranging Class ID

The CMTS enables UCD TLV for ranging hold-off after detecting the TLVs from the cable modem registration request (REG-REQ) or multipart registration request (REG-REQ-MP), and saves these TLVs as a cable modem ranging class ID.

By default, DOCSIS load balance is supported for all cable modems with all types of ranging class IDs. In the event of DOCSIS load balance, a cable modem moves to the target upstream channel only if the ranging class ID matches with the upstream channel class ID.

Cable Modem Exclusion for DOCSIS Load Balance

You can exclude a cable modem or a group of cable modems from DOCSIS load balance based on their device type, MAC address, and Organizational Unique Identifier (OUI) using the **cable load-balance exclude** command in global configuration mode.

How to Configure Cable Modem Steering on the CMTS Router

This section describes the following required and optional procedures:

Configuring an Upstream Channel Class ID

This configuration is optional. A channel class ID must be configured for an upstream logical channel if you want to configure UCD TLV ranging hold-off on the CMTS router.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**controller upstream-cable** *slot/subslot/controller-port-number* **Example:** `Router(config)# controller Upstream-Cable 3/0/0`	Specifies the cable interface and enters cable interface configuration mode. Arguments for this command may vary depending on the CMTS router, line card, and Cisco IOS software release. For details, see the Cisco IOS CMTS Cable Command Reference . • *Slot*— Slot where the interface card resides. The valid range is from 0 to 3 and 6 to 9 on the Cisco cBR-8 router. • *Subslot*—Subslot where the interface card resides. The valid value is 0 on the Cisco cBR-8 router. • *controller-port-number*—Controller port number. The valid values are from 0 to 7 on the Cisco cBR-8 router.
Step 4	**us-channel** *us-channel-id* **chan-class-id** *id* **Example:** `Router(config-controller)# us-channel 3 chan-class-id ff`	Configures the channel class ID for an upstream logical channel. • *us-channel-id*—Specifies the upstream channel id. • *id*—Channel class ID for the logical upstream channel in the hexadecimal format. The valid range is from 0 to ffffffff. The default value is 0.

	Command or Action	Purpose
Step 5	**end** **Example:** `Router(config-controller)# end`	Returns to privileged EXEC mode.

Configuring an Upstream Ranging Hold-off Priority Value

This configuration is optional

Maximum time that a cable modem can inhibit transmissions on an upstream channel in response to its ranging class ID matching a bit value in the Ranging Hold-off Priority field in the cable modem configuration file is 300 seconds (five minutes) per *DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification* . This default timer value cannot be changed.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**controller upstream-cable** *slot/subslot/controller-port-number* **Example:** `Router(config)# controller upstream-cable 3/0/0`	Specifies the cable interface and enters cable interface configuration mode. Arguments for this command may vary depending on the CMTS router, line card, and Cisco IOS software release. For details, see the Cisco IOS CMTS Cable Command Reference . • *Slot*—Slot where the interface card resides. The valid range is from 0 to 3 and 6 to 9 on the Cisco cBR-8 router. • *Subslot*—Subslot where the interface card resides. The valid value is 0 on the Cisco cBR-8 router. • *controller-port-number*—Controller port number. The valid values are from 0 to 7 on the Cisco cBR-8 router.
Step 4	**us-channel** *us-channel-id* **rng-holdoff** *priority* **Example:** `Router(config-controller)# us-channel 3 rng-holdoff 1`	Configures the ranging hold-off priority value for an upstream logical channel. • *us-channel-id*—Specifies the upstream channel id.

	Command or Action	Purpose
		• **rng-holdoff** *priority*—Specifies the ranging hold-off priority value in the hexadecimal format. The valid range is from 0 to ffffffff. The default value is 0.
Step 5	**end** **Example:** `Router(config-controller)# end`	Returns to privileged EXEC mode.

Verifying and Troubleshooting Cable Modem Steering

This section provides the verification and troubleshooting information:

Verifying an Upstream Ranging Class ID Configuration

To verify an upstream ranging class ID of a cable modem, use the **show cable modem** command with the **verbose** keyword.

Following is a sample output of the **show cable modem verbose** command:

```
Router# show cable modem 68b6.fcfe.22e5 verbose

MAC Address                         : 68b6.fcfe.22e5
IP Address                          : 192.168.0.8
IPv6 Address                        : 2001:DB8:10:1:9951:1972:33F9:9867
Dual IP                             : Y
Prim Sid                            : 8
Host Interface                      : C8/0/0/UB
MD-DS-SG / MD-US-SG                 : 1 / 2
MD-CM-SG                            : 0x5A0102
Primary Wideband Channel ID         : 3073 (Wi8/0/0:0)
Primary Downstream                  : Mo8/0/0:0 (RfId : 2304)
Wideband Capable                    : Y
RCP Index                           : 3
RCP ID                              : 00 10 00 00 18
Downstream Channel DCID RF Channel : 45    8/0/0:0
Downstream Channel DCID RF Channel : 46    8/0/0:1
Downstream Channel DCID RF Channel : 47    8/0/0:2
Downstream Channel DCID RF Channel : 48    8/0/0:3
Downstream Channel DCID RF Channel : 49    8/0/0:4
Downstream Channel DCID RF Channel : 50    8/0/0:5
Downstream Channel DCID RF Channel : 51    8/0/0:6
Downstream Channel DCID RF Channel : 52    8/0/0:7
Downstream Channel DCID RF Channel : 53    8/0/0:8
Downstream Channel DCID RF Channel : 54    8/0/0:9
Downstream Channel DCID RF Channel : 55    8/0/0:10
Downstream Channel DCID RF Channel : 56    8/0/0:11
Downstream Channel DCID RF Channel : 57    8/0/0:12
Downstream Channel DCID RF Channel : 58    8/0/0:13
Downstream Channel DCID RF Channel : 59    8/0/0:14
Downstream Channel DCID RF Channel : 60    8/0/0:15
Downstream Channel DCID RF Channel : 61    8/0/0:16
Downstream Channel DCID RF Channel : 62    8/0/0:17
Downstream Channel DCID RF Channel : 63    8/0/0:18
Downstream Channel DCID RF Channel : 64    8/0/0:19
```

```
Downstream Channel DCID RF Channel : 65    8/0/0:20
Downstream Channel DCID RF Channel : 66    8/0/0:21
Downstream Channel DCID RF Channel : 67    8/0/0:22
Downstream Channel DCID RF Channel : 68    8/0/0:23
UDC Enabled                        : N
Extended Upstream Transmit Power   : 61dB
Multi-Transmit Channel Mode        : Y
Number of US in UBG                : 8
Upstream Channel                   : US0        US1        US2        US3
Ranging Status                     : sta        sta        sta        sta
Upstream SNR (dB)                  : 30.62      32.32      18.25      24.26
Upstream Data SNR (dB)             : --         --         --         --
Received Power (dBmV)              : 0.50       0.00       -0.50      -0.50
Reported Transmit Power (dBmV)     : 30.75      30.75      29.25      29.25
Peak Transmit Power (dBmV)         : 61.00      61.00      61.00      61.00
Phy Max Power (dBmV)               : 48.00      48.00      48.00      48.00
Minimum Transmit Power (dBmV)      : 21.00      21.00      21.00      21.00
Timing Offset             (97.6 ns): 1800       1800       1800       1800
Initial Timing Offset              : 1544       1544       1544       1544
Rng Timing Adj Moving Avg(0.381 ns): -1         0          -1         -1
Rng Timing Adj Lt Moving Avg       : -7         0          -7         -7
Rng Timing Adj Minimum             : -256       0          -256       -256
Rng Timing Adj Maximum             : 65536      65536      65536      65536
Pre-EQ Good                        : 0          0          0          0
Pre-EQ Scaled                      : 0          0          0          0
Pre-EQ Impulse                     : 0          0          0          0
Pre-EQ Direct Loads                : 0          0          0          0
Good Codewords rx                  : 1201       1262       833        656
Corrected Codewords rx             : 0          0          169        117
Uncorrectable Codewords rx         : 0          0          205        335
Phy Operating Mode                 : atdma*     atdma*     atdma*     atdma*
Upstream Channel                   : US4        US5        US6        US7
Ranging Status                     : sta        sta        sta        sta
Upstream SNR (dB)                  : 15.53      31.62      31.1       31.87
Upstream Data SNR (dB)             : --         --         --         --
Received Power (dBmV)              : 0.00       0.00       -0.50      0.50
Reported Transmit Power (dBmV)     : 29.25      30.75      30.75      30.75
Peak Transmit Power (dBmV)         : 61.00      61.00      61.00      61.00
Phy Max Power (dBmV)               : 48.00      48.00      48.00      48.00
Minimum Transmit Power (dBmV)      : 21.00      21.00      21.00      21.00
Timing Offset             (97.6 ns): 1800       1800       1800       1800
Initial Timing Offset              : 1544       1800       1544       1544
Rng Timing Adj Moving Avg(0.381 ns): -1         -1         46         0
Rng Timing Adj Lt Moving Avg       : -7         -7         104        0
Rng Timing Adj Minimum             : -256       -256       0          0
Rng Timing Adj Maximum             : 65536      256        65536      65536
Pre-EQ Good                        : 0          0          0          0
Pre-EQ Scaled                      : 0          0          0          0
Pre-EQ Impulse                     : 0          0          0          0
Pre-EQ Direct Loads                : 0          0          0          0
Good Codewords rx                  : 718        1328       1173       1252
Corrected Codewords rx             : 110        0          0          0
Uncorrectable Codewords rx         : 298        0          0          0
Phy Operating Mode                 : atdma*     atdma*     atdma*     atdma*
sysDescr                           : DOCSIS 3.0 Cable Modem Router
Downstream Power                   : 7.40 dBmV (SNR = 43.30 dB)
MAC Version                        : DOC3.0
QoS Provisioned Mode               : DOC1.1
Enable DOCSIS2.0 Mode              : Y
Modem Status                       : {Modem= w-online, Security=disabled}
Capabilities                       : {Frag=N, Concat=N, PHS=Y}
Security Capabilities              : {Priv=, EAE=Y, Key_len=}
L2VPN Capabilities                 : {L2VPN=Y, eSAFE=Y}
Sid/Said Limit                     : {Max US Sids=8, Max DS Saids=64}
```

```
Optional Filtering Support          : {802.1P=N, 802.1Q=N, DUT=Y}
Transmit Equalizer Support          : {Taps/Symbol= 1, Num of Taps= 24}
Number of CPE                       : 0(Max CPE = 16)
Number of CPE IPs                   : 0(Max CPE IPs = 16)
Number of CPE IPv6                  : 0(Max CPE IPv6 = 16)
CFG Max-CPE                         : 16
Flaps                               : 19(Oct 11 04:00:25)
Errors                              : 0 CRCs, 0 HCSes
Stn Mtn Failures                    : 0 aborts, 12 exhausted
Total US Flows                      : 1(1 active)
Total DS Flows                      : 1(1 active)
Total US Data                       : 3294 packets, 577031 bytes
Total US Throughput                 : 0 bits/sec, 0 packets/sec
Total DS Data                       : 2263 packets, 200777 bytes
Total DS Throughput                 : 0 bits/sec, 0 packets/sec
LB group ID assigned (index)        : 2153382146 (55810)
LB group ID in config file (index)  : N/A (N/A)
LB policy ID                        : 0
LB policy ID in config file         : 0
LB priority                         : 0
Tag                                 :
Required DS Attribute Mask          : 0x0
Forbidden DS Attribute Mask         : 0x0
Required US Attribute Mask          : 0x0
Forbidden US Attribute Mask         : 0x0
Service Type ID                     :
Service Type ID in config file      :
Ranging Class ID                    : 0x2
Active Classifiers                  : 0 (Max = NO LIMIT)
CM Upstream Filter Group            : 0
CM Downstream Filter Group          : 0
CPE Upstream Filter Group           : 0
CPE Downstream Filter Group         : 0
DSA/DSX messages                    : permit all
Voice Enabled                       : NO
DS Change Times                     : 0
Boolean Services                    : 2
Number of Multicast DSIDs Support   : 63
MDF Capability Mode                 : 2
IGMP/MLD Version                    : MLDv2
FCType10 Forwarding Support         : Y
Features Bitmask                    : 0x0
Total Time Online                   : 9h27m   (9h27m    since last counter reset)
CM Initialization Reason            : BAD_DHCP_ACK
CFG Max IPv6 CPE Prefix             : 16 (-1 used)
```

Additional References

The following sections provide references related to the Cable Modem Steering feature.

Related Documents

Related Topic	Document Title
CMTS cable commands	Cisco IOS CMTS Cable Command Reference

Standards and RFCs

Standard	Title
CM-SP-MULPIv3.0-I07-080215	DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification
CM-SP-MULPIv3.0-I18-120329	DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification
CM-SP-RFI2.0-I13-080215	DOCSIS 2.0 Radio Frequency Interface Specification

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Cable Modem Steering

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

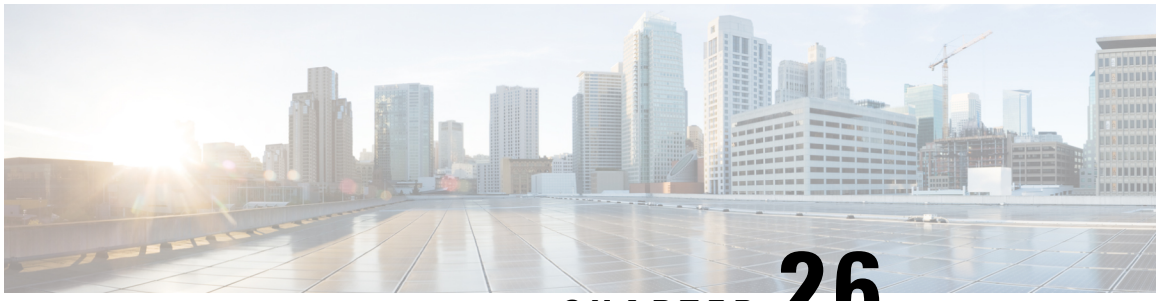
Table 71: Feature Information for Cable Modem Steering on the Cisco CMTS Routers

Feature Name	Releases	Feature Information
Cable Modem Steering	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Routers.

PART IV

Layer 2 and DOCSIS 3.1 Configuration

- DOCSIS 3.1 OFDM Channel Configuration, on page 471
- OFDM Channel Power Profile, on page 485
- DOCSIS 3.1 Path Selection, on page 491
- DOCSIS 3.1 Downstream Profile Selection, on page 497
- DOCSIS 3.1 Commanded Power for Upstream SC-QAMs, on page 503



CHAPTER 26

DOCSIS 3.1 OFDM Channel Configuration

This document describes how to configure the OFDM channel on the Cisco cBR Series Converged Broadband Router.

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 471
- Information about OFDM Channel Configuration, on page 472
- How to Configure OFDM Channel, on page 473
- Configuration Examples, on page 481
- Additional References, on page 482
- Feature Information for DOCSIS 3.1 OFDM Channel Configuration, on page 483

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 72: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information about OFDM Channel Configuration

OFDM Channels

DOCSIS 3.1 introduces modes for higher throughput and higher spectral efficiency while still allowing backward compatibility to DOCSIS 3.0. OFDM Channel support includes one OFDM channel per port with channel bandwidth from 24 MHz to 192 MHz wide.

Each OFDM channel supports a control profile, an NCP profile, and up to five data profiles. Profiles support one or more modulations.

Starting from Cisco IOS-XE release 3.18.1SP, you can configure the guard band of an OFDM channel to potentially trade off some performance margin using command **guardband-override**. By default, Cisco cBR-8 router use the default guard band, which is based on the roll off and spacing in OFDM channel profile.

DOCSIS 3.1 OFDM support also allows the user to configure the RF-channels 158 to 162 under the mac-domain as primary channel.

Channel Profile

A globally configured OFDM channel profile contains channel parameters, and the modulation or modulation profile associated with the control, NCP, and data profiles.

Each OFDM channel must specify an OFDM channel profile in its configuration.

Modulation Profile

A globally configured OFDM modulation profile assigns different modulations to ranges of sub-carriers, or lists of individual sub-carriers.

A modulation profile may be assigned to a control, NCP, or data profile in a channel profile.

OFDM Channel Exclusion Band

Ranges of frequencies can be excluded from all OFDM channels on a port using the **ofdm-freq-excl-band** command.

How to Configure OFDM Channel

Configuring OFDM Modulation Profile

To configure the OFDM modulation profile, follow the steps below:

```
enable
configure terminal
cable downstream ofdm-modulation-profile id
description text
subcarrier-spacing value
width value
start-frequency value
assign {modulation-default mod_prof_id | modulation mod_prof_id {list-subcarriers
 {freq-abs | freq-offset} value | range-subcarriers {freq-abs | freq-offset}
 value width value}}
```

Note Subcarrier spacing must match the subcarrier spacing of each channel profile in which it is configured.

Verifying OFDM Modulation Profile Configuration

To display the OFDM modulation profile details, use the **show cable ofdm-modulation-profiles** command as shown in the example below:

```
Router# show cable ofdm-modulation-profile 10

**** OFDM Modulation Profile Configuration ****

Prof  FFT  Width     Start-freq Modulations
ID    KHz  Hz        Hz
10    50   96000000  627000000  64   default
                                512  freq-abs  709050000 width   12000000
                                2048 freq-abs  629000000 width    6000000
```

```
Profile Subcarrier Modulations
  Modulation: Start-freq-abs[start-sc] - End-freq-abs[end-sc] Width-freq[num-sc]
  64  : 572600000[   0] -  626950000[1087]    54400000[1088]
  64  : 627000000[1088] -  628950000[1127]     2000000[  40]
  2048: 629000000[1128] -  634950000[1247]     6000000[ 120]
  64  : 635000000[1248] -  709000000[2728]    74050000[1481]
  512 : 709050000[2729] -  721000000[2968]    12000000[ 240]
  64  : 721050000[2969] -  722950000[3007]     1950000[  39]
  64  : 723000000[3008] -  777350000[4095]    54400000[1088]

**** OFDM Modulation Profile Assigned Channel Profiles ****

Prof  Channel
ID    Profiles
10    30
```

To display the associations between OFDM modulation profiles and OFDM channel profiles, use the **show cable ofdm-modulation-profile** command with **channel-profiles** option as shown in the example below:

```
Router# show cable ofdm-modulation-profile channel-profiles

**** OFDM Modulation Profile Assigned Channel Profiles ****

Prof  Channel
ID    Profiles
8     None
9     28
10    30
192   192
```

To display the OFDM modulation profile configurations, use the **show cable ofdm-modulation-profile** command with **configuration** option as shown in the example below:

```
Router# show cable ofdm-modulation-profile configuration

**** OFDM Modulation Profile Configuration ****

Prof  FFT  Width     Start-freq Modulations              Description
ID    KHz  Hz        Hz                                 (Limited to 20)
8     50   192000000 NA         2048 default
                                 512  freq-off   48000000
                                      width      24000000
9     50   96000000  627000000  512  default             512-1k-4k
                                 1024 freq-abs  635000000
                                      width      74050000
                                 4096 freq-abs  629000000
                                      width       6000000
10    50   96000000  627000000  64   default
                                 512  freq-abs  709050000
                                      width      12000000
                                 2048 freq-abs  629000000
                                      width       6000000
```

Configuring OFDM Channel Profile

To configure the OFDM channel profile, follow the steps below:

```
enable
configure terminal
cable downstream ofdm-chan-profile id
```

description *text*
cyclic-prefix *value*
guardband-override *value*
interleaver-depth *value*
pilot-scaling *value*
roll-off *value*
subcarrier-spacing *value*
profile-ncp modulation-default *mod_prof_id*
profile-control {**modulation-default** *mod_prof_id* | **modulation-profile** *mod_prof_id*}
profile-data *channel_data_prof_id* {**modulation-default** *mod_prof_id* |
modulation-profile *mod_prof_id*}

Verifying OFDM Channel Profile Configuration

To display the OFDM channel profile details, use the **show cable ofdm-chan-profiles** command as shown in the example below:

```
Router# show cable ofdm-chan-profile 21
**** OFDM Channel Profile Configuration ****

Prof  Cycl  Roll  Guardband  FFT  Intr   Pilot  Modulation (D-Default, P-Profile)

ID    Prfx  Off   Override   KHz  Depth  Scale  Cntrl   NCP     Data Profiles (count = 0)
                                                             1       2       3       4
      5
21    1024  128   2400000    50   16     48     D:1024  D:16    NA      NA      NA      NA
      NA

**** OFDM Channel Profile Assigned Channels ****

Prof  Admin  Controller:channels
ID
21    Up     6/0/4:158
```

To display the associations between OFDM channel profiles and OFDM channels, use the **show cable ofdm-chan-profiles** command with **channels** option as shown in the example below:

```
Router# show cable ofdm-chan-profile channels

**** OFDM Channel Profile Assigned Channels ****

Prof  Admin  Controller:channels
ID
20    Up     3/0/1:158             3/0/2:158             3/0/3:158             3/0/5:158

             3/0/6:158             3/0/7:158
30    Up     3/0/4:158
101   Up     3/0/0:158
```

To display the OFDM channel profile configurations, use the **show cable ofdm-chan-profiles** command with **configuration** option as shown in the example below:

```
Router# show cable ofdm-chan-profile configuration

**** OFDM Channel Profile Configuration ****

Prof  Cycl  Roll  Guardband  FFT  Intr   Pilot  Modulation (D-Default, P-Profile)
```

```
ID    Prfx  Off   Override   KHz  Depth  Scale  Cntrl   NCP     Data Profiles
             (Limited to 20)
                                                              1       2       3       4
      5
0     1024  128   NA         50   16     48     D:256   D:16    D:1024  NA      NA      NA
      NA
1     1024  128   NA         50   16     48     D:256   D:16    D:2048  D:1024  NA      NA
      NA
2     1024  128   NA         50   16     48     D:256   D:16    D:4096  D:2048  D:1024  NA
      NA
3     1024  128   NA         50   16     48     D:256   D:16    P:0     D:4096  D:2048
D:1024  NA
4     1024  128   NA         50   16     48     D:256   D:16    D:512   P:0     D:4096
D:2048  D:1024
5     1024  128   NA         25   16     48     D:256   D:16    D:1024  NA      NA      NA
      NA
6     1024  128   NA         25   16     48     D:256   D:16    D:2048  D:1024  NA      NA
      NA
7     1024  128   NA         25   16     48     D:256   D:16    D:4096  D:2048  D:1024  NA
      NA
8     1024  128   NA         25   16     48     D:256   D:16    P:1     D:4096  D:2048
D:1024  NA
9     1024  128   NA         25   16     48     D:256   D:16    D:512   P:1     D:4096
D:2048  D:1024
20    1024  128   NA         50   16     48     D:1024  D:16    NA      NA      NA      NA
      NA
21    1024  128   1000000    50   16     48     D:1024  D:16    NA      NA      NA      NA
      NA
```

Configuring OFDM Channel as Primary Channel

To configure an RF-channel in the mac-domain as an OFDM primary channel, use the following commands.

```
enable
configure terminal
interface cable <slot/subslot/port> downstream Integrated-Cable <slot/subslot/port>
rf-channel <ofdm-channel-number: 158-162>
end
```

Verifying OFDM Primary Channel Configuration

To display the OFDM channel configuration details, where the OFDM channel is the primary channel, use the command as shown in the following example:

```
Router#sh run int c3/0/3
Building configuration...

Current configuration : 539 bytes
!
interface Cable3/0/3
 load-interval 30
 downstream Integrated-Cable 3/0/3 rf-channel 0
 downstream Integrated-Cable 3/0/3 rf-channel 158
 upstream 0 Upstream-Cable 3/0/6 us-channel 0
 upstream 1 Upstream-Cable 3/0/6 us-channel 1
 upstream 2 Upstream-Cable 3/0/6 us-channel 2
 upstream 3 Upstream-Cable 3/0/6 us-channel 3
 cable upstream bonding-group 1
  upstream 0
```

```
  upstream 1
  upstream 2
  upstream 3
  attributes 80000000
 cable bundle 1
 cable cm-status enable 3 6-11 16-18 20-27
 cable privacy accept-self-signed-certificate
end
```

You can also use the following command to display the OFDM primary channel configuration details as shown in this example.

```
Router#sh cable mac-domain c3/0/3 cgd-associations
CGD Host  Resource  DS Channels            Upstreams    (ALLUS)   Active DS
Ca3/0/3   3/0/3     0                      0-3          Yes        0
                    158                    0-3          Yes        158
```

The **show cable mac-domain Cable <slot>/<subslot>/<port> mdd** command also displays the OFDM primary channel configuration details as shown in the example.

```
...
Downstream Active Channel List
        Channel ID:             159
        Frequency:              836000000Hz
        Primary Capable:        Primary-Capable
        CM-STATUS Event Bitmask:0x36
                                MDD Timeout
                                QAM FEC failure
                                MDD Recovery
                                QAM FEC recovery
        MAP/UCD Transport Indicator:  Can carry MAPs and UCDs
        OFDM PLC Params Bitmask:
               Tukey raised cosine window: 0.625
               Cyclic Prefix: 5.0
               Sub carrier spacing: 50
```

RF channels use a zero-based numbering scheme, whereas the downstream channel IDs are numbered starting from one. Thus RF channel 158 is equivalent to channel ID 159. The Channel ID in this example is 159. The `MAP/UCD Transport Indicator` shows that MAPs and UCDs are sent only on Primary Channels.

Configuring Port/Controller and Channel

To configure the port/controller and channel, follow the steps below:

```
enable
configure terminal
controller integrated-cable slot/subslot/port
max-ofdm-spectrum value
ofdm-freq-excl-band start-frequency value width value
rf-chan start_id [end_id]
ofdm channel-profile id start-frequency value width value [plc value]
```

Note The range of *start_id* is 158 to 162 in the OFDM channel configuration.

The maximum OFDM spectrum is assigned to OFDM channels, which is used by the the CMTS to calculate default port base power.

Ranges of frequencies can be excluded from all OFDM channels using the **ofdm-freq-excl-band** command.

Verifying Port/Controller and Channel Configuration

To display the RF port details, use the **show controller integrated-cable** command with **rf-port** option as shown in the example below:

```
Router# show controller integrated-cable 3/0/0 rf-port

Admin:  UP  MaxCarrier: 128  BasePower: 33 dBmV  Mode: normal
  Rf Module 0: UP
  Free freq block list has 3 blocks:
    45000000  - 107999999
    624000000 - 644999999
    837000000 - 1217999999
  Rf Port Status: UP
  MaxOfdmSpectrum:  192000000   Equivalent 6MHz channels: 32
  UsedOfdmSpectrum: 192000000   AvailOfdmSpectrum:        0
  DefaultBasePower: 33 dBmV     Equivalent 6MHz channels: 160
  OFDM frequency exclusion bands: None
```

To display the summary information on OFDM channel, use the **show controller integrated-cable** command with **rf-channel** option as shown in the example below:

```
Router# show controller integrated-cable 3/0/0 rf-port 158

Chan State Admin Mod-Type  Start       Width     PLC        Profile-ID    dcid  power
output
                           Frequency
 158   UP   UP   OFDM      627000000   96000000  663000000       20       159   34
NORMAL
```

To display detailed information on OFDM channel, use the **show controller integrated-cable** command with **rf-channel** and **verbose** options as shown in the example below:

```
Router# show controller integrated-cable 3/0/0 rf-port 158 verbose

Chan State Admin Mod-Type  Start       Width     PLC        Profile-ID    dcid  power
output
                           Frequency
 158   UP   UP   OFDM      627000000   96000000  663000000       30       159   32
NORMAL
 Resource status:   OK
 License: granted <17:02:35 EDT May 18 2016>
 OFDM channel license spectrum width: 92200000
 OFDM modulation license (spectrum width): 2K (6000000)
 OFDM config state: Configured

 OFDM channel details: [3/0/4:158]
----------------------------------------
 OFDM channel frequency/subcarrier range     : 627000000[1088] - 722999999[3007]
 OFDM spectrum frequency/subcarrier range    : 572600000[   0] - 777399999[4095]
 Active spectrum frequency/subcarrier range  : 628900000[1126] - 721049999[2969]
 OFDM channel center frequency/subcarrier    : 675000000[2048]
 PLC spectrum start frequency/subcarrier     : 663000000[1808]
 PLC frequency/subcarrier                    : 665800000[1864]
 Channel width                               : 96000000
 Active Channel width                        : 92200000
 OFDM Spectrum width                         : 204800000
 Chan prof id                                : 30
 Cyclic Prefix                               : 1024
 Roll off                                    : 128
 Interleave depth                            : 16
 Spacing                                     : 50KHZ
```

```
Pilot Scaling                               : 48
Control modulation profile                  : 10
NCP modulation default                      : 16
Data modulation default                     : None
Data modulation profile                     : None
Lower guardband width in freq/subcarriers   : 1900000[38]
Upper guardband width in freq/subcarriers   : 1900000[38]
Licensed 4K modulation spectrum width       : 0
Licensed 2K modulation spectrum width       : 6000000

PLC spectrum frequencies [subcarriers]              :
 663000000[1808] - 668999999[1927]

PLC channel frequencies [subcarriers]                :
 665800000[1864] - 666199999[1871]   Size: 8 subcarriers

Excluded frequencies [subcarriers]                  :
 572600000[   0] - 628899999[1125]     721100000[2970] - 777399999[4095]
 Count: 2252

Pilot frequencies [subcarriers]                     :
 *:PLC pilots
 630700000[1162]   634300000[1234]   637900000[1306]   641500000[1378]
 645100000[1450]   648700000[1522]   652300000[1594]   655900000[1666]
 659500000[1738]   663450000[1817]*  664050000[1829]*  664600000[1840]*
 665050000[1849]*  666900000[1886]*  667350000[1895]*  667900000[1906]*
 668500000[1918]*  669100000[1930]   672700000[2002]   676300000[2074]
 679900000[2146]   683500000[2218]   687100000[2290]   690700000[2362]
 694300000[2434]   697900000[2506]   701500000[2578]   705100000[2650]
 708700000[2722]   712300000[2794]   715900000[2866]   719500000[2938]
 Count: 32

Active frequencies [subcarriers]                    :
 628900000[1126] - 721099999[2969]
 Count: 1844

Data frequencies [subcarriers]                      :
 628900000[1126] - 630699999[1161]     630750000[1163] - 634299999[1233]
 634350000[1235] - 637899999[1305]     637950000[1307] - 641499999[1377]
 641550000[1379] - 645099999[1449]     645150000[1451] - 648699999[1521]
 648750000[1523] - 652299999[1593]     652350000[1595] - 655899999[1665]
 655950000[1667] - 659499999[1737]     659550000[1739] - 663449999[1816]
 663500000[1818] - 664049999[1828]     664100000[1830] - 664599999[1839]
 664650000[1841] - 665049999[1848]     665100000[1850] - 665799999[1863]
 666200000[1872] - 666899999[1885]     666950000[1887] - 667349999[1894]
 667400000[1896] - 667899999[1905]     667950000[1907] - 668499999[1917]
 668550000[1919] - 669099999[1929]     669150000[1931] - 672699999[2001]
 672750000[2003] - 676299999[2073]     676350000[2075] - 679899999[2145]
 679950000[2147] - 683499999[2217]     683550000[2219] - 687099999[2289]
 687150000[2291] - 690699999[2361]     690750000[2363] - 694299999[2433]
 694350000[2435] - 697899999[2505]     697950000[2507] - 701499999[2577]
 701550000[2579] - 705099999[2649]     705150000[2651] - 708699999[2721]
 708750000[2723] - 712299999[2793]     712350000[2795] - 715899999[2865]
 715950000[2867] - 719499999[2937]     719550000[2939] - 721099999[2969]
 Count: 1804

Profiles:
 Number of profiles: 2
 CTRL profile (Profile A): rate: 461916 kbps, usable rate: 368000 kbps
 Active frequencies [subcarriers]:
 Modulation:Start-freq[start-subcarrier] - End-freq[end-subcarrier]
 ------------------------------------------------------------------
 64   :628900000[1126] - 628950000[1127]     2048 :629000000[1128] - 630650000[1161]
 2048 :630750000[1163] - 634250000[1233]     2048 :634350000[1235] - 634950000[1247]
```

```
 64   :635000000[1248] - 637850000[1305]     64   :637950000[1307] - 641450000[1377]
 64   :641550000[1379] - 645050000[1449]     64   :645150000[1451] - 648650000[1521]
 64   :648750000[1523] - 652250000[1593]     64   :652350000[1595] - 655850000[1665]
 64   :655950000[1667] - 659450000[1737]     64   :659550000[1739] - 663400000[1816]
 64   :663500000[1818] - 664000000[1828]     64   :664100000[1830] - 664550000[1839]
 64   :664650000[1841] - 665000000[1848]     64   :665100000[1850] - 665750000[1863]
 64   :666200000[1872] - 666850000[1885]     64   :666950000[1887] - 667300000[1894]
 64   :667400000[1896] - 667850000[1905]     64   :667950000[1907] - 668450000[1917]
 64   :668550000[1919] - 669050000[1929]     64   :669150000[1931] - 672650000[2001]
 64   :672750000[2003] - 676250000[2073]     64   :676350000[2075] - 679850000[2145]
 64   :679950000[2147] - 683450000[2217]     64   :683550000[2219] - 687050000[2289]
 64   :687150000[2291] - 690650000[2361]     64   :690750000[2363] - 694250000[2433]
 64   :694350000[2435] - 697850000[2505]     64   :697950000[2507] - 701450000[2577]
 64   :701550000[2579] - 705050000[2649]     64   :705150000[2651] - 708650000[2721]
 64   :708750000[2723] - 709000000[2728]     512  :709050000[2729] - 712250000[2793]
 512  :712350000[2795] - 715850000[2865]     512  :715950000[2867] - 719450000[2937]
 512  :719550000[2939] - 721000000[2968]     64   :721050000[2969] - 721050000[2969]
 Active subcarrier count: 1804, ZBL count: 0
 Discontinuity time [days:hours:mins:secs]: 00:00:54:32 [16:15:02 EDT May 18 2016]

 NCP profile:
 Active frequencies [subcarriers]:
 Modulation:Start-freq[start-subcarrier] - End-freq[end-subcarrier]
 ----------------------------------------------------------------
 16   :628900000[1126] - 630650000[1161]     16   :630750000[1163] - 634250000[1233]
 16   :634350000[1235] - 637850000[1305]     16   :637950000[1307] - 641450000[1377]
 16   :641550000[1379] - 645050000[1449]     16   :645150000[1451] - 648650000[1521]
 16   :648750000[1523] - 652250000[1593]     16   :652350000[1595] - 655850000[1665]
 16   :655950000[1667] - 659450000[1737]     16   :659550000[1739] - 663400000[1816]
 16   :663500000[1818] - 664000000[1828]     16   :664100000[1830] - 664550000[1839]
 16   :664650000[1841] - 665000000[1848]     16   :665100000[1850] - 665750000[1863]
 16   :666200000[1872] - 666850000[1885]     16   :666950000[1887] - 667300000[1894]
 16   :667400000[1896] - 667850000[1905]     16   :667950000[1907] - 668450000[1917]
 16   :668550000[1919] - 669050000[1929]     16   :669150000[1931] - 672650000[2001]
 16   :672750000[2003] - 676250000[2073]     16   :676350000[2075] - 679850000[2145]
 16   :679950000[2147] - 683450000[2217]     16   :683550000[2219] - 687050000[2289]
 16   :687150000[2291] - 690650000[2361]     16   :690750000[2363] - 694250000[2433]
 16   :694350000[2435] - 697850000[2505]     16   :697950000[2507] - 701450000[2577]
 16   :701550000[2579] - 705050000[2649]     16   :705150000[2651] - 708650000[2721]
 16   :708750000[2723] - 712250000[2793]     16   :712350000[2795] - 715850000[2865]
 16   :715950000[2867] - 719450000[2937]     16   :719550000[2939] - 721050000[2969]
 Active subcarrier count: 1804, ZBL count: 0

CCCs:
  OCD CCC: 2
  DPD CCCs:
    Control profile  (Profile A) CCC: 2
        NCP profile  CCC: 2
Resource config time taken: 2286 msecs


JIB channel number: 776
Chan Pr EnqQ  Pipe  RAF   SyncTmr DqQ ChEn   RAF Pipe Phy0 Phy1 Tun# SessId 0[TkbRt MaxP]
    1[TkbRt MaxP]
 776  0   384     1   725       0 384 0100 13032    1    0    1    2      0 479610000 4485120
 383688000 4485120
 776  1   384     1  4786       0 384 0100  2190    1    0    1    2      0 479610000 4485120
 383688000 4485120
 776  2   384     1  4786       0 384 0100  2190    1    0    1    2      0 479610000 4485120
 383688000 4485120
 776  3   384     1  4786       0 384 0100  2190    1    0    1    2      0 479610000 4485120
 383688000 4485120
 776  4   384     1  4786       0 384 0100  2190    1    0    1    2      0 479610000 4485120
 383688000 4485120
```

```
 776  5   384     1  4786       0 384 0100  2190    1    0    1    2      0 479610000 4485120
 383688000 4485120
 776  6   384     1  4786       0 384 0100  2190    1    0    1    2      0 479610000 4485120
 383688000 4485120
 776  7   384     1     0       0 384 0100     0    1    0    1    2      0 479610000 4485120
 383688000 4485120

Chan   Qos-Hi   Qos-Lo    Med-Hi   Med-Lo    Low-Hi   Low-Lo
 776   368640   245760    368640   245760    614400   368640
Chan  Med Low TB-neg Qos_Exc  Med_Xof  Low_Xof  Qdrops(H-M-L)  Pos    Qlen(Hi-Med-lo) Fl
Tgl_cnt Rdy_sts
  776   0   0      0       0        0        0    0    0    0  Y      0      0      0  0
      0 ff
Chan       Rate   Neg   Pos    LastTS  CurrCr  Pos  [PLC Rate Neg Pos]
 776   10485750 65535 65535 116199669 268431360   Y [MM 86 128 1114][EM 87 128 6204][TR 2
9 3102]
DSPHY Info:
 Local rf port 0 , rf chan 158 pic loss 123
 non short CWs: = 235681130, shorts = 0, stuff bytes = 235639172 bch 235681130
 NCP msgs: = 453809753, PLC encodings = 16902476
 flow0 rcv 70203 flow1 rcv 3 flow0 drops 0 flow1 drops 0
```

Configuration Examples

This section provides examples for configuring the OFDM channel.

Example1: Configuring OFDM Channel

Note The OFDM modulation profile must be configured before the OFDM channel profile which references it.

The following example shows how to configure the OFDM channel:

```
enable
configure terminal
cable downstream ofdm-modulation-profile 9
description 512-1k-4k
subcarrier-spacing 50KHz
width 96000000
start-frequency 627000000
assign modulation-default 512-QAM
assign modulation 1024-QAM range-subcarriers freq-abs 635000000 width 74050000
assign modulation 4096-QAM range-subcarriers freq-abs 629000000 width 6000000
exit
configure terminal
cable downstream ofdm-chan-profile 20
description Data profiles: 2 single mod, 1 mixed mod
cyclic-prefix 192
interleaver-depth 16
pilot-scaling 48
roll-off 128
subcarrier-spacing 50KHz
profile-ncp modulation-default 16-QAM
```

```
profile-control modulation-default 256-QAM
profile-data 1 modulation-default 1024-QAM
profile-data 2 modulation-default 2048-QAM
profile-data 3 modulation-profile 9
exit
configure terminal
controller integrated-cable 3/0/0
max-ofdm-spectrum 96000000
ofdm-freq-excl-band start-frequency 683000000 width 10000000
rf-chan 158
power-adjust 0
docsis-channel-id 159
ofdm channel-profile 20 start-frequency 627000000 width 96000000 plc 663000000
```

Example 2: Configuring OFDM Primary Channel in the MAC Domain

```
enable
configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
interface cable 3/0/0
downstream Integrated-Cable 3/0/3 rf-channel 158
end
```

Additional References

Related Document

Document Title	Link
Cisco cBR Converged Broadband Routers Layer 2 and DOCSIS 3.0 Configuration Guide	http://www.cisco.com/c/en/us/td/docs/cable/cbr/configuration/guide/b_cbr_layer2_docsis30.html

MIBs

MIBs	MIBs Link
DOCS-IF31-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

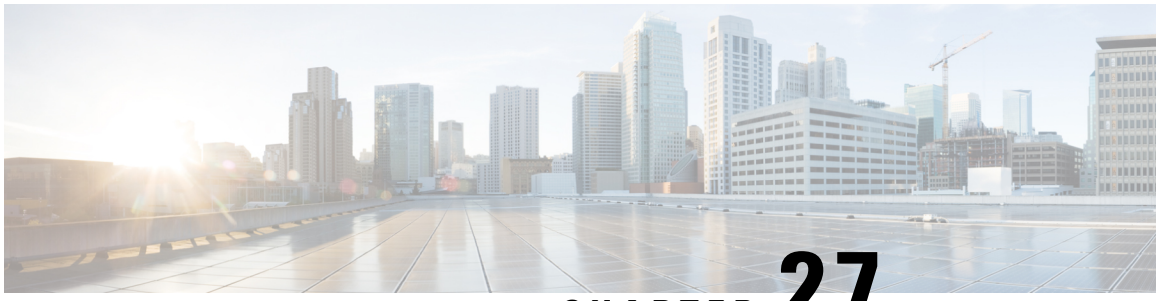
Feature Information for DOCSIS 3.1 OFDM Channel Configuration

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 73: Feature Information for DOCSIS 3.1 OFDM Channel Configuration

Feature Name	Releases	Feature Information
DOCSIS 3.1 OFDM Channel Support	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.
Full Spectrum 108-1218 MHz Support	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.
DOCSIS 3.1 OFDM Primary Channel Support	Cisco IOS XE Everest 16.5.1	This feature was introduced in Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 27

OFDM Channel Power Profile

The OFDM Channel Power Profile feature helps in adjusting the power-level of 6 MHz bands in a DOCSIS 3.1 downstream OFDM channel.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 485
- Information About OFDM Channel Power Profile, on page 486
- How to Configure the OFDM Channel Power Profile, on page 487
- Configuration Example for OFDM Power Profile, on page 489
- Feature Information for OFDM Channel Power Profile, on page 489

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 74: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About OFDM Channel Power Profile

The OFDM power profile provides a better, consistent power-level output at the cable modem, compensating the power levels at a finer granularity. It reduces the differing amounts of cable-loss over the bandwidth of OFDM channel.

This feature enables the Cisco cBR Series Converged Broadband Router to correct the transmission loss due to the cable in the plant.

The OFDM power profile (ofdm-power-profile) adjusts the transmission power level of each 6 MHz in an OFDM channel. The OFDM channel width can range from 24 MHz to 192 MHz, resulting in band-counts between 4 and 32 for the profile.

Each 6 MHz band is referenced by a band index (band-index) that is zero-based, with a maximum band range of 192 MHz OFDM channel being 0 to 31. Each band within the OFDM channel can have a unique power level setting. The OFDM power profile allows a total band adjustment range of 8 dB. Under some specific conditions, if the OFDM channel's downstream controller's base channel power is set to exceed the maximum DRFI specification power level, the OFDM power profile adjustment range can become as high as 9 dB.

In a power profile, you can set the power level (power-adjust-default) to a default value. This default value is applied to any band that is not configured through any other means.

You can configure band power levels in two methods: through the power tilt config (power-tilt-linear) or by configuring the power level for a band or range of bands (band-index). You can use both methods for configuring the band power levels simultaneously within an OFDM power profile.

The power tilt configuration applies a linear power-adjust value between the `power-adjust-default` value applied to the band index 0, and the power-tilt-linear adjust value applied to the highest band index of the profile. For example, an OFDM Power Profile of 96 MHz, with a power-tilt-linear of 4 dB, and power-adjust-default of 0 dB, has16 bands numbered 0 to 15, band index 0 is +0 dB, band index 15 is +4 dB, and bands 1 to 14 contain the linear power level setting based on the slope of the line between the band 0 and band 15 to the nearest 1/10th dB.

The band-index configuration applies a specified value to the indicated bands. The band-index configuration can specify a single band or a range of bands. A power-adjust configuration is used to specify the power level for the bands to the nearest 1/10th dB.

You can simultaneously use both power tilt and band index, where band-index is applied last. When you use both, the power-tilt-linear values can be overridden using the band-index power-adjust values.

A maximum of 64 OFDM power profiles can be configured on the Cisco cBR routers, numbered from 1 to 64. You can apply a single OFDM power profile to multiple controller OFDM channels, across line cards, as long as all validity checks pass during configuration. The router console displays an error message explaining any configuration errors or warnings.

Restrictions for Configuring OFDM Power Profile

The following restrictions are applicable for configuring an OFDM power profile:

- OFDM power profile can be configured only on DOCSIS 3.1 system
- The power profile can be applied only to downstream controller OFDM channels (RF-channels 158 to 162)

How to Configure the OFDM Channel Power Profile

Configuring OFDM Power Profile Using Band-index

Use the following commands along with the band-index configuration to configure OFDM Power Profile, where the band-index values act as an override.

```
enable
configure terminal
cable downstream ofdm-power-profile <profile_id>
  power-adjust-default -2.1
  band-index 0 7
    power-adjust -1.0
  band-index 8 15
    power-adjust -0.5
  band-index 16 23
    power-adjust 0.5
  band-index 24 31
    power-adjust 1.5

controller Integrated-Cable {slot}/{subslot}/{port}
```

```
rf-channel {158 - 162 }
power-profile {ofdm-power-profile-id}
```

Verifying the Power Profile Configuration

To display the power profile configuration details, use the **show cable ofdm-power-profile** command as given in the following example. This command also displays the actual power-band power levels as set by the profile.

```
Router> show cable ofdm-power-profile 3
OFDM Power Profile 3

Power-Adjust-Default(*): -2.1

Power-Band:
[00-07]   -1.0   -1.0   -1.0   -1.0   -1.0   -1.0   -1.0   -1.0
[08-15]   -0.5   -0.5   -0.5   -0.5   -0.5   -0.5   -0.5   -0.5
[16-23]    0.5    0.5    0.5    0.5    0.5    0.5    0.5    0.5
[24-31]    1.5    1.5    1.5    1.5    1.5    1.5    1.5    1.5

  +4 .0 |
        |
  +3 .0 |
        |
  +2 .0 |                                                       * * * * * * * *
        |
  +1 .0 |                                     * * * * * * * *
        |
  +0 .0 | ------------------------------------------------------------------------
        |                     * * * * * * * *
  -1 .0 | * * * * * * * * *
        |
  -2 .0 |
        |
  -3 .0 |
        |
  (dB)    0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3 3
          0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                band-index
```

Configuring OFDM Power Profile with Linear Power-tilt

Use the following commands to configure the OFDM power profile with a linear power-tilt and the band-index override.

```
enable
configure terminal
cable downstream ofdm-power-profile <profile_id>
  power-adjust-default 0.0
  power-tilt-linear 3.5
  band-index 0
  power-adjust 4.0
```

Verifying the Power Profile Using show controller Command

Use the **show controller** command to display the absolute power-band levels as set by the power profile. When the power-profile is applied to the controller, the power level displayed is the actual transmit power level in dBmV.

```
Router>show controller Integrated-Cable 3/0/0 rf-channel 158 verbose
Chan State Admin Mod-Type  Start  Width     PLC        Profile-ID    dcid  power  output
                           Frequency
 158   UP   UP   OFDM      849000000   96000000  856000000       20       159   33.0
NORMAL
 Resource status:   OK
 License: granted <09:23:14 EDT Aug 1 2016>
 OFDM channel license spectrum width: 92200000
 OFDM config state: Configured

 OFDM Power Profile: 3
Power-Band:
[00-07]   32.0   32.0   32.0   32.0   32.0   32.0   32.0   32.0
[08-15]   32.5   32.5   32.5   32.5   32.5   32.5   32.5   32.5
[16-23]   33.5   33.5   33.5   33.5   33.5   33.5   33.5   33.5
[24-31]   34.5   34.5   34.5   34.5   34.5   34.5   34.5   34.5

OFDM channel details: [3/0/0:158]
-----------------------------------------
```

Configuration Example for OFDM Power Profile

This section provides example for the OFDM Power Profile configuration.

Example: OFDM Power Profile with Linear Power-tilt Configuration

```
enable
configure terminal
cable downstream ofdm-power-profile 3
  power-adjust-default 0.0
  power-tilt-linear 3.5
  band-index 0
  power-adjust 4.0
```

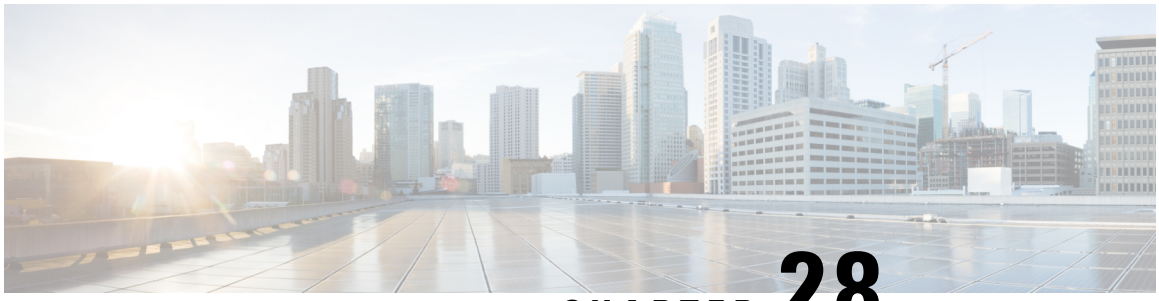
Feature Information for OFDM Channel Power Profile

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 75: Feature Information for OFDM Channel Power Profile

Feature Name	Releases	Feature Information
OFDM Channel Power Profile	Cisco IOS-XE Release 16.5.1	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 28

DOCSIS 3.1 Path Selection

This document describes how to configure the path selecion on the Cisco cBR Series Converged Broadband Router.

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 491
- Information about Path Selection, on page 492
- How to Configure Path Selection, on page 492
- Additional References, on page 495
- Feature Information for DOCSIS 3.1 Path Selection, on page 496

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 76: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information about Path Selection

DOCSIS 3.1 Path Selection feature is enhanced to support OFDM downstream channels. The RCC decision process is enhanced to include OFDM channels.

How to Configure Path Selection

Configuring Downstream Bonding Group with OFDM Channel

To configure the downstream bonding group with OFDM channel, follow the steps below:

```
enable
configure terminal
interface wideband-cable slot/subslot/bay:wideband-channel
description text
cable bundle id
cable rf-channels channel-list grouplist bandwidth-percent percentage-bandwidth
```

Note Channel 158 to 162 are specified as OFDM channel.

Verifying Downstream Bonding Group with OFDM Channel Configuration

To display the details of the downstream bonding group with OFDM channel, use the **show running-config interface** command as shown in the example below:

```
Router# show running-config interface wideband-cable 3/0/0:13

Building configuration...

Current configuration : 212 bytes
!
interface Wideband-Cable3/0/0:13
 description D31-DSBG: 1 SC-QAM plus 1 OFDM
 cable bundle 1
 cable rf-channels channel-list 8 bandwidth-percent 30
 cable rf-channels channel-list 158 bandwidth-percent 25
end
```

Verifying the Path Selection Status

To display the path selection status of a cable modem, use the **show cable modem path-sel** command as shown in the example below:

```
router#show cable modem 38c8.5cfe.efa6 path-sel

CM 38c8.5cfe.efa6 Path-Sel Info: 07:20

RCS Filter Result: Succeed
Candidate RCS List: 2
  RCC-Id   Owner-Id   Preliminary  RCP      TLV-56   LBG      SF-Attr   CM-Attr
  1        1 :12289   Pass         Pass      --      Pass     Pass      Pass
  2        1 :12290   Pass         Pass      --      Pass     Pass      Pass

TCS Filter Result: Succeed
TCS Info:
  TCS in CGD            : 0x7         UCID: 1 2 3
  TCS in Freq Range     : 0x7         UCID: 1 2 3
  TCS Impaired          : 0x0
TCS Passed filters:
  Preliminary           : 0x7         UCID: 1 2 3
  LB Group              : 0x7         UCID: 1 2 3
  SF Attr Mask          : 0x7         UCID: 1 2 3
  CM Attr Mask          : 0x7         UCID: 1 2 3

Candidate US-BG List: 4
  UBG-Id   Chan-Mask    Preliminary  TLV-56   LBG      SF-Attr   CM-Attr
  1        0x7          Pass          --      Pass     Pass      Pass
  65537    0x2          Pass          --      Pass     Pass      Pass
  65538    0x4          Pass          --      Pass     Pass      Pass
  65536    0x1          Pass          --      Pass     Pass      Pass

Primary DS Chan Result: Skipped
Candidate Primary DS Chan List: 0
```

```
Primary US Chan Result: Skipped
Candidate Primary US Chan List: 0
```

Clearing the Path Selection Status

To clear the path selection status for all CMs, use the **clear cable modem all path-sel** command as shown in the example below:

```
Router# clear cable modem all path-sel

Router# show cable modem c8fb.26a6.c46a path-sel

CM c8fb.26a6.c46a Path-Sel Info: N/A
Path-Sel status has been cleared after register online.
```

Verifying the RCC Configuration

To verify the runtime RCCs on a cable interface, use the **show cable mac-domain rcc** command as shown in the example below:

```
Router# show cable mac-domain cable 7/0/0 rcc

RCC-ID   RCP               RCs MD-DS-SG CMs   WB/RCC-TMPL      D3.0 D3.1
4        00 00 00 00 00    16  0        1     WB (Wi7/0/0:0)   Y    Y
5        00 00 00 00 00    25  0        2     WB (Wi7/0/0:1)   N    Y
6        00 10 00 00 08    8   0        0     RCC-TMPL(3:1)    Y    N
7        00 00 00 00 00    4   0        0     WB (Wi7/0/0:4)   Y    Y
```

To display the detailed information for only DOCSIS 3.1 capable RCC, use the **show cable mac-domain rcc simplified** command as shown in the example below:

```
router#show cable mac-domain cable 7/0/0 rcc 5 simplified

RCC ID                 : 5
Created Via            : Wideband - Wi7/0/0:1
CM attribute mask      : 0x80000000

Primary Receive Channel List:
  Chan Idx   RF Chan         DCID  Freq
  1          In7/0/0:0       1     453000000

Non-Primary Receive Channel List:
  Chan Idx   RF Chan         DCID  Freq
  2          In7/0/0:1       2     459000000
  3          In7/0/0:2       3     465000000
  4          In7/0/0:3       4     471000000
  5          In7/0/0:4       5     477000000
  6          In7/0/0:5       6     483000000
  7          In7/0/0:6       7     489000000
  8          In7/0/0:7       8     495000000
  9          In7/0/0:8       9     501000000
  10         In7/0/0:9       10    507000000
  11         In7/0/0:10      11    513000000
  12         In7/0/0:11      12    519000000
  13         In7/0/0:12      13    525000000
  14         In7/0/0:13      14    531000000
  15         In7/0/0:14      15    537000000
  16         In7/0/0:15      16    543000000
  17         In7/0/0:16      17    549000000
```

```
  18          In7/0/0:17      18    555000000
  19          In7/0/0:18      19    561000000
  20          In7/0/0:19      20    567000000
  21          In7/0/0:20      21    573000000
  22          In7/0/0:21      22    579000000
  23          In7/0/0:22      23    585000000
  24          In7/0/0:23      24    591000000
  25          In7/0/0:158     159   663000000

OFDM Receive Channel List:
  Chan Idx    RF Chan         DCID  PLC-Freq        Profiles
  25          In7/0/0:158     159   663000000       0 1 2
```

Additional References

Related Document

Document Title	Link
Cisco cBR Converged Broadband Routers Layer 2 and DOCSIS 3.0 Configuration Guide	http://www.cisco.com/c/en/us/td/docs/cable/cbr/configuration/guide/b_cbr_layer2_docsis30.html

MIBs

MIBs	MIBs Link
• DOCS-IF31-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

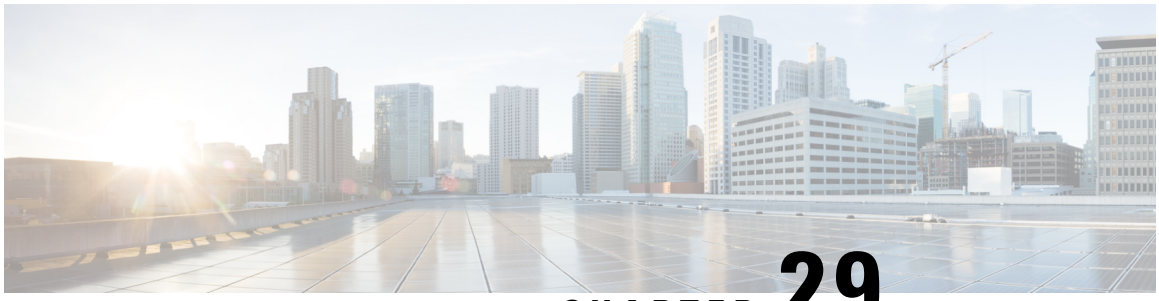
Feature Information for DOCSIS 3.1 Path Selection

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 77: Feature Information for DOCSIS 3.1 Path Selection

Feature Name	Releases	Feature Information
DOCSIS 3.1 Path Selection	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Routers.



CHAPTER 29

DOCSIS 3.1 Downstream Profile Selection

First Published: July 13, 2016

DOCSIS 3.1 introduces the concept of downstream profiles for OFDM channels.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 497
- Information about Downstream Profiles, on page 498
- How to Configure Profiles, on page 499
- Additional References, on page 501
- Feature Information for Downstream Profile Selection, on page 501

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 78: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information about Downstream Profiles

A profile is a list of modulation orders that are defined for each of the subcarriers within an OFDM channel. The CMTS can define multiple profiles for use in an OFDM channel, where the profiles differ in the modulation orders assigned to each subcarrier.

You can use the following commands to view the profiles:

- To display the profiles associated with the cable modems (CMs), use the **show cable modem** *[ip-address| mac-address| cable| {slot | subslot | cable-interface-index}]* **phy ofdm-profile** command.
- To display detailed profile management data associated with specific cable modem, use the **show cable modem** *[ip-address| mac-address]* **prof-mgmt** command.

The CMTS can assign different profiles for different groups of CMs.

Default Data Profile

The first time a CM registers, it is assigned a default data profile. The default data profile is "profile-data 1". If "profile-data 1" is not configured, "profile-control" is assigned to the CM.

Note Profile A, with profile ID 0, is also referred to as the control profile.

Recommended Profile

Based on the Receive Modulation Error Ratio (RxMER) values collected from a modem using the **cable modem** *ip* **opt0** command, and collected automatically and periodically in the background, the CMTS finds among the existing profiles the one that may provide the highest speed, and yet at the same time may have sufficient Signal to Noise Ratio (SNR) margin for the modem to receive code words with acceptable error. This profile is called the recommended profile for that CM. The **show cable modem phy ofdm-profile** command displays the recommended profile for each CM.

Internal PMA is enabled by default on the cBR-8. If external PMA is enabled, internal PMA is disabled and can be enabled by running the following config command on the cBR-8:

```
no cable downstream ofdm-prof-mgmt prof-upgrade-pma
```

In external PMA, the cBR-8 does not automatically upgrade a CM's profile to the recommended profile. In internal PMA, the profile can be upgraded to the recommended profile in the next OFDM Profile Test (OPT) cycle.

A user configurable age is associated with each recommended profile, which can be configured as follows:

```
Router (config)#cable downstream ofdm-prof-mgmt recommend-profile-age age-in-minutes
```

If the recommended profile exceeds this age, it is no longer valid for that CM.

Unfit Profile

When the CMTS receives CM-STATUS Event 16 (DS OFDM Profile Failure), the profile indicated in the CM-STATUS message is marked as 'unfit profile' for this modem.

A user configurable maximum age is associated with each unfit profile, which can be configured as follows:

```
Router (config)#cable downstream ofdm-prof-mgmt unfit-profile-age age-in-minutes
```

If the unfit profile for a modem exceeds this age, it is no longer valid.

How to Configure Profiles

Configuring Profile Downgrade

A CM sends a CM-STATUS Event 16 message to indicate a DS OFDM profile failure. When this indication is received by the CMTS, it takes immediate action to downgrade the modem to a lower profile, as per the profile ordering displayed by the following command:

```
Router# show controllers integrated-Cable 2/0/3 rf-channel 158 prof-order
```

The following table, extracted from [DOCSIS 3.1 MULPI], lists the CM-Status events that will trigger a profile downgrade:

Table 79: Table: CM-Status Events for Profile Downgrade

Event Type	Event Condition	Status Report Events		Parameters reported by CM	
		Trigger event to "on"	Trigger event to "off"	DCID	Profile ID
16	DS OFDM profile failure	Loss of FEC lock on one of the assigned downstream OFDM profiles of a channel	Re-establishment of FEC lock for that OFDM profile; OR Removal of the channel from the active channel list in the primary channel MDD; OR Removal of the channel from the CM's Receive Channel set via DBC-REQ	Yes	Yes

To disable the automatic profile downgrade, use the following command in global configuration mode:

```
Router (config)#no cable downstream ofdm-prof-mgmt prof-dwngrd-auto
```

Configuring RxMER to Bit Loading Mapping

There are many ways to map the Receive Modulation Error Ratio (RxMER) values to bit loading values. We use the following mapping recommended in [DOCSIS 3.1 OSSI], as our baseline mapping:

RxMER (in ¼ DB)	QAM	Bit Loading
60	16	4
84	64	6
96	128	7
108	256	8
122	512	9
136	1024	10
148	2048	11
164	4096	12

RxMER (in ¼ DB)	QAM	Bit Loading
184	8192	13
208	16384	14

- To configure a margin to adjust the RxMER to bit loading mapping, use the following command:

  ```
  Router(config)# cable downstream ofdm-prof-mgmt mer-margin-qdb interval-in-minutes
  ```

  This configured value (*quarter-DB*) is added to the RxMER values collected by CMTS before using the above mapping table, thus giving a user more control in selecting the recommended profiles.
- To specify the percentage of subcarriers that can be ignored in the recommended profile calculation, use the following command:

  ```
  Router(config)# cable downstream ofdm-prof-mgmt exempt-sc-pct percent
  ```

  This provides a way to specify the extent that the outliers can be ignored.
- To configure the RxMER poll interval, use the following command:

  ```
  Router(config)# cable downstream ofdm-prof-mgmt rxmer-poll-interval interval-in-minutes
  ```

  The CMTS uses OPT message with bit-0 option to collect RxMER data from CMs, after the initial CM registration and periodically thereafter. The collected RxMER data is used to compute the recommended profile for each CM.

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

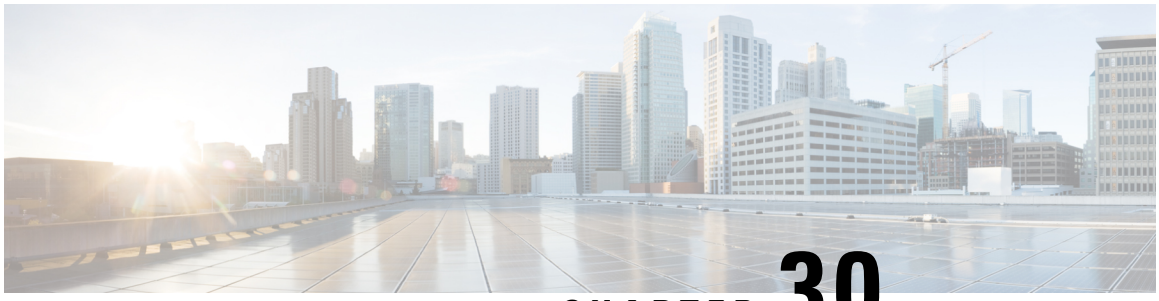
Feature Information for Downstream Profile Selection

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 80: Feature Information for Downstream Profile Selection

Feature Name	Releases	Feature Information
Downstream Profile Selection	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Routers.



CHAPTER 30

DOCSIS 3.1 Commanded Power for Upstream SC-QAMs

This guide describes commanded power for upstream SC-QAMs on the Cisco cBR Router.

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 503
- Information About Commanded Power Feature for Upstream SC-QAMs, on page 504
- Feature TLVs, on page 505
- Additional References, on page 506
- Feature Information for Commanded Power for US SC-QAMs, on page 506

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 81: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About Commanded Power Feature for Upstream SC-QAMs

Commanded Power feature for UpStream Single Carrier Quadrature Amplitude Modulation (US SC-QAMs) supports a new method during ranging to dynamically set the transmit power level of a DOCSIS 3.1 cable modem.

To view the new commanded power levels pr upstream, use the following command:

```
Router# show cable modem [ ip-address | mac-address | cable {slot /subslot
/cable-interface-index}] verbose
```

Note DOCSIS 3.1 Commanded Power feature is enabled by default.

Feature TLVs

TLVs Affected by Commanded Power for US SC-QAMs

The following table lists the TLVs affected by the DOCSIS 3.1 Ranging Response (RNG-RSP) Commanded Power for upstream SC-QAMs:

Name	Type	DOCSIS 3.1 Value
Power Level Adjust	2	TX Power offset adjustment (signed 8-bit, 1/4-dB units)
Power Offset	12.4.4	TX Power offset adjustment (signed 8-bit, 1/4-dB units)
Dynamic Range Window Upper Edge	14	The upper edge of the Dynamic Range Window expressed in units 1/4 dB below the max allowable setting (Phi) [DOCSIS PHYv3.0].
Commanded Power	17	This TLV contains the Dynamic Range Window value, P1.6load_min_set as well as the Transmit Power Level for each of the channels in the CM's Transmit Channel Set, expressed in units of quarter dBmV.

Commanded Power Sub-TLVs

The following table lists the sub-TLVs for DOCSIS 3.1 Commanded Power:

Name	Type (1 byte)	Length (1 byte)	Value (Variable Length)
Commanded Power	17	5 + 3*N	
Dynamic Range Window	17.1	1	The range, in decibels, of the maximum difference in power per 1.6 MHz between multiple transmitters in a cable modem’s Transmit Channel Set.

Name	Type (1 byte)	Length (1 byte)	Value (Variable Length)
List of Upstream Channel IDs and Corresponding Transmit Power Levels	17.2	3*N	Values for each channel in the TCS: • Bits 23 to 16: UCID • Bits 15 to 0: Transmit Power Level (quarter dBmV)

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Commanded Power for US SC-QAMs

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

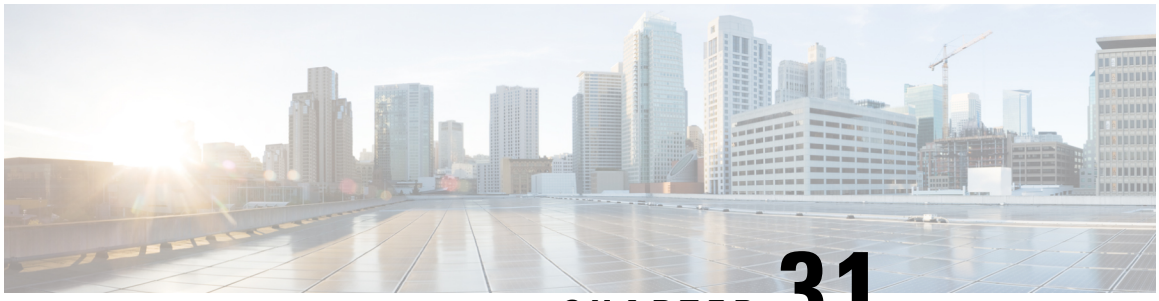
Table 82: Feature Information for Commanded Power Feature

Feature Name	Releases	Feature Information
DOCSIS 3.1 Commanded Power for US SC-QAMs	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Routers.

PART V

Layer 2 and Layer 3 VPN Configuration

- L2VPN Support over Cable, on page 511
- L2VPN Over Port-Channel, on page 527
- MPLS Pseudowire for Cable L2VPN, on page 531
- MPLS VPN Cable Enhancements, on page 563
- Multicast VPN and DOCSIS 3.0 Multicast QoS Support, on page 581
- EtherChannel for the Cisco CMTS, on page 593
- Flow-Based per Port-Channel Load Balancing, on page 603
- MPLS QoS via TLV for non-L2VPN Service Flow, on page 613



CHAPTER 31

L2VPN Support over Cable

The Layer 2 VPN (L2VPN) Support over Cable feature on the Cisco CMTS provides point-to-point Transparent LAN Service (TLS) in support of the Business Services over DOCSIS (BSOD) Cable Labs specification.

The L2VPN Support over Cable feature supports the following:

- The feature uses an Ethernet trunking interface to transport traffic for multiple L2VPNtunnels in support of different cable modems (CMs) and service flows (SFs) based on IEEE 802.1qVLAN IDs. For the legacy TLS service, only the primary upstream or downstream SFs are used. With the new L2VPNSupport over Cable feature, both primary and secondary SFs can be used.
- The TLS feature uses CLI to provision the service. The L2VPN Support over Cable feature uses the CM configuration file to provision the service, and a single CLI to identify the default Ethernet Network System Interface (NSI).
- Downstream traffic is forwarded on a per-CM basis and upstream traffic is forwarded on a per-SF basis. For L2VPN Support over Cable feature, upstream traffic for the same L2VPN can use multiple upstream service flows and downstream traffic can use different downstream service flows.

- Finding Feature Information, on page 511
- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 512
- Prerequisites for L2VPN Support over Cable, on page 512
- Restrictions for L2VPN Support over Cable, on page 513
- Information About L2VPN Support over Cable, on page 514
- Voice-Call Support on L2VPN CM, on page 517
- How to Configure L2VPN Support over Cable, on page 517
- Configuration Examples for L2VPN over Cable, on page 522
- Additional References, on page 524
- Feature Information for L2VPN Support over Cable, on page 526

Finding Feature Information

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 83: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for L2VPN Support over Cable

- You should use crypto-supported images.
- Cable modems must be configured to support BPI+.

Restrictions for L2VPN Support over Cable

The L2VPN Support over Cable feature has the following general restrictions:

- DOCSIS 1.0 CMs are not supported.
- Load balancing and Dynamic Channel Change (DCC) are not supported for CMs that are enabled for L2VPN support.
- DSx messages (Dynamic Service Add [DSA], Dynamic Service Change [DSC], and Dynamic Service Delete [DSD]) are supported for L2VPN-provisioned CMs. However, DSx with L2VPN type, length, values (TLVs) are not supported.
- Multipoint L2VPN is not supported, and any Simple Network Management Protocol (SNMP) MIBs for multipoint L2VPN are not supported.
- eSAFE (embedded Service/Application Functional Entities) DHCP snooping is not supported (L2VPN subtype 43.5.3)
- Maximum of 1024 L2VPNs are supported on a single MAC domain.
- Maximum of eight upstream SFs are supported per L2VPN service.
- Maximum of eight downstream classifiers are supported per L2VPN service.
- eSAFE exclusion is supported for only one eSAFE host. If the REG-REQ message for a compliant CM specifies multiple eSAFE hosts, then the eMTA (ifIndex 16) is selected as the eSAFE host to be excluded by the Cisco CMTS router. If the eMTA is not included as part of the capability of the CM, then the first eSAFE host in the capability is selected for exclusion.
- Maximum length of the Cable Modem Interface Mask (CMIM) is 4 bytes.
- Areas of the Business Services over DOCSIS (BSOD) Layer 2 Virtual Private Networks specification that are not supported are:
  - Vendor-specific L2VPN encodings for the replacement of the required VPN ID and NSI Encapsulation subtype are not supported.
  - Mapping of egress user priority to an NSI port transmission traffic class as specified by IEEE 802.1s is not supported.
  - Forwarding with non-zero default user priority values with vendor-specific configuration is not supported.
  - Accepting multiple Downstream Classifier L2VPN Encoding with the same VPN ID to clasify packets to different service flows is not supported.
  - Assigning multiple SAIDs to the same L2VPN on the same CM is not supported. The primary SAID is used for encrypting all downstream traffic.
  - Assigning of the same group-level L2VPN SAID to different CMs on the same MAC domain attached to the same L2VPN identifier is not supported.
  - Implementing the DOCSIS Spanning Tree Protocol (DSTP) and transmission of DSTP BPDUs on all NSI and RF interfaces configured for L2VPN operation is not supported.
  - Implementing a DSTP SAID specifically for DSTP forwarding to the customer premises equipment (CPE) ports of all L2VPN CMs is not supported.

VPN ID Restrictions

- A maximum of four VPN IDs are supported for each CM.
- A maximum of one VPN ID can be associated with each SF in a CM; although multiple SFs in a CM can belong to the same L2VPN.

- A maximum of 4093 unique VPN IDs are supported per Cisco CMTS router.
- The maximum length of a VPN ID is 16 bytes.
- All L2VPN encodings must contain a VPN ID, except for upstream classifier encodings.

Information About L2VPN Support over Cable

L2VPN Support Over Cable provides the following benefits and functions on a Cisco CMTS router:

- Supports point-to-point L2VPN forwarding mode.
- Supports up to four VPN IDs per CM.
- Supports multiple upstream SFs per CM, with one or more SFs belonging to the same VPN ID.
- Supports a single Ethernet NSI that serves as a trunking port for one or more L2VPN tunnels on the Cisco CMTS router.
- Supports BPI+ encryption using primary SAID of the CM.
- Supports L2VPN encodings in the CM configuration file and CM registration (REG-REQ with L2VPN encoding).
- Supports upstream L2VPN tunnel in support of per-CM and per-SF forwarding.
- Supports synchronization and recovery of the L2VPN database and upstream and downstream SFs during SUP NSF/SSO and N+1 line card redundancy switchovers.
- Supports QoS in upstream and downstream.
- Supports stacked IEEE 802.1q tags.
- Supports exclusion of traffic from the L2VPN tunnel for a single Embedded Service/Application Functional Entity (eSAFE) host.
- Supports Layer 2 classifier via CMIM and IEEE 802.1p priority bits.
- Supports detection of provisioning errors, such as duplicate VLAN IDs across CMs or existing VLAN IDs in use, and moves a CM offline with a corresponding error message.
- Supports coexistence of L2VPN and non-L2VPN traffic on the same RF MAC domain, with non-L2VPN traffic isolated from other tunnel traffic.
- Supports voice calls from L2VPN-provisioned CMs. However, voice calls are not part of the L2VPN.
- Supports BSOD VLAN Redundancy feature, which allows users to configure a backup WAN interface in addition to the primary WAN interface. When the primary WAN interface is down, the L2VPN traffic flows through the backup WAN interface.
- Supports manual switchover for VLAN Redundancy feature, which allows users to manually switch active uplink port from the current port to another port when both the uplink ports are up.
- Supports 2000 bytes layer 2 MTU.

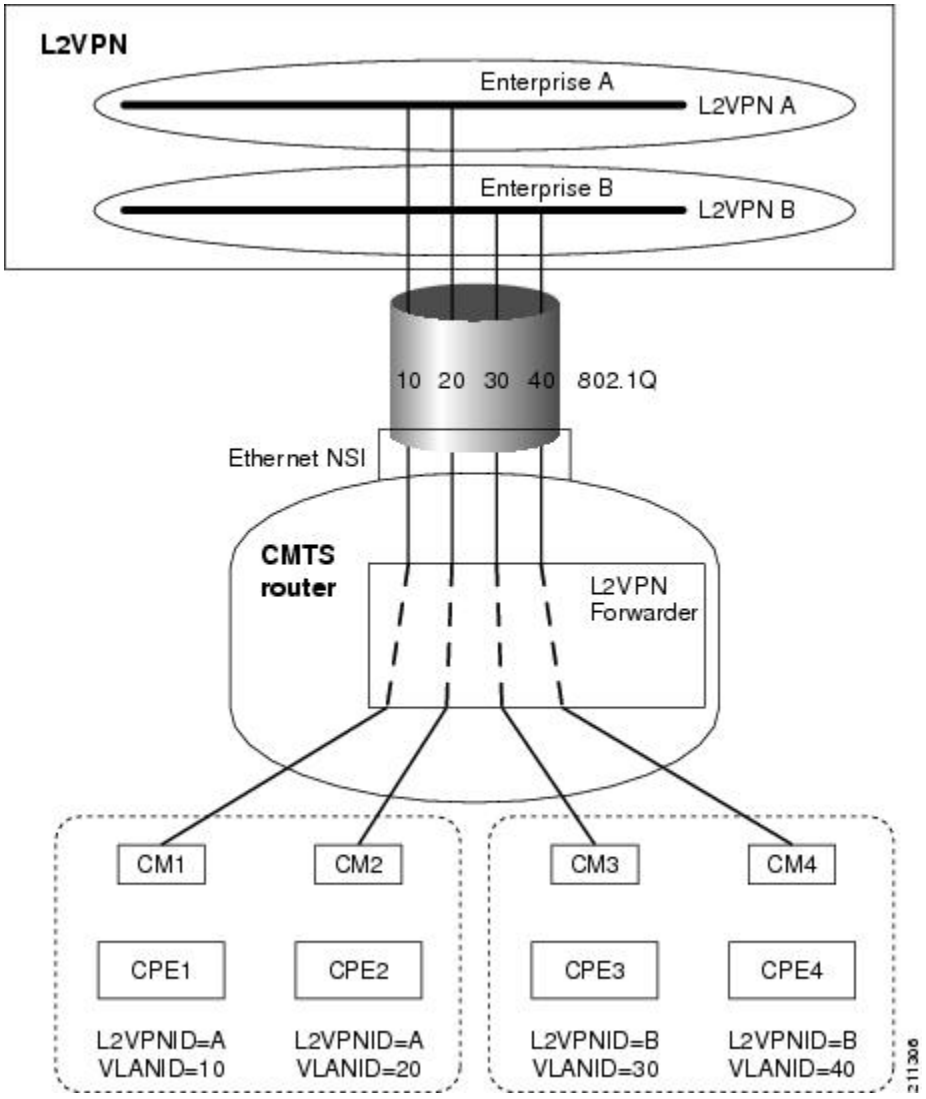
Point-to-Point L2VPN Forwarding Mode

The Cisco CMTS routers supports the point-to-point L2VPN forwarding mode described in the BSOD specification. Each attachment circuit (either SF or CM) on the Cisco CMTS router has a NSI encapsulation value, and is configured with an IEEE 802.1q VLAN ID.

The L2VPN forwarder on the Cisco CMTS router forwards both upstream and downstream traffic between the NSI port on the router and an attachment circuit without using MAC address learning for the forwarding decision. A L2VPN bridge on the backbone network of the cable operator performs the MAC-address learning to bridge packets between VLAN IDs.

The image below shows an example of a point-to-point L2VPN network using IEEE 802.1q NSI encapsulation. In this example, four CMs are associated with four different VLAN IDs: 10, 20, 30, and 40. The L2VPN encoding of the CM includes the logical L2VPN ID (in this case, A or B) with an NSI encapsulation subtype for IEEE 802.1q with the associated VLAN ID.

Figure 22: Point-to-Point L2VPN Network Diagram



The logical L2VPN IDs allow creation of separate broadcast domains for certain VLAN IDs. In the diagram, traffic for VLANs 10 and 20 from CM1 and CM2 can be sent to the network of Enterprise A, and traffic for VLAN's 30 and 40 from CM3 and CM4 can be sent to the network of Enterprise B.

L2VPN Encodings in the CM Configuration File

The CM configuration file contains a set of L2VPN encodings that control how the Cisco CMTS processes L2VPN forwarding of upstream and downstream CPE packets. As per the BSOD specification, the L2VPN encoding is encapsulated using a General Extension Information (GEI) encoding, which uses the type code 43 and subtype of 5 (43.5) with the reserved Vendor ID of 0xFFFFFF.

L2VPN defines the following types of encodings:

- Per-CM L2VPN encodings—An encoding that appears at the top level of the CM configuration file.
- Per-SF L2VPN Encoding—An encoding that appears as a subtype of the Upstream Service Flow Encoding (type 24).
- Upstream Classifier L2VPN Encoding—An encoding that appears in an Upstream Packet Classification Configuration Setting (type 22).
- Downstream Classifier L2VPN Encoding—An encoding that appears in a Downstream Packet Classification Configuration Setting (type 23).

The simplest CM configuration file has a single per-SF L2VPN Encoding within the primary upstream SF definition and a single per-CM L2VPN Encoding with a NSI Encapsulation subtype for that L2VPN.

Note When BSOD (CM configuration file) is used for L2VPN configuration, and QoS policy-map settings are applied to Cisco CMTS WAN interfaces, the packets do not match the QoS policy-map. When CLI mode is used for L2VPN configuration, and QoS policy-map settings are applied to Cisco CMTS WAN interfaces, the packets will match the QoS policy-map first.

Note Cisco CMTS supports BSOD VLAN redundancy feature with support for two Ethernet Network Side Interface (NSI) configuration and a backup WAN interface. When the active NSI WAN interface is down, the L2VPN traffic flows through the backup WAN interface.

Supported L2VPN Encodings

This section describes the supported L2VPN encodings in the CM configuration file that are supported by the Cisco CMTS routers.

- The Cisco CMTS routers support the following CM capabilities:
  - L2VPN capability (5.17)
  - eSAFE host capability (5.18)
  - Downstream Unencrypted Traffic (DUT) filtering (5.19)
- The Cisco CMTS routers support the following top-level encodings:
  - VPN identifier (43.5.1)
  - CMIM (43.5.4)—When provided, applies to all upstream SFs associated with an L2VPN tunnel; Supports only one eSAFE host.
  - NSI encapsulation (43.5.2) with format code 2 for IEEE 802.1q (43.5.2.2)
  - DUT filtering encoding
- The Cisco CMTS routers support the following per-SF encodings:
  - VPN identifier (43.5.1)

- Ingress user priority (43.5.8)

- The Cisco CMTS routers support the following downstream classifier encodings:
  - VPN identifier (43.5.1)
  - CMIM (43.5.4) and (22/23.13)
  - User priority range (43.5.9)

For more information about the CM configuration file and L2VPN encodings, see the "Business Services over DOCSIS (BSOD) Layer 2 Virtual Private Networks" specification.

For information about how to use the configuration file generator on the Cisco CMTS, see the “DOCSIS Internal Configuration File Generator for the Cisco CMTS” document.

Voice-Call Support on L2VPN CM

Voice calls are supported on L2VPN CMs. This feature enables the Cisco CMTS routers to support dynamic service flows on L2VPN-provisioned cable modems to permit voice calls from a non-L2VPN CPE.

To provide voice-call support on a L2VPN CM, you have to configure correct classifiers and create two static service flows (primary and secondary) using the cable modem configuration file. If the eMTA is L2VPN-capable with the embedded CPE configured as an eSAFE host, then only one service flow is required. When correct CMIM bits are configured, the Cisco CMTS does not send packets from the eSAFE host to the L2VPN.

Though the L2VPN can be configured on the primary or secondary service flow, it cannot coexist with eMTAs on the same service flow. The eMTAs should always use a different service flow from that of L2VPN. The classifiers to direct the traffic should also be based on the service flows the L2VPN and eMTAs are using. When the above configuration is in place, the dynamic service flows are created automatically whenever voice calls are initiated.

How to Configure L2VPN Support over Cable

This section contains the following procedures:

Configuring the Ethernet Network System Interface

To configure the L2VPN Support over Cable feature, you need to specify an Ethernet NSI to operate as the trunking interface for the L2VPN traffic. You must configure the NSI using a command on the Cisco CMTS router. It is not configurable through the CM configuration file.

Before you begin

The following interface types can be configured as an NSI for L2VPN Support over Cable:

- Cisco cBR Series Converged Broadband Router—GigabitEthernet and TenGigabitEthernet

Note The Cisco CMTS routers only support the configuration of a single L2VPN NSI per CMTS.

>

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable l2-vpn-service xconnect nsi dot1q interface***ethernet-intf*[**backup-interface ethernet-intf**] **Example:** `Router(config)# cable l2-vpn-service xconnect nsi dot1q interface Te4/1/0 backup-interface Te4/1/4`	Configures WAN interface for DOT1Q L2VPN . (Optional) Backup-interface - If backup-interface is configured it means that BSoD VLAN redundancy feature is enabled.

Preparing the DOCSIS Configuration File for L2VPN Support

To support L2VPN, the DOCSIS configuration file must be configured with the appropriate encodings. For information about the supported encodings by the Cisco CMTS routers, see the L2VPN Encodings in the CM Configuration File, on page 515.

Manual Switchover Command Line Interface

For BSoD VLAN Redundancy feature, users can manually switch active uplink ports from the active port to another port when both the uplink ports are up through the command line interface. To manually switchover, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted

	Command or Action	Purpose
Step 2	**cable l2-vpn dot1q-nsi-redundancy force-switchover from** *active-nsi-interface* **Example:** `Router# cable l2-vpn dot1q-nsi-redundancy force-switchover from Te4/0/1`	Switches the active uplink port from the current active port to the specified port.

To display the dot1q L2VPN uplink redundancy information, use the **show cable l2-vpn dot1q-nsi-redundancy** as shown in the following example:

```
Router# show cable l2-vpn dot1q-nsi-redundancy
Primary-NSI Backup-NSI Active-NSI Elapsed-after-SW
Te4/1/0       Te4/0/4    Te4/1/0    31m9s
Te4/1/2       Te4/0/5    Te4/1/2    59s
```

Verifying L2VPN Support over Cable

To verify L2VPN information on the Cisco CMTS router, use the **show cable l2-vpn xconnect dot1q-vc-map** command.

Procedure

Step 1 To display VLAN information for all cable modems, use the **show cable l2-vpn xconnect dot1q-vc-map** command as shown in the following example:

Example:

```
Router# show cable l2-vpn xconnect dot1q-vc-map
MAC Address    Ethernet Interface      VLAN ID   Cable Intf  SID  Customer Name/VPN ID
0014.f8c1.fd66 GigabitEthernet4/0/0        68    Cable6/0/0  3    0234560001
```

Step 2 To display VLAN information for a particular L2VPN ID or customer, use the **show cable l2-vpn xconnect dot1q-vc-map customer** form of the command as shown in the following example:

Example:

```
Router# show cable l2-vpn xconnect dot1q-vc-map customer 0234560001

MAC Address    Ethernet Interface      VLAN ID   Cable Intf  SID  Customer Name/VPNID
0014.f8c1.fd66 GigabitEthernet4/0/0        68    Cable6/0/0  3    0234560001
```

Step 3 To display information for a particular L2VPN ID on a specific cable modem, use the **show cable l2-vpn xconnect dot1q-vc-map vpn** form of the command along with specification of the cable modem MAC address, as shown in the following example:

Example:

```
Router# show cable l2-vpn xconnect dot1q-vc-map 0014.f8c1.fd66 vpn 0234560001

MAC Address    Ethernet Interface      VLAN ID   Cable Intf  SID  Customer Name/VPNID
0014.f8c1.fd66 GigabitEthernet4/0/0        68    Cable6/0/0  3    0234560001
```

Step 4 To display detailed information for a particular L2VPN ID on a specific cable modem, use the **show cable l2-vpn xconnect dot1q-vc-map vpn verbose** form of the command along with specification of the cable modem MAC address, as shown in the following example:

Example:

```
Router# show cable l2-vpn xconnect dot1q-vc-map 0014.f8c1.fd66 vpn 0234560001 verbose
MAC Address                       : 0014.f8c1.fd66
Prim Sid                          : 3
Cable Interface                   : Cable6/0/0
VPN ID                            : 0234560001
L2VPN SAID                        : 12294
Upstream SFID                     : 23
Downstream CFRID[SFID]            : 2[24]
CMIM                              : 0x60
Ethernet Interface                : GigabitEthernet4/0/0
DOT1Q VLAN ID                     : 68
Total US pkts                     : 1372
Total US bytes                    : 500226
Total US pkt Discards             : 0
Total US byte Discards            : 0
Total DS pkts                     : 1248
Total DS bytes                    : 415584
Total DS pkt Discards             : 0
Total DS byte Discards            : 0
```

Step 5 To display detailed information and the current redundancy information for a particular cable modem, use the the **show cable l2-vpn xconnect dot1q-vc-map verbose** form of the command along with specification of the cable modem MAC address, as shown in the following example:

Example:

```
Router# show cable l2-vpn xconnect dot1q-vc-map 0014.f8c1.fd66 verbose
MAC Address                       : 5039.5589.4302
Prim Sid                          : 45
Cable Interface                   : Cable6/0/2
L2VPNs provisioned                : 1
DUT Control/CMIM                  : Disable/0x8000FFFF

VPN ID                            : 000234560001
L2VPN SAID                        : 45
Upstream SFID Summary             : 77
Upstream SFID [77   ]             : SID 45
Downstream CFRID[SFID] Summary    : Primary SF
CMIM                              : 0x60
Primary Ethernet Interface        : GigabitEthernet4/0/0
Backup  Ethernet Interface        : GigabitEthernet4/0/1
Active  Ethernet Interface        : GigabitEthernet4/0/0
DOT1Q VLAN ID                     : 207
Total US pkts                     : 151269
Total US bytes                    : 211755224
Total DS pkts                     : 150502
Total DS bytes                    : 210463324
```

Step 6 To display the dot1q L2VPN uplink redundancy information, use the **show cable l2-vpn dot1q-nsi-redundancy** as shown in the following example:

Example:

```
Router# show cable l2-vpn dot1q-nsi-redundancy
Primary-NSI Backup-NSI Active-NSI Elapsed-after-SW
```

```
Te4/1/0       Te4/0/4    Te4/1/0    31m9s
Te4/1/2       Te4/0/5    Te4/1/2    59s
```

Enabling Voice-Call on a L2VPN CM

You can enable the Voice-Call Support on a L2VPN CM feature by registering a cable modem with a SID to VPN mapping cable modem configuration file (MPLS or 802.1q).

- If the L2VPN is on the primary service flow, you should use a cable modem configuration file with static secondary service flow and the classifiers should be configured on the secondary service flow for non-L2VPN packets.
- If the L2VPN is on the secondary service flow, then classifiers should be configured for L2VPN packets.

Note The cable modem configuration file based L2VPN configuration provides the flexibility to configure L2VPN on the primary or secondary service flow. However, we recommend that you configure L2VPN on the secondary service flow and the primary service flow is used for the default traffic.

Note In a CLI-based L2VPN configuration, the L2VPN is on the primary service flow; therefore the static secondary service flow should be used for the eMTAs.

Verifying Dynamic Service Flows

To verify dynamically created service flows on the Cisco CMTS router, use the **show interface cable service-flow** command.

Note To verify information about PacketCable operations, use **show packetcable** commands.

```
Router# show interface cable 5/1/0 service-flow
Sfid : 30191
Mac Address : 000a.739e.140a
Type : Secondary(Dynamic)
Direction : Upstream
Current State : Active
Current QoS Indexes [Prov, Adm, Act] : [0, 24, 24]
Active Time : 00:55
Sid : 7140
Admitted QoS Timeout : 200 seconds
Active QoS Timeout : 0 seconds
Packets : 1824
Bytes : 466944
Rate Limit Delayed Grants : 0
Rate Limit Dropped Grants : 0
Current Throughput : 68356 bits/sec, 32 packets/sec
Classifiers:
Classifier Id : 41
```

```
Service Flow Id : 30191
CM Mac Address : 000a.739e.140a
Direction : upstream
Activation State : active
Classifier Matching Priority : 128
PHSI : 1
Number of matches : -
IP Classification Parameters:
IP Source Address : 10.8.230.3
Source IP Address Mask : 255.255.255.255
Destination IP Address : 172.16.2.35
Destination IP Address Mask : 255.255.255.255
IP Protocol Type : 17
Source Port Low : 53456
Source Port High : 53456
Destination Port Low : 7052
Destination Port High : 7052
```

Configuration Examples for L2VPN over Cable

This section provides configuration examples for the L2VPN over Cable feature:

Example: Specifying the Ethernet NSI Interface

You can specify the Ethernet NSI within the CM configuration file, or using the **cable l2-vpn-service xconnect** global configuration command as shown in the following example:

```
cable l2-vpn-service xconnect nsi {dot1q|mpls}
```

Example: Enabling Voice Call Support on MPLS L2VPN

The following is a sample cable modem configuration file that enables voice call support on MPLS L2VPN. In this example the L2VPN is applied to the primary service flow.

```
03 (Net Access Control)         = 1
18 (Maximum Number of CPE)      = 16
43 (Vendor Specific Options)
   S08 (Vendor ID)      = ff ff ff
   S005 (Unknown sub-type)      = 01 04 32 30 32 30 02 07 04 05 01 0a 4c 02 01 2b 06 26 04
 00 00 01 90
22 (Upstream Packet Classification Encoding Block)
   S01 (Classifier Reference)           = 2
   S03 (Service Flow Reference)         = 2
   S09 (IP Packet Encodings)
       T03 (IP Source Address)          = 050 001 005 000
       T04 (IP Source Mask)             = 255 255 255 000
22 (Upstream Packet Classification Encoding Block)
   S01 (Classifier Reference)           = 3
   S03 (Service Flow Reference)         = 2
   S10 (Ethernet LLC Packet Classification Encodings)
       T02 (Source MAC Address)         = 00 e0 f7 5a c9 21
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference)           = 21
   S03 (Service Flow Reference)         = 21
```

```
   S05 (Rule Priority)                  = 5
   S09 (IP Packet Encodings)
       T05 (IP Destination Address)     = 050 001 005 000
       T06 (IP Destination Mask)        = 255 255 255 000
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference)           = 22
   S03 (Service Flow Reference)         = 21
   S05 (Rule Priority)                  = 5
   S10 (Ethernet LLC Packet Classification Encodings)
       T01 (Destination MAC Address)    = 00 e0 f7 5a c9 21 ff ff ff ff ff ff
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference)         = 1
   S06 (QoS Parameter Set Type)         = 7
   S43 (Vendor Specific Options)
       T08 (Vendor ID)  = ff ff ff
       T005 (Unknown sub-type)  = 01 04 32 30 32 30
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference)         = 2
   S06 (QoS Parameter Set Type)         = 7
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference)         = 20
   S06 (QoS Parameter Set Type)         = 7
   S07 (Traffic Priority)               = 0
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference)         = 21
   S06 (QoS Parameter Set Type)         = 7
   S07 (Traffic Priority)               = 1
29 (Privacy Enable)             = 1
```

Example: Enabling Voice Call Support on 802.1q L2VPN

The following is a sample cable modem configuration file that enables voice call support on 802.1q L2VPN. In this example the L2VPN is applied to the secondary service flow.

```
03 (Net Access Control)         = 1
43 (Vendor Specific Options)
   S08 (Vendor ID)      = ff ff ff
   S005 (Unknown sub-type)      = 01 05 02 34 56 00 01 02 04 02 02 00 44
18 (Maximum Number of CPE)      = 16
22 (Upstream Packet Classification Encoding Block)
   S01 (Classifier Reference)           = 2
   S03 (Service Flow Reference)         = 2
   S10 (Ethernet LLC Packet Classification Encodings)
       T02 (Source MAC Address)         = 00 e0 14 e3 23 1c
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference)           = 4
   S03 (Service Flow Reference)         = 4
   S43 (Vendor Specific Options)
       T08 (Vendor ID)  = ff ff ff
       T005 (Unknown sub-type)  = 01 05 02 34 56 00 01
   S11 (IEEE 802.1P/Q Packet Classification Encodings)
       T01 (IEEE 802.1P UserPriority)   = 00 07
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference)         = 1
   S06 (QoS Parameter Set Type)         = 7
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference)         = 2
   S06 (QoS Parameter Set Type)         = 7
   S43 (Vendor Specific Options)
       T08 (Vendor ID)  = ff ff ff
       T005 (Unknown sub-type)  = 01 05 02 34 56 00 01 08 01 01
25 (Downstream Service Flow Encodings)
```

```
   S01 (Service Flow Reference)         = 3
   S06 (QoS Parameter Set Type)         = 7
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference)         = 4
   S06 (QoS Parameter Set Type)         = 7
```

Example: Enabling Voice Call Support on CLI-based L2VPN

The following is a sample cable modem configuration file that enables voice call support on L2VPN configured using CLI. L2VPN configured using the CLI is always applied to the primary service flow.

```
03 (Net Access Control)         = 1
18 (Maximum Number of CPE)      = 16
22 (Upstream Packet Classification Encoding Block)
   S01 (Classifier Reference)           = 2
   S03 (Service Flow Reference)         = 2
   S09 (IP Packet Encodings)
       T03 (IP Source Address)          = 050 001 005 000
       T04 (IP Source Mask)             = 255 255 255 000
22 (Upstream Packet Classification Encoding Block)
   S01 (Classifier Reference)           = 3
   S03 (Service Flow Reference)         = 2
   S10 (Ethernet LLC Packet Classification Encodings)
       T02 (Source MAC Address)         = 00 e0 f7 5a c9 21
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference)           = 21
   S03 (Service Flow Reference)         = 21
   S05 (Rule Priority)                  = 5
   S09 (IP Packet Encodings)
       T05 (IP Destination Address)     = 050 001 005 000
       T06 (IP Destination Mask)        = 255 255 255 000
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference)           = 22
   S03 (Service Flow Reference)         = 21
   S05 (Rule Priority)                  = 5
   S10 (Ethernet LLC Packet Classification Encodings)
       T01 (Destination MAC Address)    = 00 e0 f7 5a c9 21 ff ff ff ff ff ff
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference)         = 1
   S06 (QoS Parameter Set Type)         = 7
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference)         = 2
   S06 (QoS Parameter Set Type)         = 77
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference)         = 20
   S06 (QoS Parameter Set Type)         = 7
   S07 (Traffic Priority)               = 0
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference)         = 21
   S06 (QoS Parameter Set Type)         = 7
   S07 (Traffic Priority)               = 1
29 (Privacy Enable)             = 1
```

Additional References

The following sections provide references related to the L2VPN Support over Cable feature.

Standards

Standard	Title
CM-SP-BPI+-I12-050812	*Baseline Privacy Plus Interface Specification* http://www.cablelabs.com/wp-content/uploads/specdocs/CM-SP-BPI+-C01-081104.p
CM-SP-L2VPN-I03-061222	*Business Services over DOCSIS (BSOD) Layer 2 Virtual Private Networks* http://www.cablelabs.com/wp-content/uploads/specdocs/CM-SP-L2VPN-I12-131120.
CM-SP-RFIv2.0-I11-060602	*Radio Frequency Interface Specification* http://www.cablelabs.com/wp-content/uploads/specdocs/CM-SP-RFIv2.0-C02-090422.
IEEE 802.1ad	*IEEE 802.1ad-2005 IEEE Standards for Local and metropolitan area networks*— Virt Bridged Local Area Networks http://www.ieee.org
IEEE 802.1q	*IEEE Std 802.1Q Virtual Bridged Local Area Networks* http://www.ieee.org

MIBs

MIB	MIBs Link
DOCS-L2VPN-MIB	To locate and download MIBs for selected platforms, Cisco IOS-XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://tools.cisco.com/ITDIT/MIBS/servlet/index

RFCs

RFC	Title
RFC 2685	Virtual Private Networks Identifier http://www.ietf.org/rfc/rfc2685.txt
RFC 4364	*BGP/MPLS IP Virtual Private Networks (VPNs)* http://www.ietf.org/rfc/rfc4364.txt

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

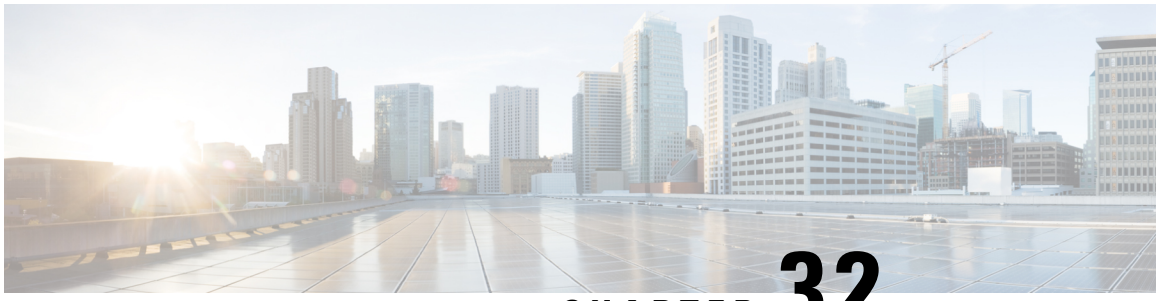
Feature Information for L2VPN Support over Cable

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 84: Feature Information for L2VPN Support Over Cable

Feature Name	Releases	Feature Information
L2VPN support over cable	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router .



CHAPTER 32

L2VPN Over Port-Channel

The Layer 2 VPN (L2VPN) over port-channel feature supports IEEE 802.1Q (dot1q) L2VPN WAN interface port-channel. Using this feature, you can configure the dot1q L2VPN traffic to pass through port-channel uplink

Contents

- Information About L2VPN Over Port-Channel, on page 527
- How to Configure the L2VPN Over Port-Channel, on page 528
- Verifying Port-Channel Configuration, on page 528
- Feature Information for L2VPN Over Port-Channel, on page 529

Information About L2VPN Over Port-Channel

The Cisco cBR-8 supports L2VPN, where the Ethernet frames from the cable modem are cross connected to a specific VLAN interface. The VLAN ID to be inserted is specified. With the L2VPN over port-channel feature, you can now support port-channel uplink interface as well as the 10 Gb uplink interface.

TLS L2VPN

For the Transparent LAN Service (TLS) L2VPN, the dot1q maps contain the cable modem MAC address, the VLAN ID, and the outbound interface. Traffic received from a specific cable modem is tagged with a VLAN ID and is sent out from the uplink interface.

DOCSIS L2VPN

For the Data-over-Cable Service Interface Specifications (DOCSIS) L2VPN, cable modem (CM) configuration file holds the L2VPN encodings for both, the CM and the service flow. At the CMTS level you have to specify the default port-channel Network Side Interface (NSI). L2VPN encodings are passed by the CM to the CMTS during registration. The CMTS installs DOCSIS service flow VLAN mapping based on the information passed to it during the registration. For upstream traffic, the CMTS sends the dot1q VLAN tagged traffic out from the uplink interface. On downstream, the CMTS receives the dot1q tagged traffic from the aggregator. The CMTS replaces the VLAN header with a DOCSIS header to the corresponding service flow.

Benefits of L2VPN Over Port-Channel

By using the dot1q L2VPN, you can utilize the port-channel interface feature instead of a single 10 Gb port.

Restrictions for L2VPN Over Port-Channel

The CMTS dot1q L2VPN is designed to support traffic from customer premises equipment to the network or verse vice. For CMTS L2VPN NSI port, port-channel interface does not support VLAN redundancy.

How to Configure the L2VPN Over Port-Channel

This section describes how to configure L2VPN over port-channel on the Cisco cBR-8.

Configuring the Port-Channel Uplink Port for TLS L2VPN

For TLS L2VPN, you must configure the overall enable CLI and the dot1q map. In dot1q map, you have to designate the port-channel uplink port.

To configure the port-channel uplink port for TLS L2VPN, complete the following procedure:

```
cable l2-vpn-service xconnect nsi dot1q
cable dot1q-vc-map mac address port-channel number vlan id custom name
```

Configuring the Port-Channel Uplink Port for DOCSIS L2VPN

For DOCSIS L2VPN, you only have to configure the overall enable CLI with port-channel uplink port. The other L2VPN related parameters are setup by the CM configuration file type-length-value parsing.

To configure the port-channel uplink port for DOCSIS L2VPN, complete the following procedure:

```
configure terminal
cable l2-vpn-service xconnect nsi dot1q interface port-channel number
```

Verifying Port-Channel Configuration

Verify the Port-Channel Mapping

To verify the port-channel mapping, use the **show cable l2-vpn xconnect dot1q-vc-map** command as shown in the example below:

```
show cable l2-vpn xconnect dot1q-vc-map

MAC Address    Ethernet Interface      VLAN ID   Cable Intf  SID  Customer Name/VPNID
c8fb.26a5.551c Port-channel64            1200    Cable6/0/0  17   Topgun
```

View the Port-Channel Interface

To view the port-channel interface, use the **show cable l2-vpn xconnect dot1q-vc-map verbose** command as shown in the example below:

```
show cable l2-vpn xconnect dot1q-vc-map c8fb.26a5.551c verbose

MAC Address                          : c8fb.26a5.551c
Customer Name                        : ats
Prim Sid                             : 17
Cable Interface                      : Cable6/0/0
Ethernet Interface                   : Port-channel64
DOT1Q VLAN ID                        : 1200
Total US pkts                        : 189
Total US bytes                       : 18200
Total DS pkts                        : 615
Total DS bytes                       : 39360
```

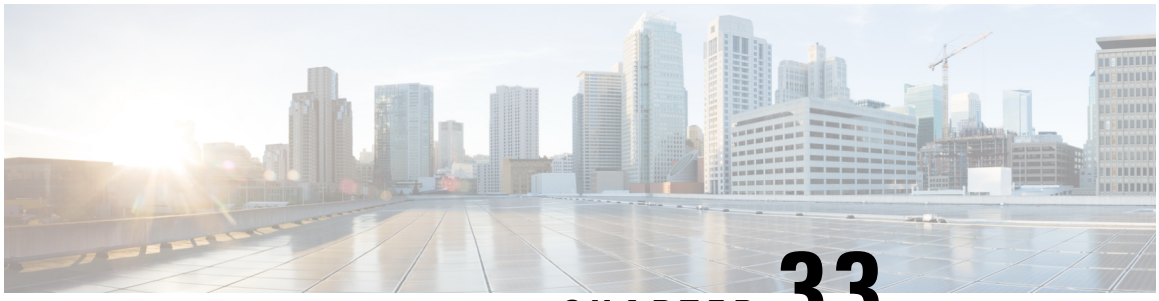
Feature Information for L2VPN Over Port-Channel

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 85: Feature Information for L2VPN Over Port-Channel

Feature Name	Releases	Feature Information
L2VPN over port-channel	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router .



CHAPTER 33

MPLS Pseudowire for Cable L2VPN

The Multiprotocol Label Switching (MPLS) Pseudowire for Cable Layer 2 Virtual Private Network (L2VPN) feature enables service providers to use a single, converged, Internet Protocol (IP)/MPLS network infrastructure to offer Ethernet data link layer (Layer 2) connectivity to two or more VPN customer sites.

- Finding Feature Information, on page 531
- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 532
- Prerequisites for MPLS Pseudowire for Cable L2VPN, on page 532
- Restrictions for MPLS Pseudowire for Cable L2VPN, on page 533
- Information About MPLS Pseudowire for Cable L2VPN, on page 533
- L2VPN Pseudowire Redundancy, on page 537
- MPLS Pseudowire Provisioning Methods, on page 537
- How to Enable MPLS on a Cisco CMTS Router, on page 541
- How to Provision MPLS Pseudowires, on page 546
- How to Configure L2VPN Pseudowire Redundancy, on page 547
- Configuration Examples for MPLS Pseudowire for Cable L2VPN, on page 550
- Verifying the MPLS Pseudowire Configuration, on page 557
- Additional References, on page 560
- Feature Information for MPLS Pseudowire for Cable L2VPN, on page 561

Finding Feature Information

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 86: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for MPLS Pseudowire for Cable L2VPN

- Enable Baseline Privacy Interface Plus (BPI+) to provide a simple data encryption scheme to protect data sent to and from cable modems in a data over cable network.
- Enable Cisco Express Forwarding (CEF) to optimize network performance.
- Ensure that the primary and backup pseudowires on the remote provider edge (PE) routers have the same pseudowire type as the Cisco cable modem termination system (CMTS).
- Create the remote pseudowire using a pw-class with VLAN as the interworking for remote PEs, if the CMTS is using VLAN as pseudowire type.

Restrictions for MPLS Pseudowire for Cable L2VPN

The following are the general restrictions for the MPLS Pseudowire for Cable L2VPN feature:

- Supports only Ethernet over MPLS (EoMPLS) pseudowires per RFC 4448.
- Supports only point-to-point forwarding. Ethernet switching is not supported.
- Requires DOCSIS 2.0, 3.0 and 3.1-certified cable modems (CMs). This feature is not supported on DOCSIS 1.0-certified cable modems.
- Supports a maximum of four VPNs per cable modem.
- Supports a maximum of eight upstream service flows and eight downstream classifiers.
- Supports a maximum of 16000 EoMPLS pseudowires per Cisco CMTS router.
- Requires the backup pseudowire to be up on the remote PE for the Cisco CMTS to switchover.
- Requires the backup pseudowire to become active on the Cisco CMTS only after the primary pseudowire fails.

Note The CLI-based (static provisioning) L2VPN supports traffic forwarding to VPN only on primary upstream and downstream service flows. Hence only primary upstream and downstream service flows must be configured in the cable modem configuration file.

Information About MPLS Pseudowire for Cable L2VPN

The MPLS Pseudowire for Cable L2VPN feature enables Ethernet-based Layer 2 VPN service over an MPLS network by encapsulating and transmitting the Layer 2 protocol data units (PDUs) over pseudowires (PWs). This feature enables service providers to offer site-to-site connectivity to their business and enterprise customers.

Layer 2 services emulated over an MPLS network are commonly referred to as MPLS-based L2VPNs or MPLS L2VPNs. Subsequently, Ethernet service emulated over an MPLS network is referred to as Ethernet over MPLS (EoMPLS) service.

The MPLS Pseudowire for Cable L2VPN feature is fully compliant with CableLabs Business Services over DOCSIS (BSOD) L2VPN specification, and is an extension to the existing DOCSIS L2VPN features supported on Cisco CMTS routers.

The MPLS Pseudowire for Cable L2VPN feature provides the following capabilities:

- Transport Ethernet frames over an MPLS network.
- Handle a DOCSIS service flow as an attachment circuit that is mapped to an EoMPLS pseudowire.
- Enable the Cisco CMTS router to be the MPLS provider edge (PE) router.
- Enable forwarding of Ethernet frames over DOCSIS (between a CM and a Cisco CMTS router) to MPLS (towards Metropolitan Area Network or Wide Area Network).
- Provide a common framework to encapsulate and transport supported Layer 2 traffic types over an MPLS network.

The MPLS Pseudowire for Cable L2VPN feature differs from the existing DOCSIS L2VPN features such as 802.1q-based L2VPN (L2VPN Support over Cable). The MPLS Pseudowire for Cable L2VPN feature uses IP/MPLS network to transport layer 2 protocol data units (PDUs), whereas 802.1q-based L2VPN feature uses layer 2 Ethernet network to transport PDUs.

How MPLS Transports Layer 2 Packets

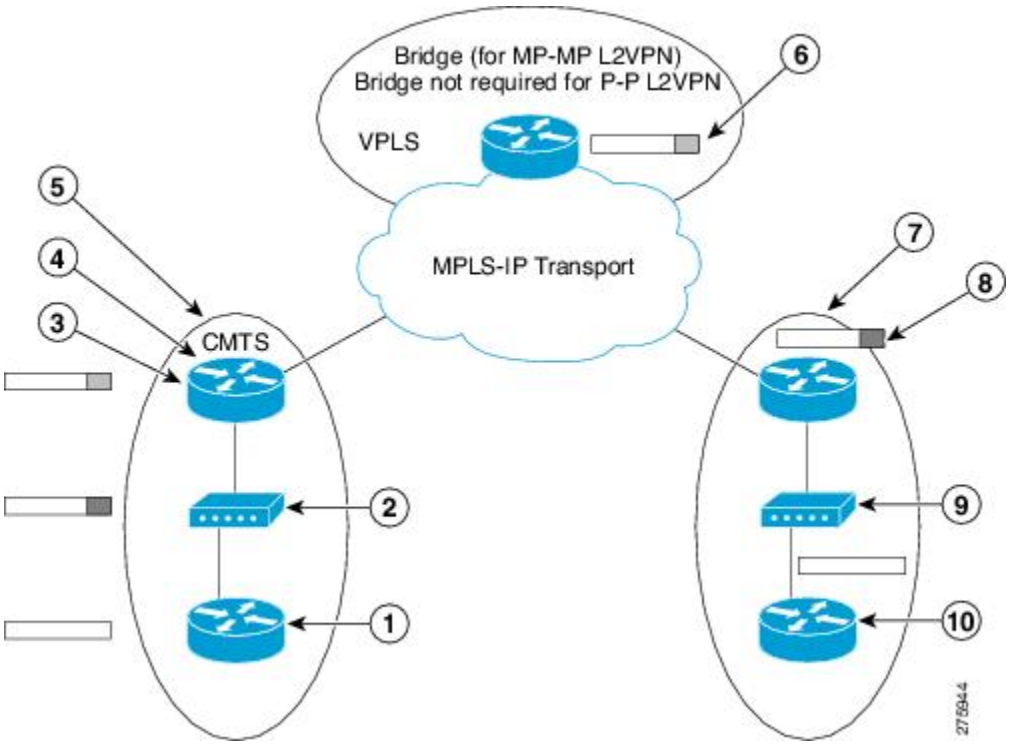
The MPLS subsystem removes DOCSIS encapsulation for Layer 2 Ethernet frames and adds MPLS labels at the ingress provider edge (PE) Cisco CMTS router. Then, the MPLS subsystem sends resulting MPLS packets to the corresponding PE router at the other end of the pseudowire. The PE routers must be configured for successful transmission of IP/MPLS packets between the two PE routers.

The cable modem classifies Ethernet frames from the customer premise equipment (CPE) in the upstream direction using upstream classifiers. Then, a DOCSIS header is added to these frames, and they are sent on a given upstream service flow with a different service identifier. On the Cisco CMTS router, the upstream packet is classified as an L2VPN packet based on the cable interface and service identifier. The Cisco CMTS router removes the DOCSIS header and adds an MPLS header. An MPLS header contains two MPLS labels: the outer label corresponding to the remote PE router and the inner label corresponding to the pseudowire label. The Cisco CMTS router forwards the MPLS packet towards the remote PE router, which is the other end of the pseudowire, over the MPLS network.

In the downstream direction, the Cisco CMTS router receives MPLS packets having only one MPLS header that contains the label that the Cisco CMTS router previously allocated for the corresponding EoMPLS pseudowire. The Cisco CMTS router uses the MPLS label to identify one of the L2VPN cable modems. Then, the Cisco CMTS router classifies the MPLS packet using the L2VPN downstream classifiers based on MPLS experimental (MPLS-EXP) bits in the MPLS header of the received MPLS packet, and removes the MPLS header. Then, the Cisco CMTS router sends the packet on the classified downstream service flow by adding the DOCSIS header. The cable modem then removes the DOCSIS header and delivers the Ethernet frame to the CPE.

A unique combination of a cable modem MAC address, VPN ID (if present in the CM configuration file), peer IP address, and a virtual circuit ID (VCID) identifies the MPLS pseudowire on the Cisco CMTS router.

Figure 23: Transporting Layer 2 Packets



The table illustrates how MPLS transports Layer 2 packets in a DOCSIS-based cable communications system.

1	A router sends an untagged Ethernet frame.	6	MPLS packets are label switched.
2	A CM adds a DOCSIS header to the frame.	7	The Cisco CMTS router receives an MPLS packet and looks up the MPLS forwarding table using the label value in the MPLS header.
3	The Cisco CMTS router removes the DOCSIS header from the frame.	8	The Cisco CMTS router replaces the MPLS header with DOCSIS header (containing the right SID value).
4	The Cisco CMTS router looks up the Service ID (SID) database using the SID value from the DOCSIS header and finds the MPLS header.	9	The DOCSIS header is removed.
5	The Cisco CMTS router adds the MPLS header to the frame.	10	The Ethernet frame is delivered untagged.

Supported Ethernet Encapsulation on UNI

The Ethernet User-Network Interface (UNI) is the connection between a cable modem and a customer premise equipment such as a router or a switch. The service provider may or may not use any encapsulation on the UNI.

The MPLS Pseudowire for Cable L2VPN feature supports the following transport types on an Ethernet UNI:

- Port-based UNI (independent of any VLAN)—The port-based UNI provides Metro Ethernet Forum (MEF)-defined Ethernet Private Line (EPL) service. In this transport type, an MPLS pseudowire is mapped to the Ethernet port.
- VLAN-based UNI—Ethernet VLAN using 802.1q encapsulation (including stacked VLANs). The VLAN-based UNI provides MEF-defined Ethernet Virtual Private Line (EVPL) service. In this transport type, the MPLS pseudowire is mapped to the 802.1q VLAN.

Note The Ethernet UNI must be attached to the Ethernet port of a cable modem.

Before configuring this feature, you should understand the following concepts:

MPLS Pseudowire

Pseudowire is a point-to-point Layer 2 connection between two PE routers. The MPLS Pseudowire for Cable L2VPN feature supports the following pseudowire types:

- Type-4 pseudowire—This is used to transport only VLAN tagged Layer 2 Ethernet frames.
- Type-5 pseudowire—This is used to transport VLAN tagged and untagged Layer 2 Ethernet frames. This is the default pseudowire type.

Bundle254 Interface

The bundle254 (Bu254) interface is an internal bundle interface on a Cisco CMTS router that is used as a circuit identifier for all MPLS pseudowires. This internal bundle interface is created automatically on a Cisco CMTS router when you enable the MPLS pseudowire functionality using the **cable l2-vpn-service xconnect** command. Only one Bu254 interface is created to handle all the MPLS pseudowires available on the Cisco CMTS router.

The output of the **show xconnect or show cable l2-vpn xconnect command displays** the circuit identifier created by the Cisco CMTS router for all the MPLS pseudowires**.**

Ingress Process

When an upstream packet received from a cable interface of the Cisco CMTS router is identified as an L2VPN packet based on the cable modem interface and Service ID (SID), the packet goes through the ingress process. The ingress process ensures that the DOCSIS header is removed, and an MPLS label header is added to the packet according to the MPLS pseudowire configuration and the packet is sent out from the Ethernet interface of the Cisco CMTS router. The ingress process is also known as the label imposition process.

Egress Process

When a downstream packet received from an Ethernet interface of the Cisco CMTS router is identified as an L2VPN packet by the innermost MPLS label, the packet goes through the egress process. The egress process ensures that the MPLS label header is deleted from the packet and the DOCSIS header is added to the packet. Then the packet is sent out from the cable interface of the Cisco CMTS router. The egress process is also known as the label disposition process.

MPLS Pseudowire Control Plane Process

When an L2VPN-compliant CM registers with a Cisco CMTS router and conveys the L2VPN related parameters to the router, the router follows the standard Label Distribution Protocol (LDP) procedures to set up an Ethernet over MPLS pseudowire with the remote PE router. When the L2VPN-compliant CM goes offline, the Cisco CMTS router brings down the pseudowire as well. If the Cisco CMTS router has no L2VPN-compliant CM registered, then the router tears down the targeted LDP session with the remote PE router.

L2VPN Pseudowire Redundancy

The L2VPN Pseudowire Redundancy feature enables a PE router to detect a pseudowire failure and reroute the Layer 2 service to a backup pseudowire that can continue to provide the service. The pseudowire redundancy can be implemented with either Cisco CMTS or a generic router as the PE router. When the primary pseudowire recovers from the failure, the L2VPN Pseudowire Redundancy feature provides the option to bring back the Layer 2 service to the primary pseudowire.

Each primary pseudowire can have up to three backup pseudowires, with unique priorities. For example, priority one cannot be given to two different pseudowires in the backup list. When the primary pseudowire goes down, the Cisco CMTS sends the traffic to the backup pseudowire with the highest priority. For a successful service transfer, the remote state of the backup pseudowire should already be ‘up’. Only the local state of the active pseudowire will be 'up' when the modem is BPI online. Similarly, if the backup pseudowire is in use, the local state of only that backup pseudowire will be 'up'.

If the active backup pseudowire goes down, the Cisco CMTS will use the next highest backup pseudowire whose remote state is ‘up’. However, the Cisco CMTS will not switchover from the lower priority pseudowire to the higher priority pseudowire when the backup pseudowire with the highest priority comes ‘up’. This is to prevent unnecessary switchovers between the backup pseudowires.

When the primary pseudowire recovers from the failure, the L2VPN Pseudowire Redundancy feature brings back the service to the primary pseudowire, after waiting for the time period set using the backup delay command. The local state of the active backup pseudowire will be marked as ‘down’ after the primary pseudowire comes up.

MPLS Pseudowire Provisioning Methods

The MPLS Pseudowire for Cable L2VPN feature supports the following provisioning methods for pseudowires:

Note Before performing the static or dynamic provisioning of MPLS pseudowires, you must enable MPLS on a Cisco CMTS router. For details on the tasks required to enable MPLS, see the How to Enable MPLS on a Cisco CMTS Router.

Static Provisioning Method for MPLS Pseudowires

The static provisioning method requires the MPLS pseudowire to be statically provisioned on the CMTS using the command line interface (CLI). This type of provisioning does not require the CM configuration file to use BSOD L2VPN-compliant TLVs. For details on how to statically provision MPLS pseudowires, see the *Static Provisioning of MPLS Pseudowires*.

Dynamic Provisioning Method for MPLS Pseudowires

The dynamic provisioning method is a CM configuration file-based provisioning method and is the recommended provisioning method for creating MPLS pseudowires. For details on how to dynamically provision MPLS pseudowires, see the Dynamic Provisioning of MPLS Pseudowires, on page 546.

The following are the benefits of dynamic provisioning of pseudowires:

- Multiple VPNs can be specified in a CM configuration file and a pseudowire can be provisioned for each VPN.
- Multiple upstream service flows and downstream classifiers can be associated with each VPN.
- Each upstream service flow can be tagged to an MPLS experimental (EXP) level for the egress WAN traffic.
- Downstream ingress WAN traffic can be classified based on the downstream MPLS-EXP range specified in each downstream classifier.
- The Cisco CMTS router will have finer control of MPLS quality of service (QoS) over cable and WAN interfaces.

For dynamic provisioning of MPLS pseudowires, you use an L2VPN-compliant CM configuration file that is stored on the Trivial File Transfer Protocol (TFTP) server. You use a common CM configuration file editor such as CableLabs Config File Editor, or a sophisticated provisioning backend system such as Broadband Access Center for Cable (BACC) to create CM configuration files.

This provisioning method requires the usage of CableLabs defined L2VPN encodings such as type, length, value (TLV) objects in the CM configuration file. These L2VPN encodings control L2VPN forwarding of upstream and downstream Ethernet frames.

You can specify the L2VPN encodings in the following ways:

- Per CM
- Per downstream classifier
- Per service flow
- Per upstream classifier

Note The CM L2VPN encoding is mandatory.

The CM L2VPN encoding contains many TLVs, out of which the two most important TLVs are VPN Identifier and NSI Encapsulation. To configure an MPLS pseudowire, you must set the NSI Encapsulation to MPLS. The other TLVs are used to specify the pseudowire identifiers in the form of source attachment individual identifier (SAII), target attachment individual identifier (TAII), and attachment group identifier (AGI).

The L2VPN encoding parameter is encoded as a general extension information (GEI) parameter in the CM configuration file. This indicates that the parameter is encoded as a subtype of the vendor-specific information type parameter using the vendor ID (0xFFFFFF).

The table lists the important CableLabs defined TLVs that are used at the top level of the CM configuration file for the MPLS Pseudowire for Cable L2VPN feature. See the BSOD specification, *Business Services over DOCSIS (BSOD) Layer 2 Virtual Private Networks,* from CableLabs for *a complete list of CableLabs defined TLVs.*

Table 87: CableLabs Defined L2VPN TLVs

TLV Name	Type	Length	Value and Description
Downstream Unencrypted Traffic (DUT) Control	45.1	1	Bit 0 DUT Filtering DUT Filtering = 0: Disable (default) DUT Filtering = 1: Enable DUT Filtering
Downstream Unencrypted Traffic (DUT) CMIM	45.2	N	DUT CMIM (optional) CM Interface Mask (CMIM) limiting outgoing interfaces of DUT traffic. If the DUT CMIM is omitted, its default value includes the eCM and all implemented eSAFE interfaces, but not any CPE interfaces.
VPN Identifier	43.5.1	1 to N	An opaque octet string that identifies an L2VPN. N is vendor- specific, and the valid range is from 6 to 255.
NSI Encapsulation Subtype	43.5.2	n	A single NSI encapsulation format code/length/value tuple. This TLV uses any of the following values: NSI encapsulation = 0 : Other NSI encapsulation = 1 : IEEE 802.1Q (specify VLAN ID) NSI encapsulation = 2 : IEEE 802.1AD (specify Q-in-Q) NSI encapsulation = 3 : MPLS peer (specify IPv4 or IPv6 address) The value must be set to 3 to ensure MPLS pseudowire usage. The address must identify the remote PE (by its IP address assigned to the loopback interface).
Attachment Group ID	43.5.5	0 to 16	Opaque byte string that identifies the CM or SF as an attachment circuit for IETF Layer 2 VPN signaling protocols.
Source Attachment Individual ID	43.5.6	0 to 16	Opaque byte string signaled as SAII circuit for IETF Layer 2 VPN signaling protocols.
Target Attachment Individual ID	43.5.7	0 to 16	Opaque byte string that identifies the CM or SF as an attachment circuit for IETF Layer 2 VPN signaling protocols.
Ingress User Priority	43.5.8	1	Ingress IEEE 802.1 user priority value in the range of 0 to 7 encoded in the least significant three bits. Higher values indicate higher priority.
User Priority Range	43.5.9	2	The lower user priority value of the user priority range is encoded in the least significant three bits of the first byte, and the higher value of the range is encoded in the least significant three bits of the second byte.

Cisco-Specific L2VPN TLVs

Even though CableLabs defined L2VPN TLVs are sufficient for dynamic provisioning of MPLS pseudowires, CMTS operators can use Cisco-specific TLVs at the top level of the CM configuration file to enable additional functions.

This table lists the new Cisco-specific TLVs that are defined for the MPLS Pseudowire for Cable L2VPN feature.

Table 88: Cisco-Specific L2VPN TLVs

TLV Name	Type	Length	Value	Description
MPLS-PW-TYPE	43.5.43.36	1	• 4 = Type-4 Ethernet VLAN • 5 = Type-5 Ethernet port	The Cisco CMTS router interprets this subtype as MPLS pseudowire type (Type-4 or Type-5). If this TLV value is not specified, then the router accepts the default value (5) for Type-5.
MPLS-VCID	43.5.43.38	4	4 bytes unsigned number = MPLS VCID	This subtype is interpreted as MPLS VCID. This TLV is ignored, and the value of TAII is used as VCID for the pseudowire, if the following conditions are met: • The CableLabs BSOD specification-compliant TLVs, SAII and TAII, are present in the CM configuration file. • Both are of 4 bytes length. • Value of SAII is equal to TAII.
MPLS-PEERNAME	43.5.43.39	N	ASCII encoded data	The Cisco CMTS router interprets this optional subtype as MPLS peer name in ASCII encoded data.

This table lists the new Cisco-specific type, length, values (TLVs) that are defined for the L2VPN Pseudowire Redundancy feature.

Table 89: Cisco-Specific L2VPN TLVs for Pseudowire Redundancy

TLV Name	Type	Length	Value	Description
BACKUP-PW	45.5.43.40	N	Backup pseudowire related parameters	The Cisco CMTS router interprets this subtype as related parameters for the MPLS backup pseudowire. This TLV indicates the start of a new backup pseudowire.
BACKUP-PEERIP	43.5.43.40.1	4	IP address of the backup peer (IPv4)	The Cisco CMTS router interprets this optional subtype as the peer IP address of the MPLS backup pseudowire. This TLV is an IPv4 address.
BACKUP-PEERNAME	43.5.43.40.2	N	ASCII encoded data	The Cisco CMTS router interprets this optional subtype as the MPLS backup peer name in ASCII encoded data. This TLV is resolved to IPv4 address through DNS.

TLV Name	Type	Length	Value	Description
BACKUP-MPLS-VCID	43.5.43.40.3	4	4 bytes unsigned number = MPLS VCID for backup pseudowire	The Cisco CMTS router interprets this subtype as the VCID of the backup pseudowire. This TLV is ignored, and the value of TAII is used as the VCID for the pseudowire, if the following conditions are met: • The CableLabs BSOD specification-compliant TLVs, SAII, and TAII, are present in the CM configuration file. • SAII, and TAII are of 4 bytes length. • Value of SAII is equal to TAII.
BACKUP-MPLS-PRIORITY	43.5.43.40.4	1	1 byte unsigned number = priority for the backup pseudowire	The Cisco CMTS router interprets this subtype as the MPLS priority. Each primary pseudowire can have up to three backup pseudowires, with unique priorities. The priority indicates the order in which the CMTS should switch to the backup peer when the primary peer is down.
BACKUP-ENABLE-DELAY	43.5.43.41	1	1 byte unsigned number = number of seconds	The Cisco CMTS router interprets this subtype as the number of seconds the backup pseudowire should wait to take over after the primary pseudowire goes down. If the TLV value is not specified, then the router uses the default value of 0 seconds.
BACKUP-DISABLE-DELAY	43.5.43.42	1	1 byte unsigned number = number of seconds	The Cisco CMTS router interprets this subtype as the number of seconds the primary pseudowire should wait to take over after the remote state of the primary pseudowire comes up. If the TLV value is not specified, then the router uses the default value of 0 seconds.
BACKUP-DISABLE-NEVER	43.5.43.43	1	1 byte unsigned number = never disable backup pseudowire	The Cisco CMTS router interprets this subtype as a flag indicating that the backup pseudowire should not be disabled even after the primary pseudowire comes up. If this TLV is not present, the router takes the default action of reverting back to the primary pseudowire.

How to Enable MPLS on a Cisco CMTS Router

Perform the following tasks in the same order to enable MPLS on a Cisco CMTS router:

Note Before performing the static or dynamic provisioning of MPLS pseudowires, you must enable MPLS on a Cisco CMTS router.

Configuring an LDP Router ID

The **mpls ldp router-id** command allows you to assign an interface IP address as the LDP router ID.

The normal process to determine the LDP router ID is as follows:

1. The router considers all the IP addresses of all operational interfaces.
2. If these addresses include loopback interface addresses, the router selects the largest loopback address. Configuring a loopback address helps ensure a stable LDP ID for the router, because the state of loopback addresses does not change. However, configuring a loopback interface and IP address on each router is not required.

The loopback IP address is not considered as the router ID of the local LDP ID under the following circumstances:

1. If the loopback interface has been explicitly shut down.
2. If the **mpls ldp router-id** command specifies that a different interface should be used as the LDP router ID.
3. If you use a loopback interface, make sure that the IP address for the loopback interface is configured with a /32 network mask. In addition, ensure that the routing protocol in use is configured to advertise the corresponding /32 network. Otherwise, the router selects the largest interface address.

The router might select a router ID that is not usable in certain situations. For example, the router might select an IP address that the routing protocol cannot advertise to a neighboring router. The router implements the router ID the next time it is necessary to select an LDP router ID. The effect of the **mpls ldp router-id** command is delayed until it is necessary to select an LDP router ID, which is the next time the interface is shut down or the address is deconfigured.

If you use the **force** keyword with the **mpls ldp router-id** command, the router ID takes effect more quickly. However, implementing the router ID depends on the current state of the specified interface:

- If the interface is up (operational) and its IP address is not currently the LDP router ID, the LDP router ID is forcibly changed to the IP address of the interface. This forced change in the LDP router ID tears down any existing LDP sessions, releases label bindings learned via the LDP sessions, and interrupts MPLS forwarding activity associated with the bindings.
- If the interface is down, the LDP router ID is forcibly changed to the IP address of the interface when the interface transitions to up. This forced change in the LDP router ID tears down any existing LDP sessions, releases label bindings learned via the LDP sessions, and interrupts MPLS forwarding activity associated with the bindings.

Before you begin

Ensure that the specified interface is operational before assigning it as the LDP router ID.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls ip** **Example:** `Router(config)# mpls ip`	Enables the dynamic MPLS forwarding function on the specified Gigabit Ethernet interface.
Step 4	**mpls ldp router-id loopback** interface-number [**force**] **Example:** `Router(config)# mpls ldp router-id loopback 2030 force`	Specifies the IP address of the loopback interface as the LDP router ID.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.

Configuring MPLS on a Gigabit Ethernet Interface

MPLS forwarding and Label Distribution Protocol must be enabled on 1-port or 10-port GE interfaces of the Cisco CMTS router to ensure that the router establishes MPLS label-switched path (LSP) to the remote PE routers. This section explains how to enable MPLS forwarding and LDP on a Gigabit Ethernet interface.

Note Configuration steps are similar for 1-port and 10-port GE interfaces.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** slot/subslot/port **Example:** `Router(config)# interface gigabitethernet 3/0/0`	Enters interface cable configuration mode and specifies the Gigabit Ethernet interface.
Step 4	**mpls ip** **Example:** `Router(config-if)# mpls ip`	Enables the dynamic MPLS forwarding function on the specified Gigabit Ethernet interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface cable configuration mode and enters privileged EXEC mode.

Configuring an MPLS Label Distribution Protocol

The MPLS label distribution protocol (LDP) allows the construction of highly scalable and flexible IP VPNs that support multiple levels of services. This section explains how to configure an MPLS label distribution protocol on a Gigabit Ethernet interface.

MPLS LDP graceful-restart may also be configured for faster L2VPN traffic recovery after a LDP session disruption. For more information see the MPLS LDP Graceful Restart guide.

Note Ensure that the loopback interface with the IP address is present on each PE router using the **show ip interface brief** command before configuring an MPLS label distribution protocol. This loopback interface identifies the Cisco CMTS router as the peer IP address of the pseudowire.

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** slot/subslot/port **Example:** `Router(config)# interface gigabitethernet 3/0/0`	Enters interface cable configuration mode and specifies the Gigabit Ethernet interface.
Step 4	**mpls label protocol ldp** **Example:** `Router(config-if)# mpls label protocol ldp`	Enables MPLS LDP parameters on the specified Gigabit Ethernet interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface cable configuration mode and enters privileged EXEC mode.

Enabling the Cisco CMTS Support for MPLS Pseudowire for Cable L2VPN

You must enable the MPLS tunnel traffic on the network side of the interface to support configuration of MPLS pseudowires on a Cisco CMTS router.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**cable l2-vpn-service xconnect nsi mpls** **Example:** `Router(config)# cable l2-vpn-service xconnect nsi mpls`	Enables the MPLS tunnel traffic, where:
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.

How to Provision MPLS Pseudowires

You can provision MPLS pseudowires in the following ways:

Note Before performing the static or dynamic provisioning of MPLS pseudowires, you must enable MPLS on a Cisco CMTS router.

Dynamic Provisioning of MPLS Pseudowires

The dynamic provisioning method supports the following types of configurations:

- BSOD Specification-Based MPLS Pseudowire Provisioning
- Type-4 MPLS Pseudowire Provisioning Using the CM Configuration File
- Type-5 MPLS Pseudowire Provisioning Using the CM Configuration File

See the Configuration Examples for Dynamic Provisioning of MPLS Pseudowires for details about the dynamic provisioning method using the CM configuration file.

Note We recommend that you use the dynamic provisioning method instead of the static provisioning method for MPLS pseudowires.

Static Provisioning Method for MPLS Pseudowires

The static provisioning method requires the MPLS pseudowire to be statically provisioned on the CMTS using the command line interface (CLI). This type of provisioning does not require the CM configuration file to use

BSOD L2VPN-compliant TLVs. For details on how to statically provision MPLS pseudowires, see the *Static Provisioning of MPLS Pseudowires*.

How to Configure L2VPN Pseudowire Redundancy

The L2VPN Pseudowire Redundancy feature enables you to switch to backup pseudowires when the primary pseudowire fails. The feature also allows the Cisco CMTS to resume operation on the primary pseudowire after it comes back up.

Configuring the Backup Pseudowire

You can configure up to three backup pseudowires for a primary pseudowire. The priority of each backup pseudowire has to be unique.

A backup pseudowire is uniquely identified by a combination of IP address or hostname and VCID. Only the IP address or hostname and VCID can be configured for the backup peer, the remaining parameters are the same as the primary pseudowire.

Backup pseudowires can also be configured using the DOCSIS configuration files.

Perform the steps given below to configure a backup pseudowire.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable l2vpn** *mac-address* **Example:** `Router(config)# cable l2vpn 0011.0011.0011`	Specifies L2VPN MAC address and enters L2VPN *configuration mode.*
Step 4	**service instance** *id service-type* **Example:** `Router(config-l2vpn)# service instance 1 ethernet`	Specifies the service instance ID and enters Ethernet service configuration mode.

	Command or Action	Purpose
Step 5	**xconnect** *peer-ip-address vc-id* **encapsulation mpls** **Example:** `Router(config-ethsrv)# xconnect 10.2.2.2 22 encapsulation mpls`	Specifies the tunneling method to encapsulate the data in the MPLS pseudowire and enters xconnect configuration mode.
Step 6	**backup peer** *peer-ip-address vc-id* [priority value] **Example:** `Router(config-xconn)# backup peer 10.3.3.3 33 priority 2`	Specifies the backup pseudowire and its priority. The priority keyword is optional, if only one backup pseudowire is configured. When multiple backup pseudowires are configured, it is required.
Step 7	**end** **Example:** `Router(config-xconn)# end`	Exits xconnect configuration mode and enters Privileged EXEC mode.

Configuring Backup Delay

Perform the steps given below to configure the period the backup pseudowire should wait to take over after the primary pseudowire goes down. You can also specify how long the primary pseudowire should wait after it becomes active to take over from the backup pseudowire.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable l2vpn** *mac-address* **Example:** `Router(config)# cable l2vpn`	Specifies the L2VPN MAC address and enters L2VPN *configuration mode.* • *mac-address*—MAC address of a CM.

	Command or Action	Purpose
	`0011.0011.0011`	
Step 4	**service instance** *id service-type* **Example:** `Router(config-l2vpn)# service instance 1 ethernet`	Specifies the service instance ID and enters Ethernet service configuration mode. • *id*—Service instance ID. • service-type—Service type for the instance.
Step 5	**xconnect** *peer-ip-address vc-id* **encapsulation mpls** **Example:** `Router(config-ethsrv)# xconnect 10.2.2.2 22 encapsulation mpls`	Specifies the tunneling method to encapsulate the data in the MPLS pseudowire and enters xconnect configuration mode. • *peer-ip-address*—IP address of the remote PE router. The remote router ID can be any IP address, as long as it is reachable. • *vc-id*—32-bit identifier of the virtual circuit between the PE routers. • **encapsulation mpls**—Specifies MPLS as the tunneling method.

Step 6	Do one of the following:
	• **backup delay** enable-delay-period {disable-delay-period | never}
	•
	Example:
	`Router(config-xconn)# backup delay 10 10`
	Example:
	`Router(config-xconn)# backup delay 10 never`
		Specifies the period to wait before enabling or disabling the backup pseudowire.
		• *enable-delay-period*—Number of seconds the backup pseudowire should wait to take over after the primary pseudowire goes down. The valid range is from 0 to 180 seconds, with a default value of 0.
		• *disable-delay-period*—Number of seconds the primary pseudowire should wait after it becomes active to take over from the backup pseudowire. The valid range is from 0 to 180 seconds, with a default value of 0.
		• **never**—Specifies the primary pseudowire should not be reactivated after moving to the backup pseudowire.

Step 7	**end** **Example:** `Router(config-xconn)# end`	Exits xconnect configuration mode and enters privileged EXEC mode.

Performing Manual Switchover

Perform the steps given below to perform a manual switchover to the primary or backup pseudowire. The **xconnect backup force-switchover** command can also be used to forcefully switch to the backup pseudowire for planned outages of the primary remote peer.

Note A manual switchover can be made only to an available member in the redundancy group. If the pseudowire specified in the command is not available, the command will be rejected.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**xconnect backup force-switchover peer 10.10.1.1 123** **Example:** `Router# xconnect backup force-switchover peer 10.10.1.1 123`	Specifies that the router should switch to the backup or to the primary pseudowire.

Troubleshooting Tips

The following commands help you troubleshoot an improper MPLS pseudowire configuration:

- **show ip interface brief**—Helps verify that the loopback interface with the IP address is present on each PE router.
- **show mpls l2transport vc**—Helps verify information about primary and backup pseudowires that have been enabled to route Layer 2 packets on a router.
- **show xconnect all**—Helps verify information about all xconnect attachment circuits and primary and backup pseudowires.
- **show cable l2-vpn xconnect mpls-vc-map**—Helps verify that the primary and backup pseudowires are configured properly.

Configuration Examples for MPLS Pseudowire for Cable L2VPN

The following sections provide MPLS pseudowire configuration examples for the static and dynamic provisioning methods:

Configuration Example for Static Provisioning of MPLS Pseudowires

The following example shows CLI-based provisioning of an MPLS pseudowire:

```
Router> enable
Router# configure terminal
Router(config)# cable l2vpn 0000.396e.6a68 customer2
Router(config-l2vpn)# service instance 2000 ethernet
Router(config-ethsrv)# xconnect 101.1.0.2 221 encapsulation mpls pw-type 4
Router(config-ethsrv)# cable set mpls-experimental 7
```

Configuration Examples for Dynamic Provisioning of MPLS Pseudowires

The following sections provide MPLS pseudowire provisioning examples based on BSOD CableLabs specification, Type-4, and Type-5 TLVs using the CM configuration file:

BSOD Specification-Based MPLS Pseudowire Provisioning: Example

The following example shows an MPLS pseudowire configuration based on BSOD CableLabs specification:

```
03 (Net Access Control) = 1
43 (Vendor Specific Options)
   S08 (Vendor ID) = ff ff ff
   S005 (L2VPN sub-type)
 =
        T01 (VPN Id) = 02 34 56 00 02 # VPNID=0234650002
        T02 (NSI) = 04 05 01 0a 4c 01 01# [04=mpls] [05=len][01=ipv4][IP=10.76.1.1]
        T05 (AGI) = 01 01 07 d1 # AGI = 0x010107d1
        T06 (SAII) = 00 00 07 d1 # SAII = TAII = VCID = 0x7d1 = 2001
        T07 (TAII) = 00 00 07 d1
18 (Maximum Number of CPE) = 16
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference) = 1
   S06 (QoS Parameter Set Type) = 7
   S43 (Vendor Specific Options)
       T08 (Vendor ID) = ff ff ff
       T005 (L2VPN sub-type)  =
            S01 (VPNID) =   02 34 56 00 02
            S08 (UserPrio) = 01

24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference) = 2
   S06 (QoS Parameter Set Type) = 7
   S43 (Vendor Specific Options)
       T08 (Vendor ID) = ff ff ff
       T005 (L2VPN sub-type) =
            S01 (VPNID) = 02 34 56 00 02
            S08 (UserPrio) = 04
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference) = 3
   S06 (QoS Parameter Set Type) = 7
   S43 (Vendor Specific Options)
       T08 (Vendor ID) = ff ff ff
       T005 (L2VPN sub-type) =
            S01 (VPNID) =   02 34 56 00 02
            S08 (UserPrio) = 05
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference) = 4
   S06 (QoS Parameter Set Type) = 7
```

```
   S43 (Vendor Specific Options)
       T08 (Vendor ID) = ff ff ff
       T005 (L2VPN sub-type) =
            S01 (VPNID) = 02 34 56 00 02
            S08 (UserPrio) = 06
22 (Upstream Packet Classification Encoding Block)
   S01 (Classifier Reference) = 2
   S03 (Service Flow Reference) = 2
   S05 (Rule Priority) = 3
   S09 (IP Packet Encodings)
       T01 (IP Type of Srv Rng & Mask) = 00 20 ff
22 (Upstream Packet Classification Encoding Block)
   S01 (Classifier Reference) = 3
   S03 (Service Flow Reference) = 3
   S05 (Rule Priority) = 3
   S09 (IP Packet Encodings)
       T01 (IP Type of Srv Rng & Mask) = 21 40 ff
22 (Upstream Packet Classification Encoding Block)
   S01 (Classifier Reference) = 4
   S03 (Service Flow Reference) = 4
   S05 (Rule Priority) = 3
   S09 (IP Packet Encodings)
       T01 (IP Type of Srv Rng & Mask)  = 41 ff ff
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference) = 11
   S06 (QoS Parameter Set Type) = 7
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference) = 12
   S06 (QoS Parameter Set Type) = 7
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference) = 13
   S06 (QoS Parameter Set Type) = 7
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference) = 14
   S06 (QoS Parameter Set Type) = 7
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference) = 12
   S03 (Service Flow Reference) = 12
   S05 (Rule Priority) = 3
   S11 (IEEE 802.1P/Q Packet Classification Encodings)
       T01 (IEEE 802.1P UserPriority)   = 00 02
   S43 (Vendor Specific Options)
       T08 (Vendor ID) = ff ff ff
       T005 (L2VPN sub-type)

              S01 (VPNID) = 02 34 56 00 02
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference) = 13
   S03 (Service Flow Reference) = 13
   S05 (Rule Priority) = 3
   S11 (IEEE 802.1P/Q Packet Classification Encodings)
       T01 (IEEE 802.1P UserPriority) = 03 04
   S43 (Vendor Specific Options)
       T08 (Vendor ID) = ff ff ff
       T005 (L2VPN sub-type)
              S01 (VPNID) = 02 34 56 00 02
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference) = 14
   S03 (Service Flow Reference) = 14
   S05 (Rule Priority) = 3
   S11 (IEEE 802.1P/Q Packet Classification Encodings)
       T01 (IEEE 802.1P UserPriority) = 05 06
   S43 (Vendor Specific Options)
       T08 (Vendor ID) = ff ff ff
```

```
T005 (L2VPN sub-type)
       S01 (VPNID) = 02 34 56 00 02
```

Type-4 MPLS Pseudowire Provisioning Using the CM Configuration File: Example

The following example shows a CM configuration file-based provisioning of a Type-4 MPLS pseudowire:

```
03 (Net Access Control) = 1
43 (Vendor Specific Options)
   S08 (Vendor ID) = ff ff ff
   S005 (L2VPN Options) =
T001 (VPN ID) = 02 34 56 00 02 # VPN-ID = "0234560002"
T043 (Cisco Vendor Specific) = 2b 16
S008 (Vendor ID) = 00 00 0c # Vendor ID = "00 00 0C" - CISCO
S036 (MPLSPWTYPE) = 24 01 04 # MPLSPWTYPE= Type4 - Ethernet-vlan Type
S039 (MPLSPEERNAME) = 27 06 63 37 36 30 30 32 MPLSPEERNAME= "c76002" in ascii
S038 (MPLSVCID) = 26 04 00 00 07 d1 = 2001 VCID
43 (Vendor Specific Options)
   S08 (Vendor ID) = ff ff ff
   S005 (L2VPN Options) =
T001 (VPN ID) = 02 34 56 00 03 # VPN-ID = "0234560003"
T043 (Cisco Vendor Specific) =  2b 16
S008 (Vendor ID) = 00 00 0c Vendor ID = "00 00 0C" - CISCO
S036 (MPLSPWTYPE) = 24 01 04 MPLSPWTYPE= Type4 - Ethernet-vlan Type
S039 (MPLSPEERNAME) = 27 06 63 37 36 30 30 32 # MPLSPEERNAME= "c76002" in ascii
S038 (MPLSVCID) = 26 04 00 00 0b b9 # = 3001 VCID
43 (Vendor Specific Options)
   S08 (Vendor ID) = ff ff ff
   S005 (L2VPN Options) =
T001 (VPN ID) = 02 34 56 00 04 # VPN-ID = "0234560004"
T043 (Cisco Vendor Specific) = 2b 16
S008 (Vendor ID) = 00 00 0c # Vendor ID = "00 00 0C" - CISCO
S036 (MPLSPWTYPE) = 24 01 04 # MPLSPWTYPE= Type4 - Ethernet-vlan Type
S039 (MPLSPEERNAME) = 27 06 63 37 36 30 30 32 # MPLSPEERNAME= "c76002" in ascii
S038 (MPLSVCID) = 26 04 00 00 0f a1 # = 4001 VCID
18 (Maximum Number of CPE) = 16
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference) = 1
   S06 (QoS Parameter Set Type) = 7
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference) = 2
   S06 (QoS Parameter Set Type) = 7
   S43 (Vendor Specific Options)
       T08 (Vendor ID) = ff ff ff
       T001 (VPN ID) = 02 34 56 00 02
       T043 (Cisco Vendor Specific) = 2b 0A
S008 (Vendor ID) = 00 00 0c # Vendor ID = "00 00 0C" - CISCO

S034 (MPLS-EXP-SET) = 22 05 # MPLSEXP-INGRESS= 5
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference) = 3
   S06 (QoS Parameter Set Type) = 7
   S43 (Vendor Specific Options)
       T08 (Vendor ID) = ff ff ff
       T001 (VPN ID) = 02 34 56 00 03
       T043 (Cisco Vendor Specific) =  2b 0A
S008 (Vendor ID) = 00 00 0c
# Vendor ID = "00 00 0C" - CISCO

S034 (MPLS-EXP-SET) = 22 06

# MPLSEXP-INGRESS= 6
```

```
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference) = 4
   S06 (QoS Parameter Set Type) = 7
   S43 (Vendor Specific Options)
       T08 (Vendor ID) = ff ff ff
       T001 (VPN ID) = 02 34 56 00 04
       T043 (Cisco Vendor Specific) = 2b 0A
S008 (Vendor ID) = 00 00 0c
# Vendor ID = "00 00 0C" - CISCO

S034 (MPLS-EXP-SET) = 22 04

# MPLSEXP-INGRESS= 4
22 (Upstream Packet Classification Encoding Block)
   S01 (Classifier Reference) = 2
   S03 (Service Flow Reference) = 2
   S11 (IEEE 802.1P/Q Packet Classification Encodings)
       T02 (IEEE 802.1Q VLAN ID) = 7d 00
   S05 (Rule Priority) = 2
22 (Upstream Packet Classification Encoding Block)
   S01 (Classifier Reference) = 3
   S03 (Service Flow Reference) = 3
   S11 (IEEE 802.1P/Q Packet Classification Encodings)
       T02 (IEEE 802.1Q VLAN ID) = bb 80
   S05 (Rule Priority) = 3
22 (Upstream Packet Classification Encoding Block)
   S01 (Classifier Reference) = 4
   S03 (Service Flow Reference) = 4
   S11 (IEEE 802.1P/Q Packet Classification Encodings)
       T02 (IEEE 802.1Q VLAN ID) = fa 00
   S05 (Rule Priority) = 4
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference) = 11
   S06 (QoS Parameter Set Type) = 7
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference) = 12
   S06 (QoS Parameter Set Type) = 7
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference) = 13
   S06 (QoS Parameter Set Type) = 7
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference) = 14
   S06 (QoS Parameter Set Type) = 7
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference) = 12
   S03 (Service Flow Reference) = 12
   S11 (IEEE 802.1P/Q Packet Classification Encodings)
       T02 (IEEE 802.1Q VLAN ID) = 7d 00
   S43 (Vendor Specific Options)
       T08 (Vendor ID) = ff ff ff
       T001 (VPN ID) = 02 34 56 00 02
       T043 (Cisco Vendor Specific) = 2b 0B
S008 (Vendor ID) = 00 00 0c # Vendor ID = "00 00 0C" - CISCO
S035 (MPLS-EXP_RANGE) = 23 02  03 # MPLSEXP-EGRESS_RANGE= 2 - 3
   S05 (Rule Priority) = 2
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference) = 13
   S03 (Service Flow Reference) = 13
   S11 (IEEE 802.1P/Q Packet Classification Encodings)
       T02 (IEEE 802.1Q VLAN ID) = bb 80
   S43 (Vendor Specific Options)
       T08 (Vendor ID) = ff ff ff
       T001 (VPN ID) =  02 34 56 00 03
       T043 (Cisco Vendor Specific) = 2b 0B
```

```
S008 (Vendor ID) = 00 00 0c # Vendor ID = "00 00 0C" - CISCO

S035 (MPLS-EXP-RANGE) = 23 04  05 # MPLSEXP-EGRESS_RANGE= 4 - 5
   S05 (Rule Priority = 3
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference) = 14
   S03 (Service Flow Reference) = 14
   S11 (IEEE 802.1P/Q Packet Classification Encodings)
       T02 (IEEE 802.1Q VLAN ID) = fa 00
   S43 (Vendor Specific Options)
       T08 (Vendor ID) = ff ff ff
       T001 (VPN ID) = 02 34 56 00 04
       T043 (Cisco Vendor Specific) = 2b 0B
S008 (Vendor ID) = 00 00 0c # Vendor ID = "00 00 0C" - CISCO

S035 (MPLS-EXP-RANGE) = 23 00  01 # MPLSEXP-EGRESS_RANGE= 0 - 1
   S05 (Rule Priority) = 4
```

Type-5 MPLS Pseudowire Provisioning Using the CM Configuration File: Example

The following example shows a CM configuration file-based provisioning of a Type-5 MPLS pseudowire:

```
03 (Net Access Control) = 1
43 (Vendor Specific Options)
   S08 (Vendor ID) = ff ff ff
   S005 (L2VPN Options) =
T001 (VPN ID) = 02 34 56 00 02 # VPN-ID = "0234560002"
T043 (Cisco Vendor Specific) = 2b 16
S008 (Vendor ID) = 00 00 0c # Vendor ID = "00 00 0C" - CISCO
S036 (MPLSPWTYPE) = 24 01 05 # MPLSPWTYPE= Type5 - Ethernet-Port Type
S039 (MPLSPEERNAME) = 27 06 63 37 36 30 30 32 # MPLSPEERNAME= "c76002" in ascii
S038 (MPLSVCID) = 26 04 00 00 07 d1 # = 2001 VCID
45 (L2VPN CMIM) = 02 04 ff ff ff ff 01 01 01
18 (Maximum Number of CPE) = 16
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference) = 1
   S06 (QoS Parameter Set Type) = 7
   S43 (Vendor Specific Options)
       T08 (Vendor ID) = ff ff ff
       T001 (VPN ID) = 02 34 56 00 02 # VPN-ID = "0234560002"
       T043 (Cisco Vendor Specific) = 2b 0A
S008 (Vendor ID) = 00 00 0c # Vendor ID = "00 00 0C" - CISCO
S034 (MPLS-EXP-SET) = 22 04 # MPLS-EXP-SET at INGRESS= 4
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference) = 12
   S06 (QoS Parameter Set Type) = 7
```

Configuration Examples for L2VPN Pseudowire Redundancy

The following sections provide L2VPN pseudowire redundancy configuration examples using the CM configuration file:

Example: Configuring Backup Pseudowire Peer and VC ID

The following example shows how to provision a file-based backup peer router based on the CM configuration:

PE Router 1

```
cable l2vpn 0025.2e2d.7252
 service instance 1 ethernet
  encapsulation default
  xconnect 10.76.2.1 400 encapsulation mpls
   backup peer 10.76.2.1 600 priority 4
```

PE Router2

```
cable l2vpn 0011.0011.0011
 service instance 1 ethernet
  encapsulation default
  xconnect 10.2.2.2 22 encapsulation mpls
   backup peer 10.3.3.3 33 priority 2
   backup delay 10 10
```

Example: Configuring Backup Delay

The following example shows how to configure a backup delay to determine how much time should elapse before a secondary line status change after a primary line status has been changed.

```
cable l2vpn 0011.0011.0011
 service instance 1 ethernet
  encapsulation default
  xconnect 10.2.2.2 22 encapsulation mpls
   backup delay 10 10
```

Example: L2VPN Backup MPLS Pseudowire Provisioning Using the CM Configuration File

The following example shows how to provision an L2VPN Backup MPLS pseudowire based on the CM configuration file:

```
03 (Net Access Control)         = 1
18 (Maximum Number of CPE)      = 3
43 (Vendor Specific Options)
   S08 (Vendor ID)      = ff ff ff
   S005 (Unknown sub-type)      = 01 04 32 30 32 30 02 07 04 05 01 0a 4c 02 01 2b 15 26 04
 00 00 00 14 28 10 01 05 01 0a 4c 02 01 03 04 00 00 07 08 04 01 05 28 0d 01 05 01 0a 4c 02
 03 03 04 00 00 00 15 28 10 01 05 01 0a 4c 02 01 03 04 00 00 b1 8e 04 01 01 29 01 03 2a 01
 01
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference)         = 4
   S06 (QoS Parameter Set Type)         = 7
   S08 (Max Sustained Traffic Rate)     = 2000000
   S09 (Max Traffic Burst)              = 3200
   S15 (Service Flow Sched Type)        = 2
   S43 (Vendor Specific Options)
       T08 (Vendor ID)      = ff ff ff
       T005 (Unknown sub-type)      = 01 04 32 30 32 30
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference)         = 2
   S06 (QoS Parameter Set Type)         = 7
   S08 (Max Sustained Traffic Rate)     = 3000000
   S09 (Max Traffic Burst)              = 250000
29 (Privacy Enable)             = 1
```

Verifying the MPLS Pseudowire Configuration

Use the following **show** commands to verify the MPLS pseudowire configuration:

- **show mpls ldp discovery**
- **show cable l2-vpn xconnect**
- **show xconnect**
- **show mpls l2transport vc**

To verify the mapping between the MPLS pseudowire and virtual circuits for all cable modems, use the **show cable l2-vpn xconnect** command as shown in the following example:

```
Router# show cable l2-vpn xconnect mpls-vc-map
MAC Address     Peer IP Address VCID Type Prio CktID     Cable Intf  SID Customer Name/VPNID
0023.bee1.eb48  123.1.1.1       30   Prim*     Bu254:4101 Cable3/0/0  3
38c8.5cac.4a62  123.1.1.1       20   Prim*     Bu254:4100 Cable3/0/0  4   customer1
602a.d083.2e1c  123.1.1.1       60   Prim*     Bu254:4102 Cable3/0/0  5
```

To verify the mapping between the MPLS pseudowire and virtual circuits for all cable modems when pseudowire redundancy is not configured, use the **show cable l2-vpn xconnect mpls-vc-map** command as shown in the following example:

```
Router# show cable l2-vpn xconnect mpls-vc-map
MAC Address    Peer IP Address VCID Type  Prio CktID     Cable Intf SID Customer Name/VPNID
0025.2e2d.7252 10.76.2.1       400  Prim*      Bu254:400  Cable8/0/3 1
0014.f8c1.fd46 10.2.3.4        1000 Prim*      Bu254:1000 Cable8/0/0 1   2020
0014.f8c1.fd46 10.76.2.1       1800 Prim*      Bu254:1800 Cable8/0/0 1   2021
```

To verify the mapping between the MPLS pseudowire and virtual circuits for all cable modems when pseudowire redundancy is configured, use the **show cable l2-vpn xconnect mpls-vc-map** command as shown in the following example:

```
Router# show cable l2-vpn xconnect mpls-vc-map
MAC Address    Peer IP Address VCID Type  Prio CktID     Cable Intf SID Customer Name/VPNID
602a.d083.2e1c 123.1.1.1       60   Prim*      Bu254:4102 Cable3/0/0 5
38c8.5cac.4a62 123.1.1.1       20   Prim*      Bu254:4103 Cable3/0/0 4   000232303230
               156.1.3.1       30   Bkup  3    Bu254:4103
               123.1.1.1       50   Bkup  8    Bu254:4103
38c8.5cac.4a62 156.1.3.1       56   Prim*      Bu254:4104 Cable3/0/0 4   000232303231
               123.1.1.1       40   Bkup  1    Bu254:4104
```

To obtain the state of all virtual circuits associated with an MPLS pseudowire when pseudowire redundancy is not configured, use the **show cable l2-vpn xconnect mpls-vc-map state** command as shown in the following example:

```
Router# show cable l2-vpn xconnect mpls-vc-map state
MAC Address      Peer IP Address VCID Type Prio State Customer Name/VPNID State
602a.d083.2e1c   123.1.1.1       60   Prim*     UP                        UP
38c8.5cac.4a62   123.1.1.1       20   Prim*     UP    000232303230        UP
38c8.5cac.4a62   156.1.3.1       56   Prim*     UP    000232303231        UP
```

To obtain the state of all virtual circuits associated with an MPLS pseudowire when pseudowire redundancy is configured, use the **show cable l2-vpn xconnect mpls-vc-map state** command as shown in the following example:

```
Router# show cable l2-vpn xconnect mpls-vc-map state
MAC Address      Peer IP Address VCID Type   Prio State Customer Name/VPNID State
602a.d083.2e1c   123.1.1.1       60   Prim*       UP                        UP
38c8.5cac.4a62   123.1.1.1       20   Prim*       UP    000232303230        UP
                 156.1.3.1       30   Bkup   3    UP    000232303230        STDBY
                 123.1.1.1       50   Bkup   8    DOWN  000232303230        STDBY
38c8.5cac.4a62   156.1.3.1       56   Prim*       UP    000232303231        UP
                 123.1.1.1       40   Bkup   1    UP    000232303230        STDBY

When the local state of the modem is DOWN, the L2VPN is not configured on the WAN interface
 and the remote state of the L2VPN will be shown as OFF.

Router#show cable l2-vpn xconnect mpls-vc-map state
MAC Address     Peer IP Address VCID Type   Prio State Customer Name/VPNID State
602a.d083.2e1c  123.1.1.1       60   Prim*       OFF               DOWN
38c8.5cac.4a62  123.1.1.1       20   Prim*       UP    000232303230 UP
38c8.5cac.4a62  156.1.3.1       56   Prim*       UP    000232303231 UP
```

To verify information about the MPLS pseudowire mapping for a particular MAC address of a CM when pseudowire redundancy is configured, use the **show cable l2-vpn xconnect mpls-vc-map** command as shown in the following example:

```
Router# show cable l2-vpn xconnect mpls-vc-map 0025.2e2d.7252
MAC Address    Peer IP Address VCID Type  Prio CktID     Cable Intf SID Customer Name/VPNID
0025.2e2d.7252 10.76.2.1       400  Prim*      Bu254:400 Cable8/0/3 1
               10.76.2.1       600  Bkup  4    Bu254:600
```

To verify the detailed information about the MPLS pseudowire mapping for a CM when pseudowire redundancy is configured, use the **show mpls l2-vpn xconnect mpls-vc-map verbose** command as shown in the following examples.

The following example shows the information for a modem for which pseudowires were configured using backup peer command:

```
Router# show cable l2-vpn xconnect mpls-vc-map 0025.2e2d.7252 verbose
MAC Address                    : 0025.2e2d.7252
Customer Name                  :
Prim Sid                       : 1
Cable Interface                : Cable8/0/3
MPLS-EXP                       : 0
PW TYPE                        : Ethernet
Backup enable delay            : 0 seconds
Backup disable delay           : 0 seconds
Primary peer
Peer IP Address (Active)       : 10.76.2.1
XConnect VCID                  : 400
Circuit  ID                    : Bu254:400
Local State                    : UP
Remote State                   : UP
Backup peers
Peer IP Address                : 10.76.2.1
XConnect VCID                  : 600
Circuit  ID                    : Bu254:600
Local State                    : STDBY
Remote State                   : UP
Priority                       : 4
```

```
Total US pkts                       : 0
Total US bytes                      : 0
Total US pkts   discards            : 0
Total US bytes  discards            : 0
Total DS pkts                       : 0
Total DS bytes                      : 0
Total DS pkts   discards            : 0
Total DS bytes  discards            : 0
```

The following example shows the information for a modem for which pseudowires were created using the modem configuration file:

```
Router# show cable l2-vpn xconnect mpls-vc-map 0014.f8c1.fd46 verbose
MAC Address                         : 0014.f8c1.fd46
Prim Sid                            : 3
Cable Interface                     : Cable8/0/0
L2VPNs provisioned                  : 1
DUT Control/CMIM                    : Disable/0x8000FFFF
VPN ID                              : 2020
L2VPN SAID                          : 12289
Upstream SFID Summary               : 15
Downstream CFRID[SFID] Summary      : Primary SF
CMIM                                : 0x60
PW TYPE                             : Ethernet
MPLS-EXP                            : 0
Backup enable delay                 : 3 seconds
Backup disable delay                : 1 seconds
Primary peer
Peer IP Address (Active)            : 10.2.3.4
XConnect VCID                       : 1000
Circuit  ID                         : Bu254:1000
Local State                         : UP
Remote State                        : UP

Backup peers
Peer IP Address                     : 10.2.3.4
XConnect VCID                       : 21
Circuit  ID                         : Bu254:21
Local State                         : STDBY
Remote State                        : DOWN
Priority                            : 2
Peer IP Address                     : 10.76.2.1
XConnect VCID                       : 1800
Circuit  ID                         : Bu254:1800
Local State                         : STDBY
Remote State                        : DOWN
Priority                            : 5
Peer IP Address                     : 10.76.2.1
XConnect VCID                       : 45454
Circuit  ID                         : Bu254:45454
Local State                         : STDBY
Remote State                        : DOWN
```

To verify information about all attachment circuits and pseudowires for online modems, use the **show xconnect** command as shown in the following example:

```
Router# show xconnect all
Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  RV=Recovering      NH=No Hardware
XC ST  Segment 1                        S1 Segment 2                        S2
```

```
------+---------------------------------+--+---------------------------------+--
UP     ac   Bu254:2001(DOCSIS)           UP mpls 10.76.1.1:2001               UP
UP     ac   Bu254:2002(DOCSIS)           UP mpls 10.76.1.1:2002               UP
UP     ac   Bu254:2004(DOCSIS)           UP mpls 10.76.1.1:2004               UP
DN     ac   Bu254:22(DOCSIS)             UP mpls 101.1.0.2:22                 DN
```

To verify information about MPLS virtual circuits and static pseudowires that have been enabled to route Layer 2 packets on a Cisco CMTS router, use the **show mpls l2transport vc** command as shown in the following example:

```
Router# show mpls l2transport vc
Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
Bu254          DOCSIS 2002                10.76.1.1       2002       UP
Bu254          DOCSIS 2003                10.76.1.1       2003       UP
Bu254          DOCSIS 2004                10.76.1.1       2004       DOWN
Bu254          DOCSIS 2017                10.76.1.1       2017       UP
Bu254          DOCSIS 2018                10.76.1.1       2018       UP
Bu254          DOCSIS 2019                10.76.1.1       2019       UP
```

Additional References

Standards

Standard	Title
CM-SP-L2VPN-I08-080522	*Business Services over DOCSIS (BSOD) Layer 2 Virtual Private Networks*
L2VPN-N-10.0918-2	*L2VPN MPLS Update*

MIBs

MIB	MIBs Link
• DOCS-L2VPN-MIB • CISCO-IETF-PW-MIB • CISCO-CABLE-L2VPN-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://tools.cisco.com/ITDIT/MIBS/servlet/index

RFCs

RFC	Title
RFC 3985	*Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture*
RFC 4385	*Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN*
RFC 4446	*IANA Allocations for Pseudowire Edge-to-Edge Emulation (PWE3)*
RFC 4447	*Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*
RFC 4448	*Encapsulation Methods for Transport of Ethernet over MPLS Networks*

RFC	Title
RFC 5085	*Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

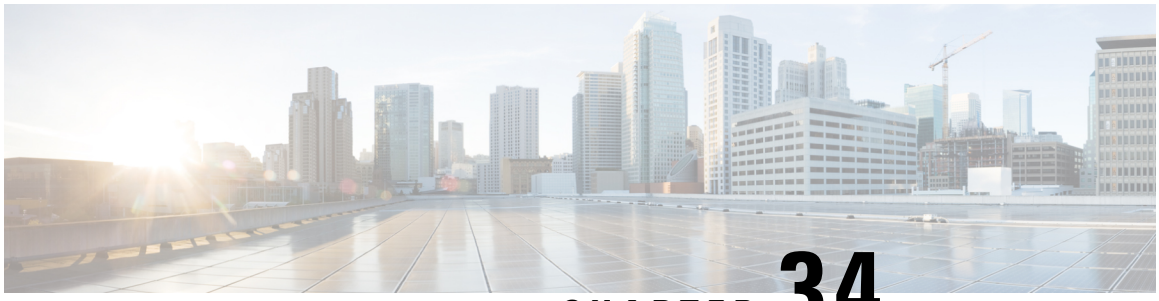
Feature Information for MPLS Pseudowire for Cable L2VPN

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 90: Feature Information for MPLS Pseudowire for Cable L2VPN

Feature Name	Releases	Feature Information
MPLS Pseudowire for Cable L2VPN	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 34

MPLS VPN Cable Enhancements

This feature module describes the Multiprotocol Label Switching Virtual Private Network (MPLS VPN) and cable interface bundling features. It explains how to create a VPN using MPLS protocol, cable interfaces, bundle interfaces and sub bundle interfaces. VPNs can be created in many ways using different protocols.

- Finding Feature Information, on page 563
- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 563
- Feature Overview, on page 564
- Prerequisites, on page 568
- Configuration Tasks, on page 569
- Configuration Examples, on page 573
- Additional References, on page 577
- Feature Information for MPLS VPN Cable Enhancements, on page 578

Finding Feature Information

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 91: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Feature Overview

Using MPLS VPN technology, service providers can create scalable and efficient private networks using a shared hybrid fiber coaxial (HFC) network and Internet protocol (IP) infrastructure.

The cable MPLS VPN network consists of:

- The Multiple Service Operator (MSO) or cable company that owns the physical infrastructure and builds VPNs for the Internet Service Providers (ISPs) to move traffic over the cable and IP backbone.
- ISPs that use the HFC network and IP infrastructure to supply Internet service to cable customers.

Each ISP moves traffic to and from a subscriber's PC, through the MSO's physical network infrastructure, to the ISP's network. MPLS VPNs, created in Layer 3, provide privacy and security by constraining the distribution of a VPN’s routes only to the routers that belong to its network. Thus, each ISP's VPN is insulated from other ISPs that use the same MSO infrastructure.

An MPLS VPN assigns a unique VPN Routing/Forwarding (VRF) instance to each VPN. A VRF instance consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine the contents of the forwarding table.

Each PE router maintains one or more VRF tables. It looks up a packet’s IP destination address in the appropriate VRF table, only if the packet arrived directly through an interface associated with that table.

MPLS VPNs use a combination of BGP and IP address resolution to ensure security. See *Configuring Multiprotocol Label Switching.*

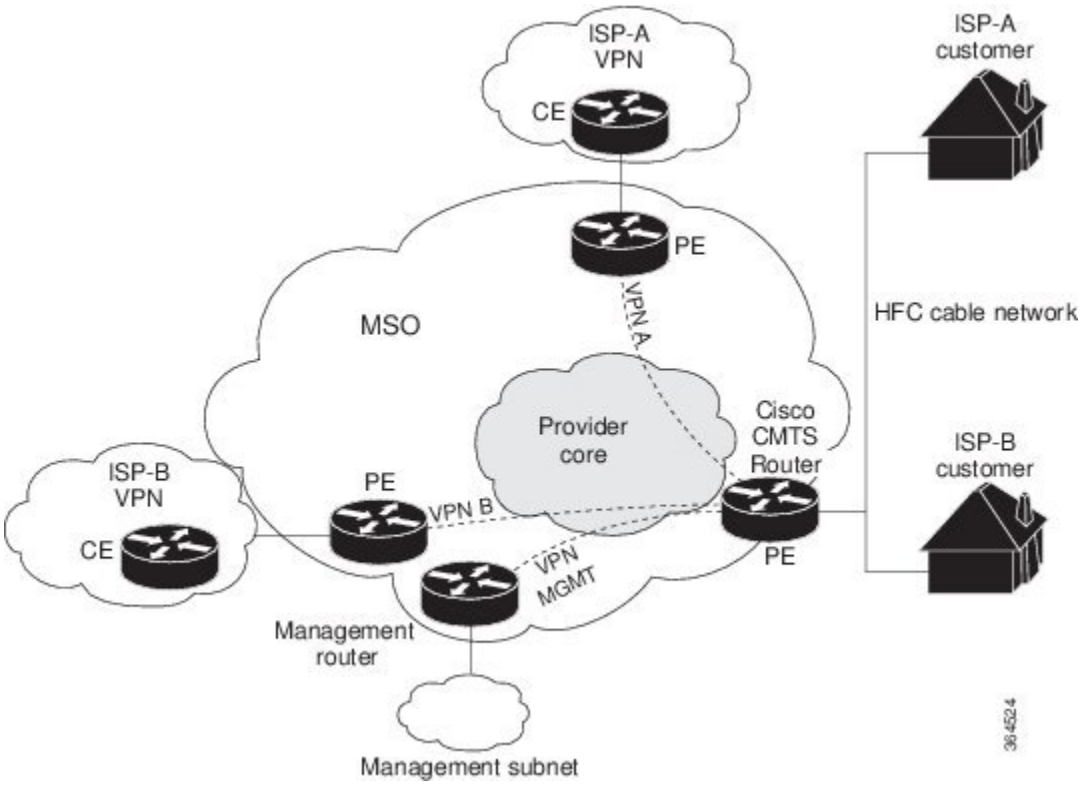
The table shows a cable MPLS VPN network. The routers in the network are:

- Provider (P) router—Routers in the core of the provider network. P routers run MPLS switching, and do not attach VPN labels (MPLS label in each route assigned by the PE router) to routed packets. VPN labels are used to direct data packets to the correct egress router.
- Provider Edge (PE) router— Router that adds the VPN label to incoming packets based on the interface or subinterface on which they are received. A PE router attaches directly to a CE router. In the MPLS-VPN approach, each Cisco CMTS router acts as a PE router.
- Customer (C) router—Router in the ISP or enterprise network.
- Customer Edge (CE) router—Edge router on the ISP's network that connects to the PE router on the MSO's network. A CE router must interface with a PE router.

The MPLS network has a unique VPN that exclusively manages the MSOs devices called the management VPN. It contains servers and devices that other VPNs can access. The management VPN connects the Cisco CMTS router to a PE router, which connects to management servers such as Cisco Network Registrar (CNR) and Time of Day (ToD) servers. A PE router connects to management servers and is a part of the management VPN. Regardless of the ISP they belong to, the management servers serve the Dynamic Host Configuration Protocol (DHCP), DNS (Domain Name System), and TOD requests coming from PCs or cable modems.

Note When configuring MPLS VPNs, you must configure the first subinterface created as a part of the management VPN.

Figure 24: MPLS VPN Network



Cable VPN configuration involves an:

- MSO domain that requires a direct peering link to each enterprise network (ISP), provisioning servers for residential and commercial subscribers, and dynamic DNS for commercial users. The MSO manages cable interface IP addressing, Data-over-Cable Service Interface Specifications (DOCSIS) provisioning, CM hostnames, routing modifications, privilege levels, and usernames and passwords.
- ISP or enterprise domain that includes the DHCP server for subscriber or telecommuter host devices, enterprise gateway within the MSO address space, and static routes back to the telecommuter subnets.

Note Cisco recommends that the MSO assign all addresses to the end user devices and gateway interfaces. The MSO can also use split management to let the ISP configure tunnels and security.

In an MPLS VPN configuration, the MSO must configure the following:

- CMTS
- P routers
- PE routers
- CE routers
- One VPN per ISP DOCSIS servers for all cable modem customers. The MSO must attach DOCSIS servers to the management VPN, and make them visible.

The MSO must configure the Cisco CMTS routers that serve the ISP, and remote PE routers connecting to the ISP, as PE routers in the VPN.

The MSO must determine the primary IP address range for all cable modems.

The ISP must determine the secondary IP address range for subscriber PCs.

To reduce security breaches and differentiate DHCP requests from cable modems in VPNs or under specific ISP management, MSOs can use the **cable helper-address** command in Cisco IOS-XE software. The MSO can specify the host IP address to be accessible only in the ISP's VPN. This lets the ISP use its DHCP server to allocate IP addresses. Cable modem IP address must be accessible from the management VPN.

The MPLS VPN approach of creating VPNs for individual ISPs or customers requires subinterfaces to be configured on the virtual bundle interface. Each ISP requires one subinterface. The subinterfaces are tied to the VPN Routing/Forwarding (VRF) tables for their respective ISPs. The first subinterface must be created on the cable interface bound to the management VPN.

To route a reply from the CNR back to the cable modem, the PE router that connects to the CNR must import the routes of the ISP VPN into the management VPN. Similarly, to forward management requests (such as DHCP renewal to CNR) to the cable modems, the ISP VPN must export and import the appropriate management VPN routes.

You can group all of the cable interfaces on a Cisco CMTS router into a single bundle so that only one subnet is required for each router. When you group cable interfaces, no separate IP subnet or each individual cable interface is required. This grouping avoids the performance, memory, and security problems in using a bridging solution to manage subnets, especially for a large number of subscribers.

Subinterfaces allow traffic to be differentiated on a single physical interface, and assigned to multiple VPNs. You can configure multiple subinterfaces, and associate an MPLS VPN with each subinterface. You can split a single physical interface (the cable plant) into multiple subinterfaces, where each subinterface is associated with a specific VPN. Each ISP requires access on a physical interface and is given its own subinterface. Create a management subinterface to support cable modem initialization from an ISP.

Using each subinterface associated with a specific VPN (and therefore, ISP) subscribers connect to a logical subinterface, which reflects the ISP that provides their subscribed services. When properly configured, subscriber traffic enters the appropriate subinterface and VPN.

Benefits

- MPLS VPNs give cable MSOs and ISPs a manageable way of supporting multiple access to a cable plant. Service providers can create scalable and efficient VPNs across the core of their networks. MPLS VPNs provide systems support scalability in cable transport infrastructure and management.
- Each ISP can support Internet access services from a subscriber's PC through an MSO's physical cable plant to their networks.
- MPLS VPNs allow MSOs to deliver value-added services through an ISP, and thus, deliver connectivity to a wider set of potential customers. MSOs can partner with ISPs to deliver multiple services from multiple ISPs and add value within the MSO's own network using VPN technology.
- Subscribers can select combinations of services from various service providers.
- The MPLS VPN cable features set build on CMTS DOCSIS 1.0 and DOCSIS 1.0 extensions to ensure services are reliably and optimally delivered over the cable plant. MPLS VPN provides systems support domain selection, authentication per subscriber, selection of QoS, policy-based routing, and ability to reach behind the cable modem to subscriber end devices for QoS and billing while preventing session spoofing.
- MPLS VPN technology ensures both secure access across the shared cable infrastructure and service integrity.
- Cable interface bundling eliminates the need for an IP subnet on each cable interface. Instead, an IP subnet is only required for each cable interface bundle. All cable interfaces in a Cisco CMTS router can be added to a single bundle.

Restrictions

- Each subinterface on the CMTS requires an address range from the ISP and from the MSO. These two ranges must not overlap and must be extensible to support an increased number of subscribers for scalability.

Note This document does not address allocation and management of MSO and ISP IP addresses. See *Configuring Multiprotocol Label Switching* for this information.

- The **cable source-verify dhcp** command enables Dynamic Host Control Protocol (DHCP) Lease query protocol from the CMTS to DHCP server to verify IP addresses of upstream traffic, and prevent MSO customers from using unauthorized, spoofed, or stolen IP addresses.
- When using only MPLS VPNs, create subinterfaces on the virtual bundle, assign it an IP address, and provide VRF configuration for each ISP. When you create subinterfaces and configure only MPLS VPNs, the cable interface bundling feature is independent of the MPLS VPN.
- When using cable interface bundling:
  - Define a virtual bundle interface and associate any cable physical interface to the virtual bundle.

- Specify all generic IP networking information (such as IP address, routing protocols, and switching modes) on the virtual bundle interface. Do not specify generic IP networking information on bundle subsidiary interfaces.
- An interface that has a subinterface(s) defined over it is not allowed to be a part of the bundle.
- Specify generic (not downstream or upstream related) cable interface configurations, such as source-verify or ARP handling, on the virtual bundle interface. Do not specify generic configuration on bundle subsidiary interfaces.

- Interface bundles can only be configured using the command line interface (including the CLI-based HTML configuration).

Prerequisites

Before configuring IP-based VPNs, complete the following tasks:

- Ensure your network supports reliable broadband data transmission. Your plant must be swept, balanced, and certified based on National Television Standards Committee (NTSC) or appropriate international cable plant recommendations. Ensure your plant meets all DOCSIS or European Data-over-Cable Service Interface Specifications (EuroDOCSIS) downstream and upstream RF requirements.
- Ensure your Cisco router is installed following instructions in the Hardware Installation Guide and the Regulatory Compliance and Safety Information guide.
- Ensure your Cisco router is configured for basic operations.
- The chassis must contain at least one port adapter to provide backbone connectivity and one Cisco cable modem card to serve as the RF cable TV interface.

Other Important Information

- Ensure all other required headend or distribution hub routing and network interface equipment is installed, configured, and operational based on the services to support. This includes all routers, servers (DHCP, TFTP, and ToD), network management systems, other configuration or billing systems and backbone, and other equipment to support VPN.
- Ensure DHCP and DOCSIS configuration files have been created and pushed to appropriate servers such that each cable modem, when initialized, can transmit a DHCP request, receive an IP address, obtain TFTP and ToD server addresses, and download a DOCSIS configuration file. Configure each subinterface to connect to the ISP's VPN.
- Ensure DOCSIS servers are visible on the management VPN.
- Be familiar with your channel plan to assign appropriate frequencies. Outline your strategies for setting up bundling or VPN solution sets if applicable to your headend or distribution hub. Obtain passwords, IP addresses, subnet masks, and device names as appropriate.
- Create subinterfaces off of a virtual bundle interface. Configure each subinterface to connect to the ISP network.

The MPLS VPN configuration steps assume the following:

- IP addressing has already been determined and there are assigned ranges in the MSO and ISP network for specific subinterfaces.
- The MSO is using CNR and has configured it (using the **cable helper-address** command) to serve appropriate IP addresses to cable modems based on the cable modem MAC address. The CMTS forwards

DHCP requests to the CNR based on the **cable helper-address** settings. The CNR server determines the IP address to assign the cable modem using the client-classes feature, which let the CNR assign specific parameters to devices based on MAC addresses.

- ISP CE routers are configured (using the **cable helper-address** command) to appropriately route relevant IP address ranges into the VPN.
- P and PE routers are already running Cisco Express Forwarding (CEF).
- MPLS is configured on the outbound VPN using the **tag switching ip** command in interface configuration mode.

Configuration Tasks

To configure MPLS VPNs, perform the following tasks:

Creating VRFs for each VPN

To create VRFs for each VPN, perform the following steps beginning in the router configuration mode.

Note Since only the CMTS has logical subinterfaces, assignments of VRFs on the other PE devices will be to specific physical interfaces.

Procedure

	Command or Action	Purpose
Step 1	Router(config)# **vrf definition** *mgmt-vpn*	Enters VRF configuration mode (config-vrf)# and maps a VRF table to the VPN (specified by *mgmt-vpn*). The management VPN is the first VPN configured.
Step 2	Router(config-vrf)# **rd** *mgmt-rd*	Creates a routing and forwarding table by assigning a route distinguisher to the management VPN.
Step 3	Router(config-vrf)# **route-target** {**export**| **import**| **both**} *mgmt-rd*	Exports and/or imports all routes for the management VPNs route distinguisher. This determines which routes will be shared within VRFs.
Step 4	Router(config-vrf)# **route-target import** isp1-vpn-rd	Imports all routes for the VPNs (*isp1-vpn)* route distinguisher.
Step 5	Router(config-vrf)# **route-target import** isp2-vpn-rd	Imports all routes for the VPNs (*isp2-vpn)* route distinguisher.
Step 6	Router(config-vrf)# **vrf definition** isp1-vpn	Creates a routing and forwarding table by assigning a route distinguisher to *isp1-vpn* .

	Command or Action	Purpose
Step 7	Router(config-vrf)# **rd** *mgmt-rd*	Creates a routing and forwarding table by assigning a route distinguisher (mgmt-rd) to the management VPN (mgmt-vpn).
Step 8	Router(config-vrf)# **route-target export** isp1-vpn-rd	Exports all routes for the VPNs (*isp1-vpn)* route distinguisher.
Step 9	Router(config-vrf)# **route-target import** isp1-vpn-rd	Imports all routes for the VPNs (*isp1-vpn)* route distinguisher.
Step 10	Router(config-vrf)# **route-target import** mgmt-vpn-rd	Exports all routes for the VPNs (*mgmt-vpn)* route distinguisher.
Step 11	Router(config-vrf)# **vrf definition** isp2-vpn	Creates a routing and forwarding table by assigning a route distinguisher to *isp2-vpn* .
Step 12	Router(config-vrf)# **route-target export** isp2-vpn-rd	Exports all routes for the VPNs (*isp2-vpn)* route distinguisher.
Step 13	Router(config-vrf)# **route-target import** isp2-vpn-rd	Imports all routes for the VPNs (*isp2-vpn)* route distinguisher.
Step 14	Router(config-vrf)# **route-target import** mgmt-vpn-rd	Imports all routes for the VPNs (*mgmt-vpn)* route distinguisher.

Defining Subinterfaces on a Virtual Bundle Interface and Assigning VRFs

To create a logical cable subinterface, perform the following steps beginning in the global configuration mode. Create one subinterface for each VPN (one per ISP). The first subinterface created must be configured as part of the management VPN (with the lowest subinterface number).

Procedure

	Command or Action	Purpose
Step 1	Router# **configure terminal**	Enters configuration mode.
Step 2	Router(config)# **interface bundle n.***x*	Enters virtual bundle interface configuration mode and defines the first (management) subinterface with the lowest subinterface number.
Step 3	Router(config-subif)# **description** *string*	Identifies the subinterface as the management subinterface.
Step 4	Router(config-subif)# **vrf forwarding** *mgmt-vpn*	Assigns the subinterface to the management VPN (the MPLS VPN used by the MSO to supply service to customers).
Step 5	Router(config-subif)# **ip address** *ipaddress mask*	Assigns the subinterface an IP address and a subnet mask.

	Command or Action	Purpose
Step 6	Router(config-subif)# **cable helper-address** *ip-address* **cable-modem**	Forwards DHCP requests from cable modems to the IP address listed.
Step 7	Router(config-subif)# **cable helper-address** *ip-address* **host**	Forwards DHCP requests from hosts to the IP address listed.
Step 8	Router(config-if)# **interface bundle n.***x*	Defines an additional subinterface for the ISP (such as isp1).
Step 9	Router(config-subif)# **description** *string*	Identifies the subinterface (such as subinterface for *isp1-vpn)* .
Step 10	Router(config-subif)# **vrf forwarding** *isp1-vpn*	Assigns the subinterface to *isp1-vpn* VPN.
Step 11	Router(config-subif)# **ip address** *ipaddress mask*	Assigns the subinterface an IP address and a subnet mask.
Step 12	Router(config-subif)# **cable helper-address** *ip-address* **cable-modem**	Forwards DHCP requests from cable modems to the IP address listed.
Step 13	Router(config-subif)# **cable helper-address** *ip-address* **host**	Forwards DHCP requests from hosts to the IP address listed.
Step 14	Router(config-if)# **interface bundle n.***x*	Defines an additional subinterface for the ISP (such as isp2).
Step 15	Router(config-subif)# **description** *string*	Identifies the subinterface (such as subinterface for *isp2-vpn)* .
Step 16	Router(config-subif)# **vrf forwarding** *isp2-vpn*	Assigns the subinterface to *isp2-vpn* VPN.
Step 17	Router(config-subif)# **ip address** *ipaddress mask*	Assigns the subinterface an IP address and a subnet mask.
Step 18	Router(config-subif)# **cable helper-address** *ip-address* **cable-modem**	Forwards DHCP requests from cable modems to the IP address listed.
Step 19	Router(config-subif)# **cable helper-address** *ip-address* **host**	Forwards DHCP requests from hosts to the IP address listed.
Step 20	Router(config)# **exit**	Returns to configuration mode.

Configuring Cable Interface Bundles

To assign a cable interface to a bundle, perform the following steps beginning in the interface configuration mode.

Procedure

	Command or Action	Purpose
Step 1	Router(config)# **interface cable** *slot/port*	Enters the cable interface configuration mode. IP addresses are not assigned to this interface. They are assigned to the logical subinterfaces created within this interface.
Step 2	Router(config-if)# **cable bundle***bundle-number*	Defines the interface as the bundle interface.
Step 3	Router(config)# **interface cable** *slot/subslot/port*	Enters the cable interface configuration mode for another cable interface. IP addresses are not assigned to this interface. They are assigned to the logical subinterfaces created within this interface.
Step 4	Router(config-if)# **cable bundle** *bundle-number*	Adds the interface to the bundle specified by *bundle-number* .

Configuring Subinterfaces and MPLS VPNs on a Virtual Bundle Interface

To configure subinterfaces on a virtual bundle interface and assign each subinterface a Layer 3 configuration:

Configure cable interface bundles.

Define subinterfaces on the virtual bundle interface and assign a Layer 3 configuration to each subinterface.

Create one subinterface for each customer VPN (one per ISP).

Configuring MPLS in the P Routers in the Provider Core

To configure MPLS in the P routers in the provider core, perform the following steps.

Procedure

	Command or Action	Purpose
Step 1	Router# **configure terminal**	Enters configuration mode.
Step 2	Router(config)#**ip cef**	Enables Cisco Express Forwarding (CEF) operation. For information about CEF configuration and command syntax, see Cisco Express Forwarding Overview and Configuring Cisco Express Forwarding.
Step 3	Router(config)#**interface Tengigabitethernet** *slot/subslot/port*	Enters GigabitEthernet interface configuration mode.
Step 4	Router(config-if)#**ip address** *ip-address mask*	Defines the primary IP address range for the interface.

	Command or Action	Purpose
Step 5	Router(config-if)#**mpls ip**	Enables the interface to be forwarded to an MPLS packet.
Step 6	Router(config-if)#**exit**	Returns to global configuration mode.
Step 7	Router(config)#**mpls label-protocol ldp**	Enables Label Distribution Protocol (LDP). For information about LDP and MPLS, see Configuring Multiprotocol Label Switching.
Step 8	Router(config)# **exit**	Returns to the configuration mode.

Verifying the MPLS VPN Configuration

Use the following commands to verify MPLS VPN operations on PE routers. For more MPLS VPN verification commands, see Configuring Multiprotocol Label Switching.

Procedure

	Command or Action	Purpose
Step 1	Router# **show ip vrf**	Displays the set of VRFs and interfaces.
Step 2	Router# **show ip route vrf [vrf-name]**	Displays the IP routing table for a VRF.
Step 3	Router# **show ip protocols vrf [vrf-name]**	Displays the routing protocol information for a VRF.
Step 4	Router# **show ip route vrf** *vrf-name*	Displays the Local and Remote CE devices that are in the PE routing table.
Step 5	Router# **show mpls forwarding-table**	Displays entries for a VPN Routing/Forwarding instance.

What to do next

For more verification instructions, see the MPLS: Layer 3 VPNs Configuration Guide.

Configuration Examples

This section provides the following configuration examples:

VRF Definition Configuration

```
vrf definition Basketball
 rd 100:2
 route-target export 100:2
 route-target import 100:0
 route-target import 100:2
```

```
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
vrf definition Football
 rd 100:1
 route-target export 100:1
 route-target import 100:0
 route-target import 100:1
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
vrf definition MGMT
 rd 100:0
 route-target export 100:0
 route-target import 100:0
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
vrf definition Mgmt-intf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
vrf definition Tennis
 rd 100:4
 route-target export 100:4
 route-target import 100:0
 route-target import 100:4
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
vrf definition Volleyball
 rd 100:3
 route-target export 100:3
 route-target import 100:0
 route-target import 100:3
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
```

Cable Bundle SubInterface Configuration

```
interface Bundle255
 description Bundle Master Interface
 no ip address
```

```
 cable arp filter request-send 3 2
 cable arp filter reply-accept 3 2

interface Bundle255.1
 description Management Interface
 vrf forwarding MGMT
 ip address 112.51.0.1 255.255.0.0
 cable helper-address 20.11.0.162
 ipv6 address 2001:100:112:B001::1/64

interface Bundle255.2
 vrf forwarding Basketball
 ip address 112.54.0.1 255.255.0.0 secondary
 ip address 112.53.0.1 255.255.0.0
 cable helper-address 20.11.0.62
 cable helper-address 20.11.0.162
 ipv6 address 2001:100:112:B003::1/64
 ipv6 address 2001:100:112:B004::1/64

interface Bundle255.3
 vrf forwarding Football
 ip address 112.56.0.1 255.255.0.0 secondary
 ip address 112.55.0.1 255.255.0.0
 cable helper-address 20.11.0.62
 cable helper-address 20.11.0.162
 ipv6 address 2001:100:112:B005::1/64
 ipv6 address 2001:100:112:B006::1/64

interface Bundle255.4
 vrf forwarding Volleyball
 ip address 112.58.0.1 255.255.0.0 secondary
 ip address 112.57.0.1 255.255.0.0
 cable helper-address 20.11.0.62
 cable helper-address 20.11.0.162
 ipv6 address 2001:100:112:B007::1/64
 ipv6 address 2001:100:112:B008::1/64

interface Bundle255.5
 vrf forwarding Tennis
 ip address 112.61.0.1 255.255.0.0 secondary
 ip address 112.60.0.1 255.255.0.0 secondary
 ip address 112.59.0.1 255.255.0.0
 cable helper-address 20.11.0.162
 ipv6 address 2001:100:112:B009::1/64
 ipv6 address 2001:100:112:B00A::1/64
```

PE WAN Interface Configuration

```
mpls label protocol ldp
mpls ldp nsr
mpls ldp graceful-restart

interface TenGigabitEthernet4/1/1
 description WAN connection to cBR8
 mtu 4470
 ip address 100.6.120.5 255.255.255.252
 ip router isis hub
 ipv6 address 2001:100:6:120::5:1/112
 ipv6 enable
 mpls ip
 mpls traffic-eng tunnels
```

```
 cdp enable
 isis circuit-type level-1
 isis network point-to-point
 isis csnp-interval 10
 hold-queue 400 in
 ip rsvp bandwidth 1000000
end
```

PE BGP Configuration

```
router bgp 100
 bgp router-id 100.120.120.120
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 timers bgp 5 60
 neighbor 100.100.4.4 remote-as 100
 neighbor 100.100.4.4 ha-mode sso
 neighbor 100.100.4.4 update-source Loopback0
 neighbor 100.100.4.4 ha-mode graceful-restart
 !
 address-family ipv4
  redistribute connected
  redistribute static route-map static-route
  redistribute rip
  neighbor 100.100.4.4 activate
  neighbor 100.100.4.4 send-community extended
  neighbor 100.100.4.4 next-hop-self
  neighbor 100.100.4.4 soft-reconfiguration inbound
  maximum-paths ibgp 2
 exit-address-family
 !
 address-family vpnv4
  neighbor 100.100.4.4 activate
  neighbor 100.100.4.4 send-community extended
 exit-address-family
 !
 address-family ipv6
  redistribute connected
  redistribute rip CST include-connected
  redistribute static metric 100 route-map static-route-v6
  neighbor 100.100.4.4 activate
  neighbor 100.100.4.4 send-community extended
  neighbor 100.100.4.4 send-label
 exit-address-family
 !
 address-family vpnv6
  neighbor 100.100.4.4 activate
  neighbor 100.100.4.4 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf Basketball
  redistribute connected
 exit-address-family
 !
 address-family ipv6 vrf Basketball
  redistribute connected
  redistribute static metric 100
 exit-address-family
 !
 address-family ipv4 vrf Football
```

```
 redistribute connected
exit-address-family
!
address-family ipv6 vrf Football
 redistribute connected
 redistribute static metric 100
exit-address-family
!
address-family ipv4 vrf MGMT
 redistribute connected
exit-address-family
!
address-family ipv6 vrf MGMT
 redistribute connected
exit-address-family
!
address-family ipv4 vrf Tennis
 redistribute connected
 redistribute static route-map static-route
 redistribute rip
exit-address-family
!
address-family ipv6 vrf Tennis
 redistribute connected
 redistribute rip CST include-connected
 redistribute static metric 100 route-map static-route-v6
exit-address-family
!
address-family ipv4 vrf Volleyball
 redistribute connected
 redistribute static route-map static-route
 redistribute rip
exit-address-family
!
address-family ipv6 vrf Volleyball
 redistribute connected
 redistribute rip CST include-connected
 redistribute static metric 100 route-map static-route-v6
exit-address-family
```

Additional References

Standards

Standard	Title
DOCSIS 1.0	*DOCSIS 1.0*

MIBs

MIB	MIBs Link
CISCO-DOCS-REMOTE-QUERY.my	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://tools.cisco.com/ITDIT/MIBS/servlet/index

RFCs

RFC	Title
RFC 1163	A Border Gateway Protocol
RFC 1164	Application of the Border Gateway Protocol in the Internet
RFC 2283	Multiprotocol Extensions for BGP-4
RFC 2547	BGP/MPLS VPNs
RFC 2233	DOCSIS OSSI Objects Support
RFC 2669	Cable Device MIB

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

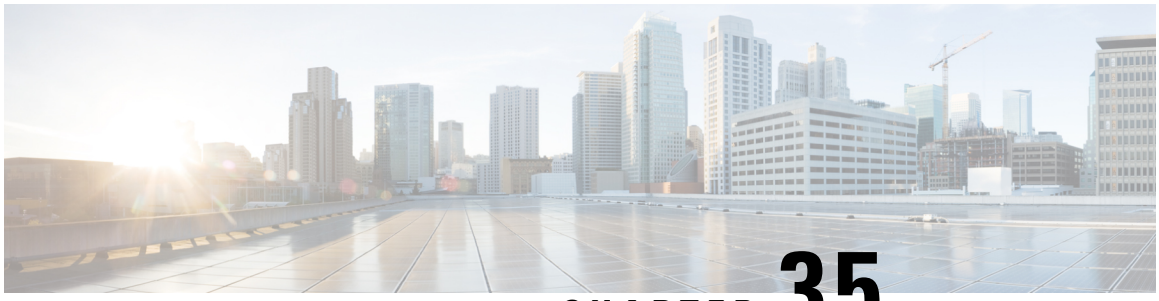
Feature Information for MPLS VPN Cable Enhancements

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 92: Feature Information for MPLS VPN Cable Enhancements

Feature Name	Releases	Feature Information
Multiprotocol Label Switching Virtual Private Network (MPLS VPN)	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router .



CHAPTER 35

Multicast VPN and DOCSIS 3.0 Multicast QoS Support

The CMTS enhanced multicast new features are consistent with DOCSIS 3.0 specifications and include:

- Enhanced multicast echo in which the Layer 3 multicast switching path uses a Cisco Packet Processor (CPP) parallel express forwarding multicast routing table.
- Enhanced multicast quality of service (MQoS) framework that specifies a group configuration (GC) to define a session range of multicast addresses and rule priorities and its associated multicast VPN (MVPN).
- Intelligent multicast admission control to include multicast service flows.
- Enhanced multicast VPN feature to configure and support multicast traffic in a multiprotocol label switching (MPLS)-VPN environment.

- Finding Feature Information, on page 581
- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 582
- Restrictions for the Multicast VPN and DOCSIS 3.0 Multicast QoS Support, on page 582
- Information About the Multicast VPN and DOCSIS 3.0 Multicast QoS Support, on page 583
- How to Configure the Multicast VPN and DOCSIS 3.0 Multicast QoS Support, on page 584
- Configuration Examples for the Multicast VPN and DOCSIS 3.0 Multicast QoS Support, on page 589
- Additional References, on page 589
- Feature Information for Multicast VPN and DOCSIS3.0 Multicast QoS Support, on page 590

Finding Feature Information

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 93: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for the Multicast VPN and DOCSIS 3.0 Multicast QoS Support

The type of service (ToS) parameter is not recognized by the Cisco cBR series routers.

Information About the Multicast VPN and DOCSIS 3.0 Multicast QoS Support

IP multicast—transmission of the same information to multiple cable network recipients—improves bandwidth efficiency and allows service providers to offer differentiated quality of service for different types of traffic. Enhanced multicast introduces multicast improvements as mandated by the introduction of DOCSIS 3.0 specifications.

Note DOCSIS 3.0 standards retain backwards compatibility with the DOCSIS 2.0 multicast mode of operation.

The following are the benefits of CMTS enhanced multicast are:

Enhanced Quality of Service

In the new multicast QoS (MQoS) framework, you can specify a group configuration (GC) that defines a session range of multicast addresses and rule priorities and its associated multicast VPN (MVPN). For every GC, there is attached a group QoS configuration (GQC) and a group encryption rule.

Based on the session range, rule priority, and MVPN, a multicast service flow is admitted into a GC and the associated GQC and group encryption rule are applied to the flow. In MQoS implementation, the source address of the multicast session is not checked because the current implementation for cable-specific multicast supports IGMP Version 2 but not IGMP Version 3. The downstream service flow, service identifier (SID), and MAC-rewrite string are created at the time of a new IGMP join (or static multicast group CLI on the interface) and MQoS is applied to the new multicast group join.

The benefits of enhanced QoS are the following:

- Group classifiers can be applied at cable interface level and also at bundle interface level.
- Group service flow (GSF) definition is based on service class names. The GSF is similar to individual service flows and commonly includes the minimum rate and maximum rate parameters for the service class. GSF is shared by all cable modems on a particular downstream channel set (DCS) that is matched to the same group classifier rule (GCR). A default service flow is used for multicast flows that do not match to any GCR. A GSF is always in the active state.
- CMTS replicates multicast packets and then classifies them.
- Single-stage replication and two-stage replication are supported.
- Enhanced QoS is compatible and integrated with DOCSIS Set-Top Gateway (DSG).

Intelligent Multicast Admission Control

Admission control allows you to categorize service flows into buckets. Examples of categories are the service class name used to create the service flow, service flow priority, or the service flow type such as unsolicited grant service (UGS). Bandwidth limits for each bucket can also be defined. For example, you can define bucket 1 for high priority packet cable service flows and specify that bucket 1 is allowed a minimum of 30 percent and a maximum of 50 percent of the link bandwidth.

Intelligent multicast admission control includes additional features such as the inclusion of multicast service flows using the GSF concept. GSFs are created based on the rules as defined in the GQC table. The rules link

the multicast streams to a GSF through the session range. The service class name in the rule defines the QoS for that GSF. Additionally, another attribute is added to the rules and the group configuration table to specify the application type to which each GSF belongs. In this way, the QoS associated with each GSF is independent of the bucket category for the GSF.

The benefits of intelligent multicast admission control are the following:

- There is explicit acknowledgment of the establishment of each multicast session.
- Admission control does not consume additional bandwidth for multicast flows once the first flow is established.
- Service flows are cleaned up as the multicast session is torn down.

Multicast Session Limit Support

In a multicast video environment, you can limit the number of multicast sessions admitted onto a particular service flow. The multicast session limit feature—which adds functionality on top of the multicast QoS infrastructure—enables you to specify the number of multicast sessions to be admitted on a particular service flow. If the current number of sessions has reached the defined limit, new sessions will be forwarded but they will make use of the default multicast service flow until a session ends to free up a slot for new sessions.

Multicast Virtual Private Network

The new multicast VPN (MVPN) feature allows you to configure and support multicast traffic in a multiprotocol label switching (MPLS)-VPN environment. This feature supports routing and forwarding of multicast packets for each individual VPN virtual routing and forwarding (VRF) instance, and also provides a mechanism to transport VPN multicast packets across the service provider backbone.

MVPN allows you to connect multiple remote sites or devices over either a Layer 3 or Layer 2 VPN. A Layer 3 VPN enables the routing of traffic inside the VPN. A Layer 2 VPN provides a bridging transport mechanism for traffic between remote sites belonging to a customer. To support multicast over Layer 3 VPNs, each VPN receives a separate multicast domain with an associated MVPN routing and forwarding (mVRF) table maintained by the provider edge (PE) router. In a cable environment, the PE router is a routing CMTS. The provider network builds a default multicast distribution tree (default-MDT) for each VPN between all the associated mVRF-enabled PE routers. This tree is used to distribute multicast traffic to all PE routers.

To enable maximum security and data privacy in a VPN environment, the CMTS distinguishes between multicast sessions on the same downstream interface that belong to different VPNs. To differentiate multicast traffic between different VPNs, the CMTS implements a per-VRF subinterface multicast security association identifier (MSAID) allocation feature that is BPI+ enabled. The MSAID is allocated for each cable bundle group for each subinterface. A multicast group has a specific MSAID for each VRF instance.

How to Configure the Multicast VPN and DOCSIS 3.0 Multicast QoS Support

This section contains the following procedures:

Configuring a QoS Profile for a Multicast Group

To configure a QoS profile that can be applied to a QoS group configuration, use the **cable multicast group-qos** command. You must configure a QoS profile before you can add a QoS profile to a QoS multicast group.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable multicast group-qos** *number* **scn** *service-class-name* **control**{ **single** | **aggregate** [**limit** *max-sessions*]} **Example:** `Router(config)#: cable multicast group-qos 2 scn name1 control single`	Configures a QoS profile that can be applied to a multicast QoS group. **Note** If a number is not specified, a default QoS profile is applied. The default group qos configuration creates a default multicast service flow for each cable interface that is used when a multicast session does not match any classifiers of a GC on the interface.

Configuring a Multicast QoS Group

You can specify a group configuration (GC) that defines a session range of multicast addresses and rule priorities and its associated multicast VPN (MVPN). For every GC, there is attached a group QoS configuration and a group encryption rule.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configureterminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**cable multicast group-qos** *number* **scn** *service-class-name* **control** {**single** | **aggregate** [limit *max-sessions*]} **Example:** `Router(config-mqos)# cable multicast group-qos 5 scn name1 control single`	(Optional) Configures a QoS profile that can be applied to a multicast QoS group. **Note** If a number is not specified, a default QoS profile is applied. The default group qos configuration creates a default multicast service flow for each cable interface that is used when a multicast session does not match any classifiers of a GC on the interface.
Step 4	**cable multicast qos group** *id* **priority** *value* [**global**] **Example:** `Router(config)# cable multicast qos group 2 priority 6`	Configures a multicast QoS group and enters multicast QoS configuration mode.
Step 5	**session-range** *ip-address ip-mask* **Example:** `Router(config-mqos)# session-range 224.10.10.10 255.255.255.224`	Specifies the session range IP address and IP mask of the multicast QoS group. You can configure multiple session ranges.
Step 6	**tos** *low-byte high-byte mask* **Example:** `Router(config-mqos)# tos 1 6 15`	(Optional) Specifies the minimum type of service (ToS) data bytes, maximum ToS data bytes, and mask for a multicast QoS group.
Step 7	**vrf***name* **Example:** `Router(config-mqos)# vrf name1`	(Optional) Specifies the name for the virtual routing and forwarding (VRF) instance. **Note** If a multicast QoS (MQoS) group is not defined for this VRF, you will see an error message. You must either define a specific MQoS group for each VRF, or define a default MQoS group that can be assigned in those situations where no matching MQoS group is found. See the Configuring a Default Multicast QoS Group for VRF, on page 587.
Step 8	**application-id***number* **Example:** `Router(config-mqos)# application-id 25`	(Optional) Specifies the application identification number of the multicast QoS group. This value is configured to enable admission control to the multicast QoS group.

Configuring a Default Multicast QoS Group for VRF

Each virtual routing and forwarding (VRF) instance that is defined must match a defined MQoS group to avoid multicast stream crosstalk between VRFs. To avoid potential crosstalk, define a default MQoS group that is assigned to the VRF whenever the multicast traffic in the VRF does not match an existing MQoS group.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable multicastgroup-qos***number* **scn***service-class-name* **control** {**single** | **aggregate** [**limit** *max-sessions*]} **Example:** `Router(config-mqos)# cable multicast group-qos 5 scn name1 control single`	(Optional) Configures a QoS profile that can be applied to a multicast QoS group.
Step 4	**cable multicast qos group** *id* **priority 255 global** **Example:** `Router(config)# cable multicast qos group 2 priority 255 global`	Configures a default multicast QoS group and enters multicast QoS configuration mode.
Step 5	**session-range 224.0.0.0 224.0.0.0** **Example:** `Router(config-mqos)# session-range 224.0.0.0 224.0.0.0`	Specifies the session-range IP address and IP mask of the default multicast QoS group. By entering 224.0.0.0 for the IP address and the IP mask you cover all possible multicast sessions.
Step 6	**vrf***name* **Example:** `Router(config-mqos)# vrf name1`	Specifies the name of the virtual routing and forwarding (VRF) instance.
Step 7	**application-id***number* **Example:** `Router(config-mqos)# application-id 5`	(Optional) Specifies the application identification number of the multicast QoS group. This value is configured to enable admission control to the multicast QoS group.

Verifying Configuration of the Multicast VPN and DOCSIS 3.0 Multicast QoS Support

To verify the configuration of the Multicast VPN and DOCSIS 3.0 Multicast QoS Support feature, use the **show** commands described below.

- To show the configuration parameters for multicast sessions on a specific bundle, use the **show interface bundle** *number* **multicast-sessions** command as shown in the following example:

```
Router# show interface bundle 1 multicast-sessions
Multicast Sessions on Bundle1
 Group            Interface    GC  SAID SFID  GQC GEn RefCount GC-Interface State
 234.1.1.45       Bundle1.1    1   8193  ---   1   5   1        Bundle1     ACTIVE
 234.1.1.46       Bundle1.1    1   8193  ---   1   5   1        Bundle1     ACTIVE
 234.1.1.47       Bundle1.1    1   8193  ---   1   5   1        Bundle1     ACTIVE
Aggregate Multicast Sessions on Bundle1
 Aggregate Sessions for SAID 8193 GQC 1 CurrSess 3
 Group            Interface    GC  SAID SFID AggGQC GEn RefCount GC-Interface
 234.1.1.45       Bundle1.1    1   8193  ---   1      5   1        Bundle1
 234.1.1.46       Bundle1.1    1   8193  ---   1      5   1        Bundle1
 234.1.1.47       Bundle1.1    1   8193  ---   1      5   1        Bundle1
```

- To show the configuration parameters for multicast sessions on a specific cable, use the **show interface cable** *ip-addr* **multicast-sessions** command as shown in the following example:

```
Router# show interface cable 7/0/0 multicast-sessions
Default Multicast Service Flow 3 on Cable7/0/0
Multicast Sessions on Cable7/0/0
 Group            Interface    GC  SAID SFID  GQC GEn RefCount GC-Interface State
 234.1.1.45       Bundle1.1    1   8193 24     1   5   1        Bundle1     ACTIVE
 234.1.1.46       Bundle1.1    1   8193 24     1   5   1        Bundle1     ACTIVE
 234.1.1.47       Bundle1.1    1   8193 24     1   5   1        Bundle1     ACTIVE
Aggregate Multicast Sessions on Cable7/0/0
 Aggregate Sessions for SAID 8193 SFID 24 GQC 1 CurrSess 3
 Group            Interface    GC  SAID SFID AggGQC GEn RefCount GC-Interface
 234.1.1.45       Bundle1.1    1   8193 24     1      5   1        Bundle1
 234.1.1.46       Bundle1.1    1   8193 24     1      5   1        Bundle1
 234.1.1.47       Bundle1.1    1   8193 24     1      5   1        Bundle1
```

- To show the MSAID multicast group subinterface mapping, use the **show interface cable** *address* **modem** command as shown in the following example:

```
Router# show interface cable 6/1/0 modem
SID   Priv Type      State          IP address      method  MAC address     Dual
      bits                                                                  IP
9     11   modem     online(pt)     101.1.0.6       dhcp    0006.28f9.8c79  N
9     11   host      unknown        111.1.1.45      dhcp    0018.1952.a859  N
10    10   modem     online(pt)     101.1.0.5       dhcp    0006.5305.ac19  N
10    10   host      unknown        111.1.0.3       dhcp    0018.1952.a85a  N
13    10   modem     online(pt)     101.1.0.3       dhcp    0014.f8c1.fd1c  N
8195  10   multicast unknown        224.1.1.51      static  0000.0000.0000  N
8195  10   multicast unknown        224.1.1.49      static  0000.0000.0000  N
8195  10   multicast unknown        224.1.1.50      static  0000.0000.0000  N
```

Configuration Examples for the Multicast VPN and DOCSIS 3.0 Multicast QoS Support

This section provides the following configuration examples:

Example: Configuring Group QoS and Group Encryption Profiles

Note To add group QoS and group encryption profiles to a QoS group, you must configure each profile first before configuring the QoS group.

In the following example, QoS profile 3 and encryption profile 35 are configured.

```
configure terminal
cable multicast group-qos 3 scn name1 control single
cable multicast group-encryption 35 algorithm 56bit-des
```

Example: Configuring a QoS Group

In the following example, QoS group 2 is configured with a priority of 6 and global application. To QoS group 2, QoS profile 3 and encryption profile 35 are applied. Other parameters are configured for QoS group 2 including application type, session range, ToS, and VRF.

```
cable multicast qos group 2 priority 6 global
group-encryption 35
group-qos 3
session-range 224.10.10.01 255.255.255.254
tos 1 6 15
vrf vrf-name1
application-id 44
```

Additional References

The following sections provide references related to the Multicast VPN and DOCSIS 3.0 Multicast QoS Support.

Related Documents

Related Topic	Document Title
CMTS cable commands	*Cisco CMTS Cable Command Reference* http://www.cisco.com/c/en/us/td/docs/cable/cmts/cmd_ref/b_cmts_cable_cmd_ref.html

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2236	*Internet Group Management Protocol, Version 2*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Multicast VPN and DOCSIS3.0 Multicast QoS Support

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 94: Feature Information for Multicast VPN and DOCSIS3.0 Multicast QoS Support

Feature Name	Releases	Feature Information
Multicast VPN and DOCSIS3.0 multicast QoS support	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.

CHAPTER 36

EtherChannel for the Cisco CMTS

This document describes the features, benefits and configuration of Cisco EtherChannel technology on the Cisco Cable Modem Termination System (CMTS).

EtherChannel is a technology by which to configure and aggregate multiple physical Ethernet connections to form a single logical port with higher bandwidth. The first EtherChannel port configured on the Cisco CMTS serves as the EtherChannel bundle primary by default, and each subsidiary interface interacts with the network using the MAC address of the EtherChannel bundle primary.

EtherChannel ports reside on a routing or bridging end-point. The router or switch uses EtherChannel to increase bandwidth utilization in either half- or full-duplex mode, and load balances the traffic across the multiple physical connections.

EtherChannel on the Cisco CMTS supports inter-VLAN routing with multiple devices and standards, and supports Ten Gigabit EtherChannel (GEC) on the Cisco cBR series routers.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 594
- Restrictions for EtherChannel on the Cisco CMTS, on page 594
- Information About EtherChannel on the Cisco CMTS, on page 595
- How to Configure EtherChannel on the Cisco CMTS, on page 595
- Verifying EtherChannel on the Cisco CMTS, on page 598
- Configuration Examples for EtherChannel on the Cisco CMTS, on page 598
- Additional References, on page 599
- Feature Information for EtherChannel on Cisco CMTS, on page 600

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 95: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for EtherChannel on the Cisco CMTS

- EtherChannel on the Cisco CMTS is limited to Network Layer 3 functions, and does not support Data-Link Layer 2 EtherChannel functions as with certain other Cisco product platforms.
- The Port Aggregation Protocol (PAgP) is not supported on the Cisco CMTS as with other Cisco product platforms (such as the CatOS switch).
- Only the IEEE 802.1Q trunking protocol is supported on the Cisco CMTS. ATM trunking is not supported on the Cisco cBR series routers.
- The maximum supported links per bundle is 8.
- EtherChannel on Cisco CMTS supports only physical ports or interfaces that have the same speed.
- EtherChannel on the Cisco cBR series routers does not support MQC QOS. You can use Equal Cost Multi Path (ECMP) load balancing instead of EtherChannel.

- Layer 3 configurations on member interfaces of EtherChannel are not supported.
- MAC Address Accounting feature on port channel is not supported.

Information About EtherChannel on the Cisco CMTS

This section contains the following:

Introduction to EtherChannel on the Cisco CMTS

EtherChannel is based on proven industry-standard technology. The Cisco CMTS supports EtherChannel with several benefits, including the following:

- EtherChannel on the Cisco CMTS supports subsecond convergence times.
- EtherChannel can be used to connect two switch devices together, or to connect a router with a switch.
- A single EtherChannel connection supports a higher bandwidth between the two devices.
- The logical port channels on either Cisco CMTS platform provide fault-tolerant, high-speed links between routers, switches, and servers.
- EtherChannel offers redundancy and high availability on the Cisco CMTS. Failure of one connection causes a switch or router to use load balancing across the other connections in the EtherChannel.
- Load balancing on the Cisco CMTS supports dynamic link addition and removal without traffic interruption.
- EtherChannel supports inter-VLAN trunking. Trunking carries traffic from several VLANs over a point-to-point link between the two devices. The network provides inter-VLAN communication with trunking between the Cisco CMTS router and one or more switches. In a campus network, trunking is configured over an EtherChannel link to carry the multiple VLAN information over a high-bandwidth channel.

Cisco Ten Gigabit EtherChannel on the Cisco cBR Series Routers

Cisco Ten Gigabit EtherChannel (GEC) is high-performance Ethernet technology that provides gigabit-per-second transmission rates. It provides flexible, scalable bandwidth with resiliency and load sharing across links for switches, router interfaces, and servers.

Ten GEC on the Cisco cBR series routers with the following EtherChannel capabilities:

- Supports IEEE 802.1Q encapsulation for inter-VLAN networking.
- Supports a maximum of eight physical Ten Gigabit Ethernet ports to be combined as one logical EtherChannel link.
- Supports bandwidth up to 40 Gbps (half duplex) for a combined total of up to 80 Gbps (full duplex).

How to Configure EtherChannel on the Cisco CMTS

This section contains the following:

Configuring Ten Gigabit EtherChannel on the Cisco CMTS

Before you begin

- Ten Gigabit Ethernet cabling is completed and the ports are operational on the router and network.
- LAN interfaces are configured and operational on the router and network, with IP addresses and subnet masks.

Note The Cisco cBR series routers support up to eight physical connectors to be configured as one logical Ten GEC port.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface port-channel** *n* **Example:** `Router(config)# interface port-channel 1`	Creates an EtherChannel interface. The first EtherChannel interface configured becomes the bundle primary for all ports in the EtherChannel group. The MAC address of the first EtherChannel interface is the MAC address for all EtherChannel interfaces in the group. To remove an EtherChannel interface from the EtherChannel group, use the **no** form of this command. If the first EtherChannel interface in the group is later removed, the second EtherChannel interface in the group becomes the bundle primary by default. Repeat this step on every EtherChannel port to be bundled into a Ten GEC group. This configuration must be present on all EtherChannel interfaces before the EtherChannel group can be configured.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 5	**interface tengigabitethernet** *slot/subslot/port* **Example:** `Router# interface gigabitethernet 4/1/0`	Selects the Ten Gigabit Ethernet interface that you wish to add as a member EtherChannel link in the EtherChannel bundle, and enters interface configuration mode. **Note** We recommend that the link being added to the Cisco CMTS EtherChannel be shut down prior to configuring it as a member of the EtherChannel. Use the **shutdown** command in interface configuration mode immediately before completing the following steps in this procedure.
Step 6	**shutdown** **Example:** `Router(config-if)# shutdown`	Shuts down the interface selected in step 5 before configuring it as a member of the EtherChannel.

Step 7	Use one of the following commands:
	• For static Ten GEC configuration, use the **channel-group** *number* command.
	• For dynamic Ten GEC configuration, use the **channel-group** *number* **mode** {**active** | **passive**} command.
	Example:
	`Router(config-if)# channel-group 1`
	`or`
	`Router(config-if)# channel-group 1 mode active`
		Adds the Ten Gigabit Ethernet interface to the EtherChannel Group, associating that interface with an EtherChannel link.
		To remove an EtherChannel group and the associated ports from the Cisco CMTS, use the **no** form of this command.

Step 8	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface on which EtherChannel is configured.
Step 9	**end** **Example:** `Router(config# end`	Returns to privileged EXEC mode. IP traffic should be visible on the network with completion of the above steps.

Troubleshooting Tips

Once interface operations are confirmed (prior to this procedure), and EtherChannel configurations have been verified (next procedure), any difficulty experienced through the EtherChannel links may pertain to inter-VLAN or IP routing on the network, or perhaps very high bandwidth consumption.

What to Do Next

Additional IP, access list, inter-VLAN or load balancing configurations may be made to the Cisco CMTS and these changes will be supported in the running EtherChannel configuration without service disruption from EtherChannel.

Verifying EtherChannel on the Cisco CMTS

Links can be added or removed from an EtherChannel interface without traffic interruption. If an Ethernet link in an EtherChannel interface fails, traffic previously carried over the failed link switches to the remaining links within the EtherChannel. There are a number of events that can cause a link to be added or removed including adding or removing a link using commands and simulating link failure and recovery (as with (no)shutdown links).

Cisco EtherChannel supports online insertion and removal (OIR) of field-replaceable units (FRUs) in the Cisco CMTS chassis. Ports that remain active during OIR of one FRU will take over and support the traffic bandwidth requirements without service disruption. However, OIR is not described in this procedure.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show interface port-channel** *n* **Example:** `Router# show interface port-channel 1`	Verifies the EtherChannel configuration on the Cisco CMTS for the selected EtherChannel group.

Configuration Examples for EtherChannel on the Cisco CMTS

The following example illustrates Ten Gigabit EtherChannel information for the port-channel interface of 2.

This configuration is comprised of three Ten GEC port channels as follows:

- Member 0 is the Ten GEC interface bundle primary.
- Member 2 is the final subsidiary interface in this Ten GEC group.
- These three port-channel interfaces (members) comprise one Ten GEC group that is set up with a Ten GEC peer on the network.

```
Router# show interface port-channel 2
Port-channel2 is up, line protocol is up
  Hardware is GEChannel, address is 8888.8888.8888 (bia 0000.0000.0000)
  Internet address is 101.101.101.1/16
  MTU 1500 bytes, BW 3000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
    No. of members in this channel: 3
    No. of configured  members in this channel: 3
    No. of passive members in this channel: 0
    No. of active members in this channel: 3
        Member 0 : TenGigabitEthernet4/1/0 , Full-duplex, 1000Mb/s
        Member 1 : TenGigabitEthernet4/1/1 , Full-duplex, 1000Mb/s
        Member 2 : TenGigabitEthernet4/1/2 , Full-duplex, 1000Mb/s
    No. of Non-active members in this channel: 0
  Last input 00:00:02, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/225/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/120 (size/max)
  30 second input rate 17292000 bits/sec, 9948 packets/sec
  30 second output rate 17315000 bits/sec, 9935 packets/sec
     866398790 packets input, 3324942446 bytes, 0 no buffer
     Received 2 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 0 multicast, 0 pause input
     0 input packets with dribble condition detected
     866394055 packets output, 3323914794 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 pause output
     0 output buffer failures, 0 output buffers swapped out
```

Additional References

Related Documents

Related Topic	Document Title
EtherChannel for Cisco Products	• Cisco EtherChannel home page http://www.cisco.com/warp/public/cc/techno/lnty/etty/fsetch/index.shtml • Cisco EtherChannel Technology white paper http://www.cisco.com/en/US/tech/tk389/tk213/technologies_white_paper09186a0080092944.shtml

Related Topic	Document Title
Configuring Additional Devices for EtherChannel	• *Configuring EtherChannel and 802.1Q Trunking Between a Catalyst 2950 and a Router (inter-VLAN Routing)* http://www.cisco.com/c/en/us/support/docs/switches/catalyst-2950-series-switches/24042-158.html • *Configuring EtherChannel and 802.1Q Trunking Between Catalyst 2900XL/3500XL and Catalyst 2940, 2950/2955, and 2970 Switches* http://www.cisco.com/c/en/us/support/docs/switches/catalyst-2900-xl-series-switches/21041-131.html

Standards and RFCs

Standards	Title
IEEE Std 802.1Q, 2003 Edition	IEEE Std 802.1Q, 2003 Edition (Incorporates IEEE Std 802.1Q-1998, IEEE Std 802.1u-2001, IEEE Std 802.1v-2001, and IEEE Std 802.1s-2002) http://ieeexplore.ieee.org/xpl/tocresult.jsp?isNumber=27089

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support

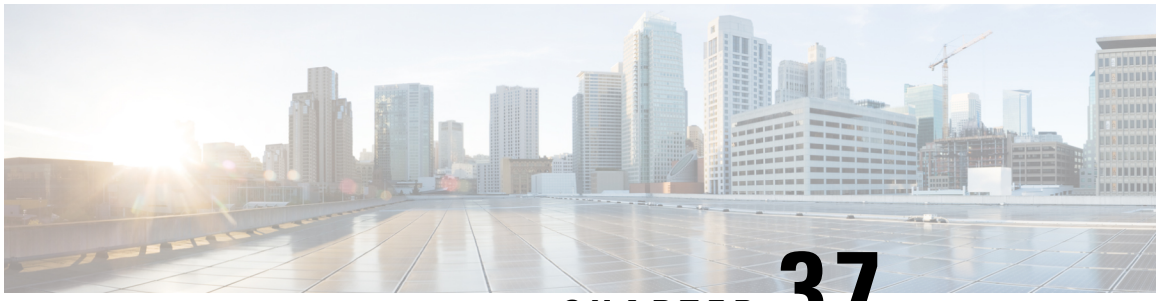
Feature Information for EtherChannel on Cisco CMTS

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 96: Feature Information for EtherChannel on Cisco CMTS

Feature Name	Releases	Feature Information
EtherChannel on Cisco CMTS	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router .



CHAPTER 37

Flow-Based per Port-Channel Load Balancing

The Flow-Based per Port-Channel Load Balancing feature allows different flows of traffic over a Ten Gigabit EtherChannel (GEC) interface to be identified based on the packet header and then mapped to the different member links of the port channel. This feature enables you to apply flow-based load balancing and VLAN-manual load balancing to specific port channels.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 603
- Restrictions for Flow-Based per Port-Channel Load Balancing, on page 604
- Information About Flow-Based per Port-Channel Load Balancing, on page 604
- How to Enable Flow-Based per Port-Channel Load Balancing, on page 607
- Verifying Load Balancing Configuration on a Ten GEC Interface, on page 608
- Configuration Examples for Flow-Based per Port-Channel Load Balancing, on page 610
- Additional References, on page 610
- Feature Information for Flow-Based per Port-Channel Load Balancing, on page 611

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 97: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for Flow-Based per Port-Channel Load Balancing

- Supports up to 64 Ten GEC interfaces.
- Supports up to 8 member links per Ten GEC interface.

Information About Flow-Based per Port-Channel Load Balancing

Flow-Based Load Balancing

Flow-based load balancing identifies different flows of traffic based on the key fields in the data packet. For example, IPv4 source and destination IP addressees can be used to identify a flow. The various data traffic flows are then mapped to the different member links of a port channel. After the mapping is done, the data traffic for a flow is transmitted through the assigned member link. The flow mapping is dynamic and changes when there is any change in the state of a member link to which a flow is assigned. The flow mappings can also change if member links are added to or removed from the GEC interface. Multiple flows can be mapped to each member link.

Buckets for Flow-Based Load Balancing

Load balancing dynamically maps traffic flows to the member links of a Ten GEC interface through the concept of buckets. The various defined traffic flows are mapped to the buckets and the buckets are evenly distributed among the member links. Each port channel maintains 16 buckets, with one active member link associated with each bucket. All traffic flows mapped to a bucket use the member link to which the bucket is assigned.

The router creates the buckets-to-member links mappings when you apply flow-based load balancing to a port channel and the port channel has at least one active member link. The mappings are also created when the first member link is added, or comes up, and the load-balancing method is set to flow-based.

When a member link goes down or is removed from a port channel, the buckets associated with that member link are redistributed among the other active member links in a round-robin fashion. When a member link comes up or is added to a port channel, some of the buckets associated with other links are assigned to this link.

If you change the load-balancing method, the bucket-to-member link mappings for flow-based load balancing are deleted. The mappings are also deleted if the port channel is deleted or the last member link in the port channel is deleted or goes down.

Load Balancing on Port Channels

GEC interfaces can use either dynamic flow-based load balancing or VLAN-manual load balancing. You can configure the load-balancing method globally for all port channels or directly on specific port channels. The global configuration applies only to those port channels for which you have not explicitly configured load balancing. The port-channel configuration overrides the global configuration.

Flow-based load balancing is enabled by default at the global level. You must explicitly configure VLAN load balancing or the load-balancing method is flow-based.

The table below lists the load-balancing method that is applied to port channels based on the configuration:

Table 98: Flow-Based Load Balancing Configuration Options

Global Configuration	Port-Channel Configuration	Load Balancing Applied
Not configured	Not configured	Flow-based
	Flow-based	Flow-based
	VLAN-manual	VLAN-manual
VLAN-manual	Not configured	VLAN-manual
	Flow-based	Flow-based
	VLAN-manual	VLAN-manual

The table below lists the configuration that results if you change the global load-balancing method.

Table 99: Results When Global Configuration Changes

Port-Channel Configuration	Global Configuration		Action Taken at Port-Channel
—	From	To	—
Not configured	Not configured	VLAN-manual	Changed from flow-based to VLAN-manual
	VLAN-manual	Not configured	Changed from VLAN-manual to flow-based
Configured	Any	Any	No change

The table below lists the configuration that results if you change the port-channel load-balancing method.

Table 100: Results When Port-Channel Configuration Changes

Port-Channel Configuration	Global Configuration		Action Taken at Port-Channel
—	From	To	—
Not configured	Not configured	VLAN-manual	Changed from flow-based to VLAN-manual
	Not configured	Flow-based	No action taken
	VLAN-manual	Flow-based	Changed from VLAN-manual to flow-based
	VLAN-manual	Not configured	Changed from VLAN-manual to flow-based
	Flow-based	VLAN-manual	Changed from flow-based to VLAN-manual
	Flow-based	Not configured	No action taken

Port-Channel Configuration	Global Configuration		Action Taken at Port-Channel
Configured	Not configured	VLAN-manual	No action taken
	Not configured	Flow-based	Changed from VLAN-manual to flow-based
	VLAN-manual	Flow-based	Changed from VLAN-manual to flow-based
	VLAN-manual	Not configured	No action taken
	Flow-based	VLAN-manual	Changed from flow-based to VLAN-manual
	Flow-based	Not configured	Changed from flow-based to VLAN-manual

How to Enable Flow-Based per Port-Channel Load Balancing

Configuring Load Balancing on a Port Channel

To configure load balancing on a port channel, perform the following steps. Repeat these steps for each GEC interface.

Before you begin

If you have already configured your desired load-balancing method globally and want to use that method for all port channels, you need not perform this task. To configure load balancing globally, use the **port-channel load-balancing vlan-manual** command. If you do not configure the global command, flow-based load balancing is applied to all port channels.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface port-channel** *channel-number* **Example:** `Router(config)# interface port-channel 1`	Enters interface configuration mode and defines the interface as a port channel.
Step 4	**load-balancing** { **flow** | **vlan** } **Example:** `Router(config-if)# load-balancing flow`	Applies a load-balancing method to the specific port channel. • If you do not configure this command, the port channel uses the global load-balancing method configured with the **port-channel load-balancing vlan-manual** command. The global default is flow-based.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits configuration mode.

Verifying Load Balancing Configuration on a Ten GEC Interface

- **show running-config interface port-channel** *channel-number* —Displays the port channel configuration.

  Following is a sample output of this command:

```
Router# show running-config interface port-channel 62
Building configuration...

Current configuration : 108 bytes
!
interface Port-channel62
 ip address 12.1.1.1 255.255.255.0
 ipv6 address 2001:12:1:1::1/64
 mpls
```

- **show etherchannel load-balancing** — Displays the load balancing method applied to each port channel.

  The following is a sample output of this command:

```
Router# show etherchannel load-balancing

EtherChannel Load-Balancing Method:
Global LB Method: flow-based

  Port-Channel:                      LB Method
    Port-channel62                :  flow-based
    Port-channel63                :  flow-based
```

- **show interfaces port-channel** *channel-number* **etherchannel** —Displays the bucket distribution currently in use.

The following is a sample output for an interface with load balancing set to flow-based:

```
Router(config)# show interface port-channel 62 etherchannel

 All IDBs List contains 8 configured interfaces
  Port: TenGigabitEthernet4/1/0 (index: 0)
  Port: TenGigabitEthernet4/1/1 (index: 1)
  Port: TenGigabitEthernet4/1/2 (index: 2)
  Port: TenGigabitEthernet4/1/3 (index: 3)
  Port: TenGigabitEthernet4/1/4 (index: 4)
  Port: TenGigabitEthernet4/1/5 (index: 5)
  Port: TenGigabitEthernet4/1/6 (index: 6)
  Port: TenGigabitEthernet4/1/7 (index: 7)

 Active Member List contains 8 interfaces
  Port: TenGigabitEthernet4/1/0
    LACP Mode: Active

  Port: TenGigabitEthernet4/1/1
    LACP Mode: Active

  Port: TenGigabitEthernet4/1/2
    LACP Mode: Active

  Port: TenGigabitEthernet4/1/3
    LACP Mode: Active

  Port: TenGigabitEthernet4/1/4
    LACP Mode: Active

  Port: TenGigabitEthernet4/1/5
    LACP Mode: Active

  Port: TenGigabitEthernet4/1/6
    LACP Mode: Active

  Port: TenGigabitEthernet4/1/7
    LACP Mode: Active

 Passive Member List contains 0 interfaces
 Load-Balancing method applied: flow-based

 Bucket Information for Flow-Based LB:
 Interface:                                Buckets
    TenGigabitEthernet4/1/0:
                           Bucket 0 , Bucket 1
    TenGigabitEthernet4/1/1:
                           Bucket 2 , Bucket 3
    TenGigabitEthernet4/1/2:
                           Bucket 4 , Bucket 5
    TenGigabitEthernet4/1/3:
                           Bucket 6 , Bucket 7
    TenGigabitEthernet4/1/4:
                           Bucket 8 , Bucket 9
    TenGigabitEthernet4/1/5:
                           Bucket 10, Bucket 11
    TenGigabitEthernet4/1/6:
                           Bucket 12, Bucket 13
    TenGigabitEthernet4/1/7:
                           Bucket 14, Bucket 15
```

Configuration Examples for Flow-Based per Port-Channel Load Balancing

Example: Flow-Based Load Balancing

The following example shows a configuration where flow-based load balancing is configured on port-channel 2 while the VLAN-manual method is configured globally:

```
!
no aaa new-model
port-channel load-balancing vlan-manual
ip source-route
.
.
.
interface Port-channel2
 ip address 10.0.0.1 255.255.255.0
 no negotiation auto
 load-balancing flow
!
interface Port-channel2.10
 ip rsvp authentication key 11223344
 ip rsvp authentication
!
interface Port-channel2.50
 encapsulation dot1Q 50
!
interface TenGigabitEthernet4/1/0
 no ip address
 negotiation auto
 cdp enable
 channel-group 2
!
```

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

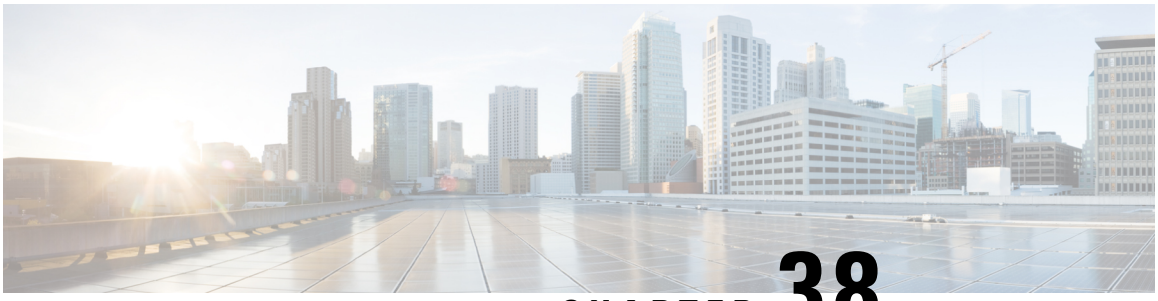
Feature Information for Flow-Based per Port-Channel Load Balancing

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 101: Feature Information for Flow-Based per Port-Channel Load Balancing

Feature Name	Releases	Feature Information
Flow-based per port-channel Load balancing	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on Cisco cBR Series Converged Broadband Router s.



CHAPTER 38

MPLS QoS via TLV for non-L2VPN Service Flow

The MPLS QoS via TLV for non-L2VPN Service Flow feature allows to mark TC bits for MPLS L3VPN imposition packets and classify downstream packets based on TC bits of MPLS disposition packets, using vendor-specific TLVs.

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 613
- Restrictions for MPLS QoS via TLV for non-L2VPN Service Flow, on page 614
- Information About MPLS QoS via TLV for non-L2VPN Service Flow, on page 614
- Configuring MPLS QoS via TLV for non-L2VPN Service Flow, on page 615
- Configuration Examples, on page 616
- Additional References, on page 619
- Feature Information for MPLS QoS via TLV for non-L2VPN Service Flow, on page 620

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 102: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for MPLS QoS via TLV for non-L2VPN Service Flow

- This feature supports only IPv4. It will not support IPv6.
- This feature does not support SNMP.
- This feature does not support dynamic service flows.
- Only up to four VPNs and eight upstream service flows per CM can be configured.
- For a VPN, only a maximum of eight DS classifiers (using TC bits in the range from 0 to 7) can be configured.
- If TC bits downstream classifiers are configured for a VPN, then the downstream MPLS packets belonging to the VPN are processed only on TC bits classification. It will not process general IP header field classification.

Information About MPLS QoS via TLV for non-L2VPN Service Flow

The MPLS QoS via TLV for non-L2VPN Service Flow feature is a QoS enhancement based on MPLS Traffic Class (TC) bits for MPLS L3VPN. The MPLS TC bits were previously known as MPLS EXP bits. RFC 5462 has renamed the MPLS EXP field to MPLS TC field.

For upstream service flow encoding, use Cisco-specific TLV to set TC bits value for MPLS imposition packets. For downstream classifier encoding, use Cisco-specific TLV to implement downstream classification based on TC bits of MPLS disposition packets.

Configuring MPLS QoS via TLV for non-L2VPN Service Flow

Note This feature is configured using a cable modem configuration file and is dependent on the general configuration of the L3VPN.

This section describes how to configure traffic class bits for MPLS imposition and disposition packets and on how to use vendor-specific TLVs with AToM L2VPN and MPLS L3VPN.

Traffic Class for MPLS Imposition Packets

The table lists the vendor-specific TLV to be included in the cable modem configuration file to configure TC bits for MPLS imposition packets. The MPLS-TC-SET TLV is defined in the upstream and is associated with the VPN RD in upstream service flow encoding.

Table 103: TLV to Configure TC Bits for MPLS Imposition Packets

TLV Name	SubType	Length	Value
MPLS-TC-SET TLV	43.5.43.34	1	Imposition MPLS-TC-SET bits

Traffic Classification for MPLS Disposition Packets

The table lists the vendor-specific TLV to be included in the cable modem configuration file to classify DS packets based on TC bits of MPLS disposition packets.

The MPLS-TC-RANGE TLV is defined only under DS classifier encodings. It supports multi-downstream flow in a CM belonging to the same MPLS L3VPN, associated with the VPN RD in downstream classifier encoding.

Table 104: TLV to Classify TC Bits for MPLS Disposition Packets

TLV Name	SubType	Length	Value
MPLS-TC-RANGE	43.5.43.35	2	MPLS-TC-low and MPLS-TC-high

Using Vendor-Specific TLVs with AToM L2VPN and MPLS L3VPN

If both AToM L2VPN (L2 MPLS) and MPLS L3VPN (L3 MPLS) are using the same set of TLVs (MPLS-TC-SET and MPLS-TC-RANGE), then you should differentiate them. Configure the TLVs for upstream service flow encoding and downstream classifier encodings as indicated below:

Upstream Service Flow Encoding

- For L2VPN, configure MPLS-TC-SET (43.5.43.34) and L2VPN ID (43.5.1).
- For MPLS L3VPN, configure MPLS-TC-SET (43.5.43.34) and VPN RD (43.5.1).

Note Do not configure the TLVs for L2VPN and MPLS L3VPN at the same time for upstream service flow encodings, as it will result in a TLV error.

Downstream Classifier Encoding

- L2VPN—Configure MPLS-TC-RANGE (43.5.43.35) and L2VPN ID (43.5.1).
- MPLS L3VPN—Configure MPLS-TC-RANGE (43.5.43.35) and VPN RD (43.5.1).

Configuration Examples

This section provides the following configuration examples:

Example: Upstream Service Flow Marking TLV

The following example shows a sample CM configuration TLV for the provisioning of TC bits for MPLS imposition packets:

```
24 (Upstream Service Flow Encoding)
   S01 (Service Flow Reference)         = 2
   S06 (QoS Parameter Set Type)         = 7
   S43 (Vendor Specific Options)
       T08 (Vendor ID)  = 00 00 0c
       T004 (VPN Route Distinguisher) = xx xx xx xx xx xx xx xx
            S005 (Vendor specific L2VPN TLV)
            S043 (Cisco Vendor Specific)
       T034 (MPLS-TC-SET) = 04 # MPLSTC-SET = 4
```

Example: Downstream Packet Classification TLV

The following example shows a sample CM configuration TLV for classifying downstream packets based on TC bits of MPLS disposition packets:

```
23 (Downstream Packet Classification Encoding)
   S01 (Classifier Reference)           = 13
   S03 (Service Flow Reference)         = 13
   S11 (IEEE 802.1P/Q Packet Classification Encodings)
   S43 (Vendor Specific Options)
       T08 (Vendor ID)  = 00 00 0c
       S004 (VPN Route Distinguisher) = xx xx xx xx xx xx xx xx
       S005 (Vendor specific L2VPN TLV)
           S043 (Cisco Vendor Specific)
           S035 (MPLS-TC-RANGE)   = 04  05      # MPLSTC-EGRESS_RANGE= 4 - 5
```

Example: MPLS QoS Configuration File

The following example shows a cable modem being configured to mark TC bits for MPLS L3VPN imposition packets and classify downstream packets based on TC bits of MPLS L3VPN disposition packets, using vendor-specific TLVs:

```
CM-CONFIG
=========
03 (Net Access Control)                      = 1
18 (Maximum Number of CPE)       = 16
22 (Upstream Packet Classification Encoding Block)
   S01 (Classifier Reference)              = 2
   S03 (Service Flow Reference)        = 2
   S05 (Rule Priority)                             = 2
   S09 (IP Packet Encodings)
       T01 (IP Type of Srv Rng & Mask)             = 00 20 ff
22 (Upstream Packet Classification Encoding Block)
   S01 (Classifier Reference)              = 3
   S03 (Service Flow Reference)        = 3
   S05 (Rule Priority)                             = 3
   S09 (IP Packet Encodings)
       T01 (IP Type of Srv Rng & Mask)             = 40 80 ff
22 (Upstream Packet Classification Encoding Block)
   S01 (Classifier Reference)              = 4
   S03 (Service Flow Reference)        = 4
   S05 (Rule Priority)                             = 4
   S09 (IP Packet Encodings)
       T01 (IP Type of Srv Rng & Mask)             = a0 e0 ff
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference)              = 12
   S03 (Service Flow Reference)        = 12
   S05 (Rule Priority)                             = 2
   S09 (IP Packet Encodings)
       T01 (IP Type of Srv Rng & Mask)             = 00 ff ff
   S43 (Vendor Specific Options)
       T08 (Vendor ID)           = 00 00 0c
       T004 (Unknown sub-type)         = 00 00 00 01 00 00 00 01
       T005 (Unknown sub-type)         = 2b 09 08 03 00 00 0c 23 02 01 01
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference)              = 13
   S03 (Service Flow Reference)        = 13
   S05 (Rule Priority)                             = 3
   S09 (IP Packet Encodings)
       T01 (IP Type of Srv Rng & Mask)             = 00 ff ff
   S43 (Vendor Specific Options)
       T08 (Vendor ID)           = 00 00 0c
       T004 (Unknown sub-type)         = 00 00 00 01 00 00 00 01
       T005 (Unknown sub-type)         = 2b 09 08 03 00 00 0c 23 02 02 02
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference)              = 14
   S03 (Service Flow Reference)        = 14
   S05 (Rule Priority)                             = 4
   S09 (IP Packet Encodings)
       T01 (IP Type of Srv Rng & Mask)             = 00 ff ff
   S43 (Vendor Specific Options)
       T08 (Vendor ID)           = 00 00 0c
       T004 (Unknown sub-type)         = 00 00 00 01 00 00 00 01
       T005 (Unknown sub-type)         = 2b 09 08 03 00 00 0c 23 02 03 03
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference)        = 1
   S06 (QoS Parameter Set Type)     = 7
24 (Upstream Service Flow Encodings)
```

```
   S01 (Service Flow Reference)        = 2
   S06 (QoS Parameter Set Type)     = 7
   S43 (Vendor Specific Options)
       T08 (Vendor ID)           = 00 00 0c
       T004 (Unknown sub-type)         = 00 00 00 01 00 00 00 01
       T005 (Unknown sub-type)         = 2b 08 08 03 00 00 0c 22 01 04
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference)        = 3
   S06 (QoS Parameter Set Type)     = 7
   S43 (Vendor Specific Options)
       T08 (Vendor ID)           = 00 00 0c
       T004 (Unknown sub-type)         = 00 00 00 01 00 00 00 01
       T005 (Unknown sub-type)         = 2b 08 08 03 00 00 0c 22 01 05
24 (Upstream Service Flow Encodings)
   S01 (Service Flow Reference)        = 4
   S06 (QoS Parameter Set Type)     = 7
   S43 (Vendor Specific Options)
       T08 (Vendor ID)           = 00 00 0c
       T004 (Unknown sub-type)         = 00 00 00 01 00 00 00 01
       T005 (Unknown sub-type)         = 2b 08 08 03 00 00 0c 22 01 06
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference)        = 11
   S06 (QoS Parameter Set Type)     = 7
   S07 (Traffic Priority)                        = 7
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference)        = 12
   S06 (QoS Parameter Set Type)     = 7
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference)        = 13
   S06 (QoS Parameter Set Type)     = 7
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference)        = 14
   S06 (QoS Parameter Set Type)     = 7
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference)        = 15
   S06 (QoS Parameter Set Type)     = 7
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference)        = 16
   S06 (QoS Parameter Set Type)     = 7
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference)        = 17
   S06 (QoS Parameter Set Type)     = 7
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference)        = 18
   S06 (QoS Parameter Set Type)     = 7
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference)              = 19
   S03 (Service Flow Reference)        = 19
   S09 (IP Packet Encodings)
       T01 (IP Type of Srv Rng & Mask)            = 00 ff ff
   S43 (Vendor Specific Options)
       T08 (Vendor ID)           = 00 00 0c
       T004 (Unknown sub-type)         = 00 00 00 01 00 00 00 01
       T005 (Unknown sub-type)         = 2b 09 08 03 00 00 0c 23 02 00 00
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference)              = 15
   S03 (Service Flow Reference)        = 15
   S05 (Rule Priority)                           = 3
   S09 (IP Packet Encodings)
       T01 (IP Type of Srv Rng & Mask)            = 00 ff ff
   S43 (Vendor Specific Options)
       T08 (Vendor ID)           = 00 00 0c
       T004 (Unknown sub-type)         = 00 00 00 01 00 00 00 01
       T005 (Unknown sub-type)         = 2b 09 08 03 00 00 0c 23 02 04 04
```

```
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference)              = 16
   S03 (Service Flow Reference)         = 16
   S05 (Rule Priority)                             = 3
   S09 (IP Packet Encodings)
       T01 (IP Type of Srv Rng & Mask)             = 00 ff ff
   S43 (Vendor Specific Options)
       T08 (Vendor ID)            = 00 00 0c
       T004 (Unknown sub-type)          = 00 00 00 01 00 00 00 01
       T005 (Unknown sub-type)          = 2b 09 08 03 00 00 0c 23 02 05 05
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference)               = 17
   S03 (Service Flow Reference)         = 17
   S05 (Rule Priority)                             = 3
   S09 (IP Packet Encodings)
       T01 (IP Type of Srv Rng & Mask)             = 00 ff ff
   S43 (Vendor Specific Options)
       T08 (Vendor ID)            = 00 00 0c
       T004 (Unknown sub-type)          = 00 00 00 01 00 00 00 01
       T005 (Unknown sub-type)          = 2b 09 08 03 00 00 0c 23 02 06 06
23 (Downstream Packet Classification Encoding Block)
   S01 (Classifier Reference)               = 18
   S03 (Service Flow Reference)         = 18
   S09 (IP Packet Encodings)
       T01 (IP Type of Srv Rng & Mask)             = 00 ff ff
   S43 (Vendor Specific Options)
       T08 (Vendor ID)            = 00 00 0c
       T004 (Unknown sub-type)          = 00 00 00 01 00 00 00 01
       T005 (Unknown sub-type)          = 2b 09 08 03 00 00 0c 23 02 07 07
25 (Downstream Service Flow Encodings)
   S01 (Service Flow Reference)         = 19
   S06 (QoS Parameter Set Type)     = 7
#<EOF>
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for MPLS QoS via TLV for non-L2VPN Service Flow

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

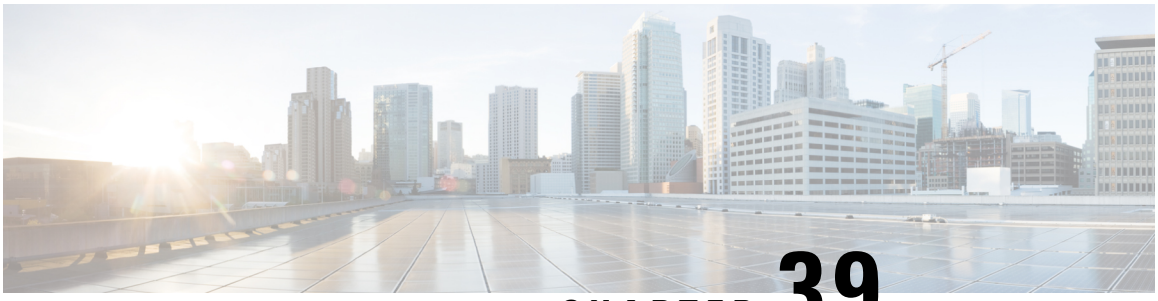
Table 105: Feature Information for MPLS QoS via TLV for non-L2VPN Service Flow

Feature Name	Releases	Feature Information
MPLS QoS via TLV for non-L2VPN Service Flow	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.

PART VI

Layer 3 Configuration

- DHCP, ToD, and TFTP Services for CMTS Routers, on page 623
- Virtual Interface Bundling, on page 643
- IPv6 on Cable, on page 653
- Cable DHCP Leasequery, on page 695
- Layer 3 CPE Mobility, on page 705
- DOCSIS 3.0 Multicast Support, on page 713
- IPv6 Segment Routing on Cisco cBR, on page 733



CHAPTER 39

DHCP, ToD, and TFTP Services for CMTS Routers

Note Cisco IOS-XE Release 16.5.1 integrates support for this feature on Cisco CMTS routers.

This document describes how to configure Cisco Cable Modem Termination System (CMTS) platforms so that they support onboard servers that provide Dynamic Host Configuration Protocol (DHCP), Time-of-Day (ToD), and Trivial File Transfer Protocol (TFTP) services for use in Data-over-Cable Service Interface Specification (DOCSIS) networks. In addition, this document provides information about optional configurations that can be used with external DHCP servers.

- Prerequisites for DHCP, ToD, and TFTP Services, on page 623
- Restrictions for DHCP, ToD, and TFTP Services, on page 623
- Information About DHCP, ToD, and TFTP Services, on page 624
- How to Configure ToD, and TFTP Services, on page 629
- How to Configure ToD, and TFTP Services, on page 641
- Configuration Examples, on page 641
- Additional References, on page 642
- Feature Information for the DHCP, ToD, and TFTP Services for the CMTS Routers, on page 642

Prerequisites for DHCP, ToD, and TFTP Services

To use the Cisco CMTS as the ToD server, either standalone or with other external ToD servers, you must configure the DHCP server to provide the IP address of the Cisco CMTS as one of the valid ToD servers (DHCP option 4) for cable modems.

Restrictions for DHCP, ToD, and TFTP Services

- The ToD server must use the UDP protocol to conform to DOCSIS specifications.
- For proper operation of the DOCSIS network, especially a DOCSIS 1.1 network using BPI+ encryption and authentication, the system clock on the Cisco CMTS must be set accurately. You can achieve this by manually using the **set clock** command, or by configuring the CMTS to use either the Network Time Protocol (NTP) or the Simple Network Time Protocol (SNTP).
- Cisco cBR series routers do not support internal DHCP servers.

Information About DHCP, ToD, and TFTP Services

This section provides the following information about the DHCP, ToD, and TFTP Services feature, and its individual components:

Feature Overview

All Cisco CMTS platforms support onboard servers that provide DHCP, ToD, and TFTP proxy-services for use in DOCSIS cable networks. These servers provide the registration services needed by DOCSIS 1.0- and 1.1-compliant cable modems:

- External DHCP Servers—Provides DHCP services. External DHCP servers are usually part of an integrated provisioning system that is more suitable when managing large cable networks.
- Time-of-DayServer_—Provides an RFC 868 -compliant ToD service so that cable modems can obtain the current date and time during the registration process. The cable modem connects with the ToD server after it has obtained its IP address and other DHCP-provided IP parameters.

Although cable modems do not need to successfully complete the ToD request before coming online, this allows them to add accurate timestamps to their event logs so that these logs are coordinated to the clock used on the CMTS. In addition, having the accurate date and time is essential if the cable modem is trying to register with Baseline Privacy Interface Plus (BPI+) encryption and authentication.

- External TFTP_Server—Downloads the DOCSIS configuration file to the cable modem. The DOCSIS configuration file contains the operational parameters for the cable modem. The cable modem downloads its DOCSIS configuration file after connecting with the ToD server.

Note You can add additional servers in a number of ways. For example, most cable operators use Cisco Network Registrar (CNR) to provide the DHCP and TFTP servers. ToD servers are freely available for most workstations and PCs. You can install the additional servers on one workstation or PC or on different workstations and PCs.

External DHCP Servers

The Cisco CMTS router provides the following optional configurations that can enhance the operation and security of external DHCP servers that you are using on the DOCSIS cable network:

Cable Source Verify Feature

To combat theft-of-service attacks, you can enable the **cable source-verify** command on the cable interfaces on the Cisco CMTS router. This feature uses the router's internal database to verify the validity of the IP packets that the CMTS receives on the cable interfaces, and provides three levels of protection:

- At the most basic level of protection, the Cable Source Verify feature examines every IP upstream packet to prevent duplicate IP addresses from appearing on the cable network. If a conflict occurs, the Cisco CMTS recognizes only packets coming from the device that was assigned the IP address by the DHCP server. The devices with the duplicate addresses are not allowed network address. The CMTS also refuses to recognize traffic from devices with IP addresses that have network addresses that are unauthorized for that particular cable segment.

- Adding the **dhcp** option to the **cable source-verify** command provides a more comprehensive level of protection by preventing users from statically assigning currently-unused IP addresses to their devices. When the Cisco CMTS receives a packet with an unknown IP address on a cable interface, the CMTS drops the packet but also issues a DHCP LEASEQUERY message that queries the DHCP servers for any information about the IP and MAC addresses of that device. If the DHCP servers do not return any information about the device, the CMTS continues to block the network access for that device.
- When you use the **dhcp** option, you can also enable the **leasetimer** option, which instructs the Cisco CMTS to periodically check its internal CPE database for IP addresses whose lease times have expired. The CPE devices that are using expired IP addresses are denied further access to the network until they renew their IP addresses from a valid DHCP server. This can prevent users from taking DHCP-assigned IP addresses and assigning them as static addresses to their CPE devices.
- In addition to the dhcp option, you can also configure prefix-based source address verification (SAV) on the Cisco CMTS using the cable source-verify group command. A CM may have a static IPv4 or IPv6 prefix configured, which belongs to an SAV group. When the SAV prefix processing is enabled on the Cisco CMTS, the source IP address of the packets coming from the CM is matched against the configured prefix and SAV group (for that CM) for verification. If the verification fails, the packets are dropped, else the packets are forwarded for further processing. For more information on SAV prefix processing and SAV prefix configuration, see Prefix-based Source Address Verification , on page 625 and Configuring Prefix-based Source Address Verification, on page 636

Prefix-based Source Address Verification

The Source Address Verification (SAV) feature verifies the source IP address of an upstream packet to ensure that the SID/MAC and IP are consistent. The DOCSIS 3.0 Security Specification introduces prefix-based SAV where every CM may have static IPv4 or IPv6 prefixes configured. These prefixes are either preconfigured on the CMTS, or are communicated to the CMTS during CM registration. The Cisco CMTS uses these configured prefixes to verify the source IP address of all the incoming packets from that CM.

An SAV group is a collection of prefixes. A prefix is an IPv4 or IPv6 subnet address. You can use the cable source-verify group command in global configuration mode to configure SAV groups. A total of 255 SAV groups are supported on a CMTS, with each SAV group having a maximum of four prefixes. Prefixes can be configured using the prefix command.

During registration, CMs communicate their configured static prefixes to the CMTS using two TLVs, 43.7.1 and 43.7.2. The TLV 43.7.1 specifies the SAV prefix group name that the CM belongs to, and TLV 43.7.2 specifies the actual IPv4 or IPv6 prefix. Each CM can have a maximum of four prefixes configured. When the Cisco CMTS receives these TLVs, it first identifies if the specified SAV group and the prefixes are already configured on the Cisco CMTS. If they are configured, the Cisco CMTS associates them to the registering CM. However if they are not configured, the Cisco CMTS automatically creates the specified SAV group and prefixes before associating them to the registering CM.

The SAV group name and the prefixes that are provided by these TLVs are considered valid by the Cisco CMTS. The packets received (from the CM) with the source IP address belonging to the prefix specified by the TLV are considered authorized. For example, if a given CM has been configured with an SAV prefix of 10.10.10.0/24, then any packet received from this CM (or CPE behind the CM) that is sourced with this address in the subnet 10.10.10.0/24 is considered to be authorized.

For more information on how to configure SAV groups and prefixes see Configuring Prefix-based Source Address Verification, on page 636.

Smart Relay Feature

The Cisco CMTS supports a Smart Relay feature (the **ip dhcp smart-relay** command), which automatically switches a cable modem or CPE device to secondary DHCP servers or address pools if the primary server

runs out of IP addresses or otherwise fails to respond with an IP address. The relay agent attempts to forward DHCP requests to the primary server three times. After three attempts with no successful response from the primary, the relay agent automatically switches to the secondary server.

When you are using the **cable dhcp-giaddr policy** command to specify that the CPE devices should use the secondary DHCP pools corresponding to the secondary addresses on a cable interface, the smart relay agent automatically rotates through the available secondary in a round robin fashion until an available pool of addresses is found. This ensures that clients are not locked out of the network because a particular pool has been exhausted.

GIADDR Field

When using separate IP address pools for cable modems and CPE devices, you can use the **cable dhcp-giaddr policy** command to specify that cable modems should use an address from the primary pool and that CPE devices should use addresses from the secondary pool. The default is for the CMTS to send all DHCP requests to the primary DHCP server, while the secondary servers are used only if the primary server does not respond. The different DHCP servers are specified using the **cable helper** commands.

DHCP Relay Agent Sub-option

The DHCP Relay Agent Information sub-option (DHCP Option 82, Suboption 9) enhancement simplifies provisioning of the CPE devices. Using this sub-option, the cable operators can relay the service class or QoS information of the CPE to the DHCP server to get an appropriate IP address.

To provision a CPE, the DHCP server should be made aware of the service class or QoS information of the CPE. The DHCP server obtains this information using the DHCP DISCOVER message, which includes the service class or QoS information of the CM behind which the CPE resides.

During the provisioning process, the Cisco CMTS uses the DHCPv4 Relay Agent Information sub-option to advertise information about the service class or QoS profile of the CMs to the DHCP server. Using the same technique, the CPE information is relayed to the DHCP server to get an appropriate IP address.

To enable the service classes option, the service class name specified in the CM configuration file must be configured on the Cisco CMTS. This is done by using the **cable dhcp-insert service-class** command.

Note To insert service class relay agent information option into the DHCP DISCOVER messages, the **ip dhcp relay information option-insert** command must be configured on the bundle interface.

Time-of-Day Server

The Cisco CMTS can function as a ToD server that provides the current date and time to the cable modems and other customer premises equipment (CPE) devices connected to its cable interfaces. This allows the cable modems and CPE devices to accurately timestamp their Simple Network Management Protocol (SNMP) messages and error log entries, as well as ensure that all of the system clocks on the cable network are synchronized to the same system time.

The DOCSIS 1.0 and 1.1 specifications require that all DOCSIS cable modems request the following time-related fields in the DHCP request they send during their initial power-on provisioning:

- Time Offset (option 2)—Specifies the time zone for the cable modem or CPE device, in the form of the number of seconds that the device's timestamp is offset from Greenwich Mean Time (GMT).

- Time Server Option (option 4)—Specifies one or more IP addresses for a ToD server.

After a cable modem successfully acquires a DHCP lease time, it then attempts to contact one of the ToD servers provided in the list provided by the DHCP server. If successful, the cable modem updates its system clock with the time offset and timestamp received from the ToD server.

If a ToD server cannot be reached or if it does not respond, the cable modem eventually times out, logs the failure with the CMTS, and continues on with the initialization process. The cable modem can come online without receiving a reply from a ToD server, but it must periodically continue to reach the ToD server at least once in every five-minute period until it successfully receives a ToD reply. Until it reaches a ToD server, the cable modem must initialize its system clock to midnight on January 1, 1970 GMT.

Note Initial versions of the DOCSIS 1.0 specification specified that the cable device must obtain a valid response from a ToD server before continuing with the initialization process. This requirement was removed in the released DOCSIS 1.0 specification and in the DOCSIS 1.1 specifications. Cable devices running older firmware that is compliant with the initial DOCSIS 1.0 specification, however, might require receiving a reply from a ToD server before being able to come online.

Because cable modems will repeatedly retry connecting with a ToD server until they receive a successful reply, you should consider activating the ToD server on the Cisco CMTS, even if you have one or more other ToD servers at the headend. This ensures that an online cable modem will always be able to connect with the ToD server on the Cisco CMTS, even if the other servers go down or are unreachable because of network congestion, and therefore will not send repeated ToD requests.

Tip To be able to use the Cisco CMTS as the ToD server, you must configure the DHCP server to provide the IP address Cisco CMTS as one of the valid ToD servers (DHCP option 4) for cable modems.

In addition, although the DOCSIS specifications do not require that a cable modem successfully obtain a response from a ToD server before coming online, not obtaining a timestamp could prevent the cable modem from coming online in the following situations:

- If DOCSIS configuration files are being timestamped, to prevent cable modems from caching the files and replaying them, the clocks on the cable modem and CMTS must be synchronized. Otherwise, the cable modem cannot determine whether a DOCSIS configuration file has the proper timestamp.
- If cable modems register using Baseline Privacy Interface Plus (BPI+) authentication and encryption, the clocks on the cable modem and CMTS must be synchronized. This is because BPI+ authorization requires that the CMTS and cable modem verify the timestamps on the digital certificates being used for authentication. If the timestamps on the CMTS and cable modem are not synchronized, the cable modem cannot come online using BPI+ encryption.

Note DOCSIS cable modems must use RFC 868 -compliant ToD server to obtain the current system time. They cannot use the Network Time Protocol (NTP) or Simple Network Time Protocol (SNTP) service for this purpose. However, the Cisco CMTS can use an NTP or SNTP server to set its own system clock, which can then be used by the ToD server. Otherwise, you must manually set the clock on the CMTS using the **clock set** command each time that the CMTS boots up.

Tip Additional servers can be provided by workstations or PCs installed at the cable headend. UNIX and Solaris systems typically include a ToD server as part of the operating system, which can be enabled by putting the appropriate line in the inetd.conf file. Windows systems can use shareware servers such as Greyware and Tardis. The DOCSIS specifications require that the ToD servers use the User Datagram Protocol (UDP) protocol instead of the TCP protocol for its packets.

TFTP Server

All Cisco CMTS platforms can be configured to provide a TFTP server that can provide the following types of files to DOCSIS cable modems:

- DOCSIS Configuration File—After a DOCSIS cable modem has acquired a DHCP lease and attempted to contact a ToD server, the cable modem uses TFTP to download a DOCSIS configuration file from an authorized TFTP server. The DHCP server is responsible for providing the name of the DOCSIS configuration file and IP address of the TFTP server to the cable modem.
- Software Upgrade File—If the DOCSIS configuration file specifies that the cable modem must be running a specific version of software, and the cable modem is not already running that software, the cable modem must download that software file. For security, the cable operator can use different TFTP servers for downloading DOCSIS configuration files and for downloading new software files.
- Cisco IOS-XE Configuration File—The DOCSIS configuration file for Cisco cable devices can also specify that the cable modem should download a Cisco IOS-XE configuration file that contains command-line interface (CLI) configuration commands. Typically this is done to configure platform-specific features such as voice ports or IPSec encryption.

Note Do not confuse the DOCSIS configuration file with the Cisco IOS-XE configuration file. The DOCSIS configuration file is a binary file in the particular format that is specified by the DOCSIS specifications, and each DOCSIS cable modem must download a valid file before coming online. In contrast, the Cisco IOS-XE configuration file is an ASCII text file that contains one or more Cisco IOS-XE CLI configuration commands. Only Cisco cable devices can download a Cisco IOS-XE file.

All Cisco CMTS platforms can be configured as TFTP servers that can upload these files to the cable modem. The files can reside on any valid device but typically should be copied to the Flash memory device inserted into the Flash disk slot on the Cisco CMTS.

Benefits

- The Cisco CMTS can act as a primary or backup ToD server to ensure that all cable modems are synchronized with the proper date and time before coming online. This also enables cable modems to come online more quickly because they will not have to wait for the ToD timeout period before coming online.
- The ToD server on the Cisco CMTS ensures that all devices connected to the cable network are using the same system clock, making it easier for you to troubleshoot system problems when you analyze the debugging output and error logs generated by many cable modems, CPE devices, the Cisco CMTS, and other services.

- The Cisco CMTS can act as a TFTP server for DOCSIS configuration files, software upgrade files, and Cisco IOS configuration files.

How to Configure ToD, and TFTP Services

See the following configuration tasks required to configure time-of-day service, and TFTP service on a Cisco CMTS:

Configuring Time-of-Day Service

This section provides procedures for enabling and disabling the time-of-day (ToD) server on the Cisco CMTS routers.

Prerequisites

To be able to use the Cisco CMTS as the ToD server you must configure the DHCP server to provide the IP address Cisco CMTS as one of the valid ToD servers (DHCP option 4) for cable modems.

Enabling Time-of-Day Service

To enable the ToD server on a Cisco CMTS, use the following procedure, beginning in EXEC mode.

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**service udp-small-servers max-servers no-limit**
	Example:
	`Router(config)# service udp-small-servers max-servers no-limit`
	`Router(config)#`
		Enables use of minor servers that use the UDP protocol (such as ToD, echo, chargen, and discard).
		The max-servers no-limit option allows a large number of cable modems to obtain the ToD server at one time, in the event that a cable or power failure forces many cable modems offline. When the problem has been resolved, the cable modems can quickly reconnect.

	Command or Action	Purpose

Step 4	**cable time-server**
	Example:
	`Router(config)# cable time-server`
	`Router(config)#`
		Enables the ToD server on the Cisco CMTS.

Step 5	**exit**
	Example:
	`Router(config)# exit`
	`Router#`
		Exits global configuration mode.

Disabling Time-of-Day Service

To disable the ToD server, use the following procedure, beginning in EXEC mode.

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**no cable time-server**
	Example:
	`Router(config)# cable time-server`
	`Router(config)#`
		Disables the ToD server on the Cisco CMTS.

Step 4	**no service udp-small-servers**
	Example:
	`Router(config)# no service udp-small-servers`
	`Router(config)#`
		(Optional) Disables the use of all minor UDP servers.
		Note Do not disable the minor UDP servers if you are also enabling the other DHCP or TFTP servers.

Step 5	**exit** **Example:**	Exits global configuration mode.

	Command or Action	Purpose
	`Router(config)# exit` `Router#`	

Configuring TFTP Service

To configure TFTP service on a Cisco CMTS where the CMTS can act as a TFTP server and download a DOCSIS configuration file to cable modems, perform the following steps:

- Create the DOCSIS configuration files using the DOCSIS configuration editor of your choice.
- Copy all desired files (DOCSIS configuration files, software upgrade files, and Cisco IOS configuration files) to the Flash memory device on the Cisco CMTS. Typically, this is done by placing the files first on an external TFTP server, and then using TFTP commands to transfer them to the router's Flash memory.
- Enable the TFTP server on the Cisco CMTS with the **tftp-server** command.

Each configuration task is required unless otherwise listed as optional.

Procedure

Step 1 Use the **show file systems** command to display the Flash memory cards that are available on your CMTS, along with the free space on each card and the appropriate device names to use to access each card.

Most configurations of the Cisco CMTS platforms support both linear Flash and Flash disk memory cards. Linear Flash memory is accessed using the **slot0** (or **flash**) and **slot1** device names. Flash disk memory is accessed using the **disk0** and **disk1** device names.

For example, the following command shows a Cisco uBR7200 series router that has two linear Flash memory cards installed. The cards can be accessed by the **slot0** (or **flash**) and **slot1** device names.

Example:

```
Router# show file systems

File Systems:
     Size(b)     Free(b)      Type  Flags  Prefixes
    48755200    48747008     flash     rw   slot0: flash:
    16384000    14284000     flash     rw   slot1:
    32768000    31232884     flash     rw   bootflash:
*          -           -      disk     rw   disk0:
           -           -      disk     rw   disk1:
           -           -    opaque     rw   system:
           -           -    opaque     rw   null:
           -           -   network     rw   tftp:
      522232      507263     nvram     rw   nvram:
           -           -   network     rw   rcp:
           -           -   network     rw   ftp:
           -           -   network     rw   scp:
Router#
```

The following example shows a Cisco uBR10012 router that has two Flash disk cards installed. These cards can be accessed by the **disk0** and **sec-disk0** device names.

Example:

```
Router# show file systems

File Systems:
     Size(b)     Free(b)      Type  Flags  Prefixes
           -           -     flash     rw   slot0: flash:
           -           -     flash     rw   slot1:
    32768000    29630876     flash     rw   bootflash:
*  128094208    95346688      disk     rw   disk0:
           -           -      disk     rw   disk1:
           -           -    opaque     rw   system:
           -           -     flash     rw   sec-slot0:
           -           -     flash     rw   sec-slot1:
*  128094208    95346688      disk     rw   sec-disk0:
           -           -      disk     rw   sec-disk1:
    32768000    29630876     flash     rw   sec-bootflash:
           -           -     nvram     rw   sec-nvram:
           -           -    opaque     rw   null:
           -           -   network     rw   tftp:
      522232      505523     nvram     rw   nvram:
           -           -   network     rw   rcp:
           -           -   network     rw   ftp:
           -           -   network     rw   scp:
Router#
```

Step 2 Verify that the desired Flash memory card has sufficient free space for all of the files that you want to copy to the CMTS.

Step 3 Use the **ping** command to verify that the remote TFTP server that contains the desired files is reachable. For example, the following shows a **ping** command being given to an external TFTP server with the IP address of 10.10.10.1:

Example:

```
Router# ping 10.10.10.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 5/6/6 ms
```

Step 4 Use the **copy tftp** *devname* command to copy each file from the external TFTP server to the appropriate Flash memory card on the CMTS, where *devname* is the device name for the destination Flash memory card. You will then be prompted for the IP address for the external TFTP server and the filename for the file to be transferred.

The following example shows the file docsis.cm being transferred from the external TFTP server at IP address 10.10.10.1 to the first Flash memory disk (disk0):

Example:

```
Router# copy tftp disk0
Address or name of remote host []? 10.10.10.1

Source filename []? config-files/docsis.cm

Destination filename [docsis.cm]?
Accessing tftp://10.10.10.1/config-file/docsis.cm......
Loading docsis.cm from 10.10.10.1 (via Ethernet2/0): !!!
[OK - 276/4096 bytes]
```

```
276 bytes copied in 0.152 secs
Router#
```

Step 5 Repeat Step 4, on page 632 as needed to copy all of the files from the external TFTP server to the Flash memory card on the Cisco CMTS.

Step 6 Use the **dir** command to verify that the Flash memory card contains all of the transferred files.

Example:

```
Router# dir disk0:

Directory of disk0:/
    1  -rw-    10705784   May 30 2002 19:12:46  ubr10k-p6-mz.122-2.8.BC
    2  -rw-        4772   Jun 20 2002 18:12:56  running.cfg.save
    3  -rw-         241   Jul 31 2002 18:25:46  gold.cm
    4  -rw-         225   Jul 31 2002 18:25:46  silver.cm
    5  -rw-         231   Jul 31 2002 18:25:46  bronze.cm
    6  -rw-          74   Oct 11 2002 21:41:14  disable.cm
    7  -rw-     2934028   May 30 2002 11:22:12  ubr924-k8y5-mz.bin
    8  -rw-     3255196   Jun 28 2002 13:53:14  ubr925-k9v9y5-mz.bin
128094208 bytes total (114346688 bytes free)
Router#
```

Step 7 Use the **configure terminal** command to enter global configuration mode:

Example:

```
Router# configure terminal

Router(config)#
```

Step 8 Use the **tftp-server** command to specify which particular files can be transferred by the TFTP server that is onboard the Cisco CMTS. You can also use the **alias** option to specify a different filename that the DHCP server can use to refer to the file. For example, the following commands enable the TFTP transfer of the configuration files and software upgrade files:

Example:

```
Router(config)# tftp-server disk0:gold.cm alias gold.cm

Router(config)# tftp-server disk0:silver.cm alias silver.cm

Router(config)# tftp-server disk0:bronze.cm alias bronze.cm

Router(config)# tftp-server disk0:ubr924-k8y5-mz.bin alias ubr924-codefile

Router(config)# tftp-server disk0:ubr925-k9v9y5-mz.bin alias ubr925-codefile

Router(config)#
```

Note The **tftp-server** command also supports the option of specifying an access list that restricts access to the particular file to the IP addresses that match the access list.

Step 9 (Optional) Use the following command to enable the use of the UDP small servers, and to allow an unlimited number of connections at one time. This will allow a large number of cable modems that have gone offline due to cable or power failure to rapidly come back online.

Example:

```
Router(config)# service udp-small-servers max-servers no-limit
```

```
Router(config)#
```

Optimizing the Use of an External DHCP Server

The Cisco CMTS offers a number of options that can optimize the operation of external DHCP servers on a DOCSIS cable network. See the following sections for details. All procedures are optional, depending on the needs of your network and application servers.

Configuring Cable Source Verify Option

To enhance security when using external DHCP servers, you can optionally configure the Cable Source Verify feature with the following procedure.

Restriction

- The Cable Source Verify feature supports only external DHCP servers. It cannot be used with the internal DHCP server.

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**interface cable** *x/y*
	Example:
	`Router(config)# interface cable 4/0`
	`Router(config-if)#`
		Enters cable interface configuration mode for the specified cable interface.

Step 4	**cable source-verify** [**dhcp** | **leasetimer** *value*] **Example:** `Router(config-if)# cable source-verify dhcp` **Example:**	(Optional) Ensures that the CMTS allows network access only to those IP addresses that DCHP servers issued to devices on this cable interface. The CMTS examines DHCP packets that pass through the cable interfaces to build a database of which IP addresses are valid on which interface.

	Command or Action	Purpose
	`Router(config-if)# cable source-verify leasetimer 30` `Router(config-if)#`	• **dhcp** = (Optional) Drops traffic from all devices with unknown IP addresses, but the CMTS also sends a query to the DHCP servers for any information about the device. If a DHCP server informs the CMTS that the device has a valid IP address, the CMTS then allows the device on the network. • **leasetimer** *value* = (Optional) Specifies how often, in minutes, the router should check its internal CPE database for IP addresses whose lease times have expired. This can prevent users from taking DHCP-assigned IP addresses and assigning them as static addresses to their CPE devices. The valid range for value is 1 to 240 minutes, with no default. **Note** The **leasetimer** option takes effect only when the **dhcp** option is also used on an interface.

Step 5	**no cable arp**
	Example:
	`Router(config-if)# no cable arp`
	`Router(config-if)#`
		(Optional) Blocks Address Resolution Protocol (ARP) requests originating from devices on the cable network. Use this command, together with the **cable source-verify dhcp** command, to block certain types of theft-of-service attacks that attempt to hijack or spoof IP addresses.
		Note Repeat Step 3, on page 634 through Step 5, on page 635 for each desired cable interface.

Step 6	**exit**
	Example:
	`Router(config-if)# exit`
	`Router(config)#`
		Exits interface configuration mode.

Step 7	**ip dhcp relay information option**
	Example:
	`Router(config)# ip dhcp relay information option`
	`Router(config)#`
		(Optional) Enables the CMTS to insert DHCP relay information (DHCP option 82) in relayed DHCP packets. This allows the DHCP server to store accurate information about which CPE devices are using which cable modems. You should use this command if you are also using the **cable source-verify dhcp** command.

Step 8	**exit** **Example:**	Exits global configuration mode.

	Command or Action	Purpose
	`Router(config)# exit` `Router#`	

Configuring Prefix-based Source Address Verification

To enhance security when using external DHCP servers, you can configure a prefix-based SAV with the following procedure, beginning in global configuration (config) mode.

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	cable source-verify enable-sav-static
	Example:
	`Router# cable source-verify enable-sav-static`
	`Router(config)#`
		Enables SAV prefix processing on the Cisco CMTS.

Step 4	**cable source-verify group** *groupname* **Example:** `Router(config)# cable source-verify group sav-1`	Configures the SAV group name. *groupname*— Name of the SAV group with a maximum length of 16 characters.

Step 5	**prefix** [**ipv4_prefix/ipv4_prefix_length** | **ipv6_prefix/ipv6_prefix_length**]
	Example:
	`Router(config-sav)# prefix 10.10.10.0/24`
	`Router(config-sav)#`
		Configures the IPv4 or IPv6 prefix associated with the SAV group.
		• *ipv4_prefix*— IPv4 prefix associated with the SAV group, specified in the X.X.X.X/X format.
		• ipv4_prefix_length—Length of the IPv4 prefix. The valid range is from 0 to 32.
		• ipv6_prefix—IPv6 prefix associated with a particular SAV group, specified in the X:X:X:X::/X format.

	Command or Action	Purpose
		• ipv6_prefix_length—Length of the IPv6 prefix. The valid range is from 0 to 128. A maximum of four prefixes can be configured in a single SAV group. These prefixes can be either IPv4s, IPv6s, or a combination of both.
Step 6	**exit** **Example:** `Router(config-sav)# exit`	Exits SAV configuration mode.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Configuring Optional DHCP Parameters

When using an external DHCP server, the Cisco CMTS supports a number of options that can enhance operation of the cable network in certain applications. To configure these options, use the following procedure, beginning in EXEC mode.

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**ip dhcp smart-relay**
	Example:
	`Router(config)# ip dhcp smart-relay`
	`Router(config)#`
		(Optional) Enables the DHCP relay agent on the CMTS to automatically switch a cable modem or CPE device to a secondary DHCP server or address pool if the primary DHCP server does not respond to three successive requests. If multiple secondary servers have been defined, the relay agent forwards DHCP requests to the secondary servers in a round robin fashion.

	Command or Action	Purpose

Step 4	**ip dhcp ping packet 0**
	Example:
	`Router(config)# ip dhcp ping packet 0`
	`Router(config)#`
		(Optional) Instructs the DHCP server to assign an IP address from its pool without first sending an ICMP ping to test whether a client is already currently using that IP address. Disabling the ping option can speed up address assignment when a large number of modems are trying to connect at the same time. However, disabling the ping option can also result in duplicate IP addresses being assigned if users assign unauthorized static IP addresses to their CPE devices.
		Note By default, the DHCP server pings a pool address twice before assigning a particular address to a requesting client. If the ping is unanswered, the DHCP server assumes that the address is not in use and assigns the address to the requesting client.

Step 5	**ip dhcp relay information check**
	Example:
	`Router(config)# ip dhcp relay information check`
	`Router(config)#`
		(Optional) Configures the DHCP server to validate the relay agent information option in forwarded BOOTREPLY messages. Invalid messages are dropped.
		Note The **ip dhcp relay information** command contains several other options that might be useful for special handling of DHCP packets. See its command reference page in the Cisco IOS-XE documentation for details.

Step 6	**interface cable** *x/y*
	Example:
	`Router(config)# interface cable 4/0`
	`Router(config-if)#`
		Enters cable interface configuration mode for the specified cable interface.

Step 7	**cable dhcp-giaddr policy** [**host** | **stb** | **mta** | **ps**] *giaddr*
	Example:
	`Router(config-if)# cable dhcp-giaddr policy mta 172.1.1.10`
	`Router(config-if)#`
		Sets the DHCP GIADDR field for DHCP request packets to the primary address for cable modems, and the secondary address for CPE devices. This enables the use of separate address pools for different clients.
		• **host**—Specifies the GIADDR for hosts.
		• **mta**—Specifies the GIADDR for MTAs.
		• **ps**—Specifies the GIADDR for PSs.
		• **stb**—Specifies the GIADDR for STBs.

	Command or Action	Purpose
		• *giaddr*—IP addresses of the secondary interface of the bundle interface. **Note** The **cable dhcp-giaddr** command also supports the **primary** option. The **primary** option forces all device types to use only the primary interface IP address as GIADDR and not rotate through the secondary address if the primary address fails.

Step 8	**cable helper-address** *address* [**cable-modem** | **host** | **mta** | **stb**]
	Example:
	`Router(config-if)# cable helper-address 10.10.10.13`
	`Router(config-if)#`
		(Optional) Enables load-balancing of DHCP requests from cable modems and CPE devices by specifying different DHCP servers according to the cable interface or subinterface. You can also specify separate servers for cable modems and CPE devices.
		• *address* = IP address of a DHCP server to which UDP broadcast packets will be sent via unicast packets.
		• **cable-modem** = Specifies this server should only accept cable modem packets (optional).
		• **host** = Specifies this server should only accept CPE device packets (optional).
		• **mta**—(Optional) Specifies this server should only accept MTA packets .
		• **stb** —(Optional) Specifies this server should only accept STB packets .
		Note If you do not specify an option, the helper-address will support all cable devices, and the associated DHCP server will accept DHCP packets from all cable device classes.
		Note If you specify only one option, the other types of devices (cable modem, host, mta, or stb) will not be able to connect with a DHCP server. You must specify each desired option in a separate command
		.

	Command or Action	Purpose
		Tip Repeat this command to specify more than one helper address on each cable interface. You can specify more than 16 helper addresses, but the Cisco IOS software uses only the first 16 valid addresses. **Tip** If you configure different helper addresses on different sub-bundles within a bundle, the cable modem may not come online. We recommend that you use the same helper address on all sub-bundles within a bundle. **Note** The **ip helper-address** command performs a similar function to **cable helper-address**, but it should be used on non-cable interfaces. The **cable helper-address** command should be used on cable interfaces because it is optimized for the operation of DHCP requests on DOCSIS networks.
Step 9	**cable dhcp-giaddr policy** **Example:** `Router(config-if)# cable dhcp-giaddr policy`	Selects the control policy, so the primary address is used for cable modems and the secondary addresses are used for hosts and other customer premises equipment (CPE) devices. This setting is typically used when the CMs on the interface are configured for routing mode, so that the cabel modems and hosts can use IP addresses on different subnets.

Step 10	**exit**
	Example:
	`Router(config-if)# exit`
	`Router(config)#`
		Exits interface configuration mode.

Step 11	**exit**
	Example:
	`Router(config)# exit`
	`Router#`
		Exits global configuration mode.

How to Configure ToD, and TFTP Services

See the following configuration tasks required to configure time-of-day service, and TFTP service on a Cisco CMTS:

Configuration Examples

This section provides examples for the following configurations:

ToD Server Example

The following example shows a typical ToD server configuration:

```
service udp-small-servers max-servers no-limit
cable time-server
```

These are the only commands required to enable the ToD server.

TFTP Server Example

The following lines are an excerpt from a configuration that includes a TFTP server. Change the files listed with the **tftp-server** command to match the specific files that are on your system.

```
! Enable the user of unlimited small servers
 service udp-small-servers max-servers no-limit
!
...
! Enable the TFTP server and specify the files that can be
!   downloaded along with their aliases
 tftp-server disk0:gold.cm alias gold.cm
 tftp-server disk0:silver.cm alias silver.cm
 tftp-server disk0:bronze.cm alias bronze.cm
 tftp-server disk0:ubr924-k8y5-mz.bin alias ubr924-codefile
 tftp-server disk0:ubr925-k9v9y5-mz.bin alias ubr925-codefile
```

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

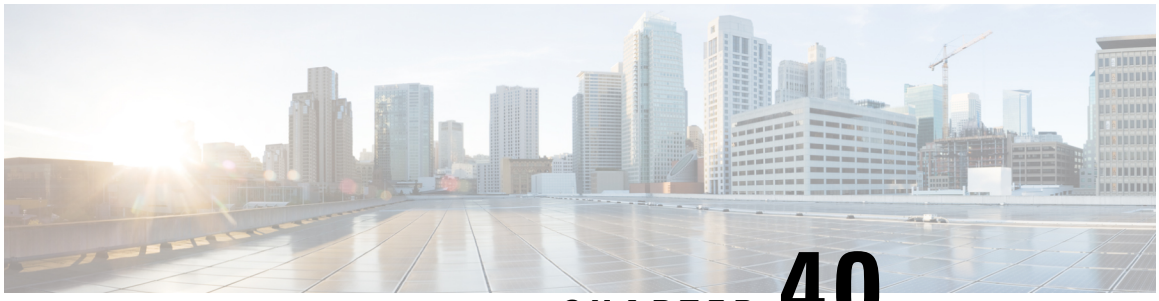
Feature Information for the DHCP, ToD, and TFTP Services for the CMTS Routers

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 106: Feature Information for Downstream Interface Configuration

Feature Name	Releases	Feature Information
DHCP, ToD, and TFTP services	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 40

Virtual Interface Bundling

Virtual Interface Bundling allows supports combining multiple cable interfaces in a Cisco cBR series router into a single logical bundle, so as to conserve IP address space and simplify network management.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 643
- Information About Virtual Interface Bundling, on page 644
- Configuring Virtual Interface Bundling, on page 646
- Verfiying the Virtual Interface Bundling Configuration, on page 649
- Additional References, on page 651
- Feature Information for Virtual Interface Bundling, on page 651

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 107: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About Virtual Interface Bundling

This section contains the following:

Overview of Virtual Interface Bundling

Note All cable bundles are automatically converted and configured to virtual interface bundles. Any standalone cable interfaces must be manually configured to be in a virtual bundle to operate properly.

Virtual interface bundling supports the following:

- Virtual interface bundling uses *bundle interface* and *bundle members* instead of master and slave interfaces.
- A virtual bundle interface is virtually defined, as with IP loopback addresses.
- Virtual interface bundling supports bundle information in multiple **show** commands.

Virtual interface bundling prevents loss of connectivity on physical interfaces should there be a failure, problematic online insertion and removal (OIR) of one line card in the bundle, or erroneous removal of configuration on the master interface.

Virtual interface bundling supports and governs the following Layer 3 settings for the bundle member interfaces:

- IP address
- IP helper-address
- source-verify and lease-timer functions
- cable dhcp-giaddr (The giaddr field is set to the IP address of the DHCP client.)
- Protocol Independent Multicast (PIM)
- Access control lists (ACLs)
- Sub-interfaces
- IPv6
- 1982 bytes layer 3 MTU.

Note In case customer wants to test 1982 bytes MTU by issuing a ping from CMTS to DOCSIS 3.1 modem, **cable mtu-override** command needs to be configured. After the test, please remove this configuration using **no cable mtu-override** command. By default, there is no cable mtu-override configured in bundle interface.

Note This virtual interface for the bundle should always remain on (enabled with **no shutdown**).

Guidelines for Virtual Interface Bundling

The following guidelines describe virtual interface bundling:

- Initial configuration of the first virtual bundle *member* automatically creates a virtual bundle interface.
- All cable bundles are automatically converted and configured to be in a virtual bundle after loading the software image.
- Standalone cable interfaces must be manually configured to be in a virtual bundle to operate properly.
- The virtual bundle interface accumulates the counters from members; counters on member links are not cleared when they are added to the bundle. If a bundle-only counter is desired, clear the bundle counter on the members before adding them to the bundle, or before loading the image.
- This feature supports a maximum of 40 virtual interface bundles, with the numeric range from 1 to 255.
- The virtual bundle interface remains configured unless specifically deleted, even if all members in the bundle are deleted.
- This feature supports subinterfaces on the virtual bundle interface.
- *Bundle-aware* configurations are supported on the virtual bundle interface.
- *Bundle-unaware* configurations are supported on each bundle member.
- While creating the virtual bundle interface, if the bundle interface existed in earlier Cisco IOS releases, then the earlier cable configurations re-appear after upgrade.

Virtual Interface Bundle-aware and Bundle-unaware Support

Virtual interface bundling uses two configurations: the virtual *bundle* itself, and the interfaces in that virtual bundle, known as *bundle members* . The virtual interface bundle and bundle members are either aware of the bundle, or unaware of the bundle, as follows.

- Bundle-aware features are maintained on the virtual *bundle* . These include:
  - IP Address
  - IP helper, cable helper
  - Dhcp-giaddr
  - Sub-interface
  - Source verify
  - Lease-query
  - Address Resolution Protocol (Cable ARP filtering, which also bundles cable interfaces, and Proxy ARP)
  - Cable match
  - Access Control Lists (ACLs)
  - Protocol Independent Multicast (PIM)
  - Cable Intercept
- Bundle-unaware features are maintained on the *bundle members* . These include:
  - DS/US configurations
  - HCCP redundancy
  - Load balancing
  - DMIC, tftp-enforce, shared-secret
  - Spectrum management
  - Admission control
  - Intercept

Configuring Virtual Interface Bundling

To enable virtual interface bundling, and to reconfigure interface information on the Cisco CMTS as required, you first configure the virtual interface bundle, then add additional bundle members for the specified virtual bundle. Perform these steps on each interface, as needed for all virtual interface bundles.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**interface bundle** *n* **Example:** `Router(config-if)# interface bundle 1`	Adds the selected interface to the virtual bundle. If this is the first interface on which the virtual bundle is configured, this command enables the bundle on the specified interface. As many as 40 virtual interface bundles can be configured on the Cisco CMTS. Numeric identifiers may range from 1 to 255.
Step 4	**ip address** *address mask* **Example:** `Router(config-if)# ip address 7.7.7.7 255.255.255.0`	Use as needed after Cisco IOS upgrade. Configures the IP address for the specified interface and virtual bundle.
Step 5	**cable helper-address address** [**cable-modem** | **host** | **mta** | **ps** | **stb**] **Example:** `Router(config-if)# cable helper-address 10.10.10.13`	(Optional) Specifies the IPv4 DHCP server address.
Step 6	**cable dhcp-giaddr** {**primary** | **policy** [**host** | **stb** | **mta** | **ps** | **strict**]} **Example:** `Router(config-if)# cable dhcp-giaddr policy host`	Sets the DHCP GIADDR field for DHCP request packets.
Step 7	**cable source-verify dhcp** **Example:** `Router(config-if)# cable source-verify dhcp`	(Optional) Ensures that the Cisco CMTS allows network access only to those IP addresses that DCHP servers issued to devices on this cable interface. The Cisco CMTS examines the DHCP packets that pass through the cable interfaces to build a database of which IP addresses are valid on which interface. Drops traffic from all devices with unknown IP addresses, but the Cisco CMTS also sends a query to the DHCP servers for any information about the device. If a DHCP server informs the Cisco CMTS that the device has a valid IP address, the CMTS then allows the device on the network.
Step 8	**no cable arp** **Example:** `Router(config-if)# no cable arp`	(Optional) Blocks the static IPv4 CPE from coming online. Also blocks Address Resolution Protocol (ARP) process destined to devices on the cable network.

	Command or Action	Purpose
		Note Use this command, together with the **cable source-verify dhcp** command, to block certain types of scanning attacks that attempt to cause denial of service (DoS) on the Cisco CMTS.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits the interface configuration mode and enters global configuration mode.
Step 10	**interface cable** *slot /subslot/port* **Example:** `Router(config)# interface cable 3/0/0`	Enters interface configuration mode for the selected interface, on which virtual interface bundling is to be enabled.
Step 11	**cable bundle** *n* **Example:** `Router(config-if)# cable bundle 1`	Configures a cable interface to belong to an interface bundle, where *n* is the bundle number.
Step 12	**no cable upstream** *n* **shut** **Example:** `Router(config-if)# no cable upstream 4 shut`	Use as needed after Cisco IOS upgrade. The cable interface must be enabled using the no shutdown command for the specified cable interface. *n* —Specifies the cable interface to enable for the virtual bundle.
Step 13	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

What to do next

To remove a virtual bundle from the interface, use the **no interface bundle** command in interface configuration mode, where *n* specifies the bundle identifier:

no interface bundle *n*

If you remove a member from a bundle, the bundle remains on the interface (even if empty) until the bundle itself is specifically removed.

For more information on configuring IPv6 parameters for bundle interface, see *IPv6 on Cable* feature guide.

Verfiying the Virtual Interface Bundling Configuration

- **show ip interface brief**—Displays the summary of interfaces.

  Following is a sample output of this command:

```
Router# show ip interface brief

Interface              IP-Address      OK? Method Status                Protocol
Cable3/0/0             Bundle1         YES unset  up                    up
GigabitEthernet0       10.86.3.175     YES NVRAM  administratively down down
Bundle1                100.1.2.1       YES manual up                    up
Bundle2                100.1.3.1       YES NVRAM  up                    up
Dti4/1/0               unassigned      YES unset  administratively down down
Dti5/1/0               unassigned      YES unset  administratively down down
Dti4/1/1               unassigned      YES unset  administratively down down
Dti5/1/1               unassigned      YES unset  administratively down down
Loopback1              1.2.3.4         YES NVRAM  up                    up
Tunnel0                unassigned      YES unset  up                    up
```

- **show running-config interface bundle** *n*—Displays the information about the specified bundle.

  Following is a sample output of this command:

```
Router# show running-config interface Bundle 1

Current configuration : 696 bytes
!
interface Bundle2
 ip address 100.1.3.1 255.255.255.0
 no cable nd
 cable arp filter request-send 3 2
 cable arp filter reply-accept 3 2
 no cable arp
 cable ipv6 source-verify dhcp
 cable source-verify dhcp
 cable dhcp-giaddr primary
 cable helper-address 10.10.0.53
 ipv6 address 2001:420:3800:910::1/64
 ipv6 enable
 ipv6 nd reachable-time 3600000
 ipv6 nd cache expire 65536
 ipv6 nd managed-config-flag
 ipv6 nd other-config-flag
 ipv6 nd ra interval msec 2000
 no ipv6 redirects
 ipv6 dhcp relay destination 2001:420:3800:800:250:56FF:FEB2:F11D
 ipv6 dhcp relay destination vrf vrfa 2001:420:3800:800:250:56FF:FEB2:F11D
 ipv6 dhcp relay source-interface Bundle2
 arp timeout 2147483
```

- **show ip interface brief | include bundle**—Displays the bundle interface information.

  Following is a sample output of this command:

```
Router# show ip interface brief | include Bundle

Bundle1 unassigned YES unset up up
Bundle1.1 100.1.2.1 YES NVRAM up up
Bundle2 100.1.3.1 YES NVRAM up up
```

- **show running-config interface bundle** *n.n*—Displays the subinterface information for the specified bundle.

  Following is a sample output of this command:

```
Router# show running-config interface bundle 1.1

Current configuration : 1415 bytes
!
interface Bundle1.1
ip address 100.1.2.1 255.255.255.0
ip pim sparse-mode
ip rip send version 2
ip rip receive version 2
ip rip authentication mode md5
ip rip authentication key-chain ubr-rip
ip igmp static-group 239.1.4.1 source 115.255.0.100
ip igmp static-group 239.1.3.1 source 115.255.0.100
ip igmp static-group 239.1.2.1 source 115.255.0.100
ip igmp static-group 232.1.4.1 source 115.255.0.100
ip igmp static-group 232.1.3.1 source 115.255.0.100
ip igmp static-group 232.1.2.1 source 115.255.0.100
ip igmp static-group 232.1.1.1 source 115.255.0.100
ip igmp static-group 230.1.4.1
ip igmp static-group 230.1.3.1
ip igmp static-group 230.1.2.1
ip igmp static-group 224.1.4.1
ip igmp static-group 224.1.3.1
ip igmp static-group 224.1.2.1
ip igmp static-group 224.1.1.1
ip igmp version 3
ip igmp query-interval 20
no cable arp
cable ipv6 source-verify dhcp
cable source-verify dhcp
cable dhcp-giaddr primary
cable helper-address 10.10.0.53
ipv6 address 2001:420:3800:909::1/64
ipv6 enable
ipv6 nd reachable-time 3600000
ipv6 nd cache expire 65536
ipv6 nd prefix default no-advertise
ipv6 nd managed-config-flag
ipv6 nd other-config-flag
ipv6 nd ra interval msec 2000
no ipv6 redirects
ipv6 dhcp relay destination 2001:420:3800:800:250:56FF:FEB2:F11D link-address
2001:420:3800:909::1
ipv6 dhcp relay source-interface Bundle1
ipv6 rip CST enable
arp timeout 2147483
```

Additional References

Related Documents

Related Topic	Document Title
CMTS Command Reference	Cisco IOS CMTS Cable Command Reference Guide

Standards and RFCs

Standards	Title
SP-RFIv1.1-I09-020830	Data-over-Cable Service Interface Specifications Radio Frequency Interface Specification, version 1.1
SP-RFIv2.0-I03-021218	Data-over-Cable Service Interface Specifications Radio Frequency Interface Specification, version 2.0
SP-OSSIv2.0-I03-021218	Data-over-Cable Service Interface Specifications Operations Support System Interface Specification, version 2.0
SP-BPI+-I09-020830	Data-over-Cable Service Interface Specifications Baseline Privacy Plus Interface Specification, version 2.0

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

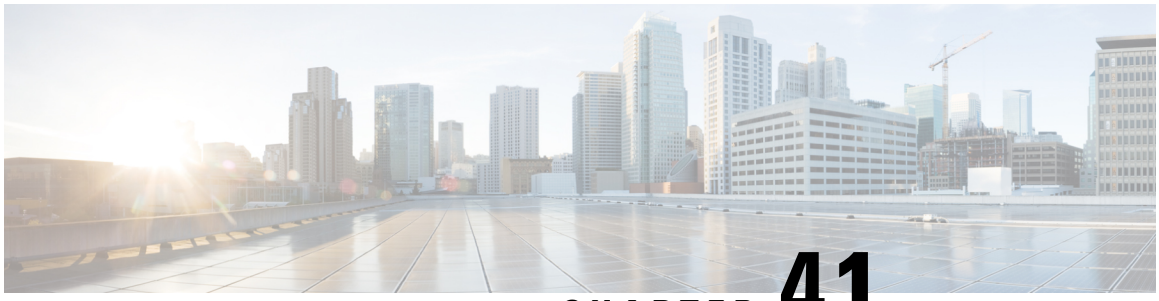
Feature Information for Virtual Interface Bundling

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 108: Feature Information for Virtual Interface Bundling

Feature Name	Releases	Feature Information
Virtual interface bundling	Cisco IOS-XE 16.5.1	This feature was integrated into Cisco <polaris release label> on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 41

IPv6 on Cable

Cisco cBR series Converged Broadband Router supports full IPv6 functionality.

The IPv6 feature support available in the Cisco IOS software and for Cisco CMTS routers is extensive. This document provides a comprehensive overview of all of the IPv6 features supported on the Cisco CMTS routers, and their restrictions.

However, the details of every feature are not covered in this document. The areas of IPv6 protocol support for the Cisco CMTS routers discussed in this document are classified by platform-independence or by platform-specific feature support.

- Platform-independent IPv6 features—Describes IPv6 features that are supported in the Cisco IOS software for several other Cisco platforms, and which generally do not have any platform-specific behavior or configuration differences on the Cisco CMTS routers.
- Documentation about the restrictions for these platform-independent features can be found in the Restrictions for IPv6 on Cable.
- Detailed information about these features, including conceptual and task-based configuration information, is documented outside of this feature and in the Cisco IOS software documentation. Detailed information about the location of this related documentation in the Cisco IOS software documentation is described in the Feature Information for IPv6 on Cable.

Platform-specific IPv6 features—Describes IPv6 features that are specific to the cable technology area and that only apply to the supported Cisco CMTS routers. The cable-specific IPv6 feature support includes new or modified cable features supporting IPv6, and any transparent support of the IPv6 protocol in existing (legacy) cable features on the CMTS router platforms.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 654
- Restrictions for IPv6 on Cable, on page 655
- Information About IPv6 on Cable, on page 656

• How to Configure IPv6 on Cable , on page 664
• How to Verify IPv6 Dual Stack CPE Support , on page 679
• Configuration Examples for IPv6 on Cable, on page 681
• Verifying IPv6 on Cable, on page 692
• Additional References, on page 694
• Feature Information for IPv6 on Cable , on page 694

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 109: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for IPv6 on Cable

Multicast Restrictions

IPv6 multicast has the following behavior restrictions on the Cisco CMTS routers:

- ICMP redirects are not sent to the originating host if the packet is destined for another CPE behind the same CM. All CPE-to-CPE traffic is processed by the Cisco CMTS router.
- IPv6 multicast forwarding is not supported in Parallel Express Forwarding (PXF), therefore, the IPv6 multicast forwarding performance is limited by the Router Processor (RP).

The following areas of IPv6 multicast are not supported by the Cisco CMTS routers:

- Address family support for Multiprotocol Border Gateway Protocol (MBGP)
- Bidirectional Protocol Independent Multicast (PIM)
- Bootstrap router (BSR)
- DOCSIS 3.0 encrypted multicast
- Explicit tracking of receivers
- IPv6 multicast echo
- Multicast Forwarding Information Base (MFIB) display enhancements
- Multicast use authentication and profile support
- PIM embedded rendezvous point
- Protocol Independent Multicast sparse mode (PIM-SM) accept register feature
- Reverse path forwarding (RPF) flooding of bootstrap router (BSR) packets
- Routable address hello option
- Source Specific Multicast (SSM) mapping for Multicast Listener Device (MLD) version 1 SSM

QoS Restrictions

In Cisco IOS-XE Release 16.5.1, the following fields are supported for theIPv6 downstream classification:

- IPv6 dest addr
- ipv6 src addr
- IPv6 next header
- IPv6 traffic class

Note IPv6 flow label field is not supported.

The following areas of DOCSIS QoS are not supported by the Cisco CMTS routers:

- Upstream IPv6 Type of Service (ToS) overwrite
- Downstream IPv6 classification

Note ToS overwrite, DOCSIS classification, and Modular QoS CLI (MQC) on Gigabit Ethernet are supported.

Information About IPv6 on Cable

This section includes the following topics:

Features Supported

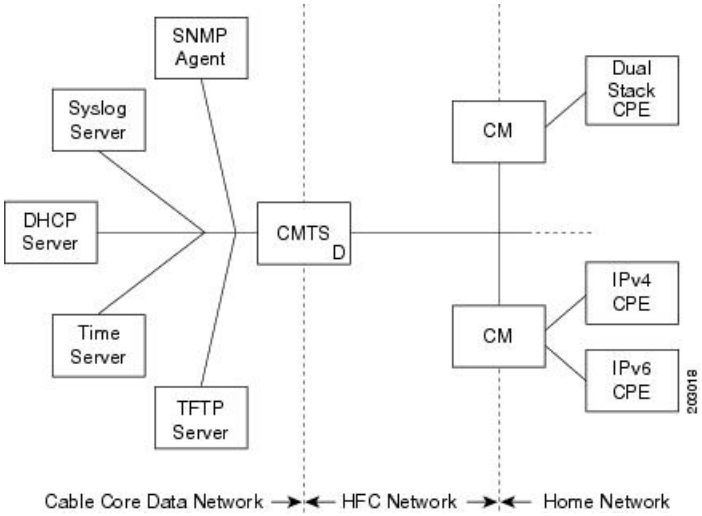
The following features are supported on the Cisco CMTS routers:

- Source verification of IPv6 packets in PXF
- ACL support for PXF
- ToS overwrite
- DOCSIS classification
- Modular QoS CLI (MQC) on Gigabit Ethernet
- IPv6 DOCSIS RP and LC HA and DCC
- MAC tapping of IPv6 packets
- Equal cost route load balancing of IPv6 packets destined to the backhaul
- IPv6 over IPv4 GRE tunnels
- Assignment of different prefixes to CM and CPE
- DHCPv6 over MPLS-VPN
- DHCPv6 relay prefix delegation VRF awareness
- Assignment of multiple IAPDs in a single advertise for each CPE.
- Assignment of multiple IA_NA and IAPD combinations to multiple CPEs behind a CM.
- The default maximum number of IA_NA and IAPD combinations for each cable modem is 16, including link-local addresses.
- IPv6 Downstream ToS overwrite.
- DHCPv6 Client Link-Layer Address Option (RFC 6939).
- Voice over IPv6. PacketCable Multimedia needs to be enabled before using this feature. For more information, see http://www.cisco.com/c/en/us/td/docs/cable/cbr/configuration/guide/b_pktcbl_pktcblmm/packetcable_and_packetcable_multimedia.html.

Overview of the DOCSIS 3.0 Network Model Supporting IPv6

Figure below illustrates the network model described by the *DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification*.

Figure 25: DOCSIS 3.0 Network Model



In this model, the different devices support the following functions and services:

- Customer premises equipment (CPE)—Supports IPv4, IPv6, or dual stack operation.

Note Cisco cBR routers support CPE devices provisioned for dual stack operation.

- Cable modem (CM)—Functions as a bridging device and supports IPv4, IPv6, or dual stack operation.
- Cable modem termination system (CMTS) router—Works with the CM over the hybrid fiber coaxial cable (HFC) network to provide IPv4 and IPv6 network connectivity to the provisioning servers and the core data network behind the CMTS router.

The CMTS router supports IPv6 address assignment, routing, and forwarding of IPv6 multicast and unicast packets.

Note The Cisco cBR router supports only a single DHCPv6 IPv6 address per client cable modem or CPE. This restriction also applies to DHCPv6 Prefix Delegation prefixes. The reason for blocking more than one DHCPv6 address or prefix for a client is because the end-to-end network requires Source Address Selection (SAS) and all nodes in the end-to-end network may not support the correct SAS. Moreover, the SAS specification (RFC 3484) is being revised by the IETF to define the correct SAS behavior.

- Simple Network Management Protocol (SNMP) agent—Provides management tools to configure and query devices on the network.
- Syslog server—Collects messages from the CM to support its functions.
- Dynamic Host Control Protocol (DHCP) server—The DOCSIS 3.0 network model supports both DHCPv4 and DHCPv6 servers to control the assignment of IP addresses.
- Time server—Provides the current time to the CM.

• Trivial File Transport Protocol (TFTP) server—Provides the CM configuration file.

Overview of Cable Modem IPv6 Address Provisioning

Prior to cable modem registration with a CMTS router, the CMTS router sends a MAC Domain Descriptor (MDD) message to provide information to the cable modem about its supported IP provisioning mode. You configure the CMTS router provisioning mode using the **cable ip-init** interface configuration command. For more information, see the Implementing IPv6 Addressing and Basic Connectivity for Cable Interfaces and Bundles, on page 666.

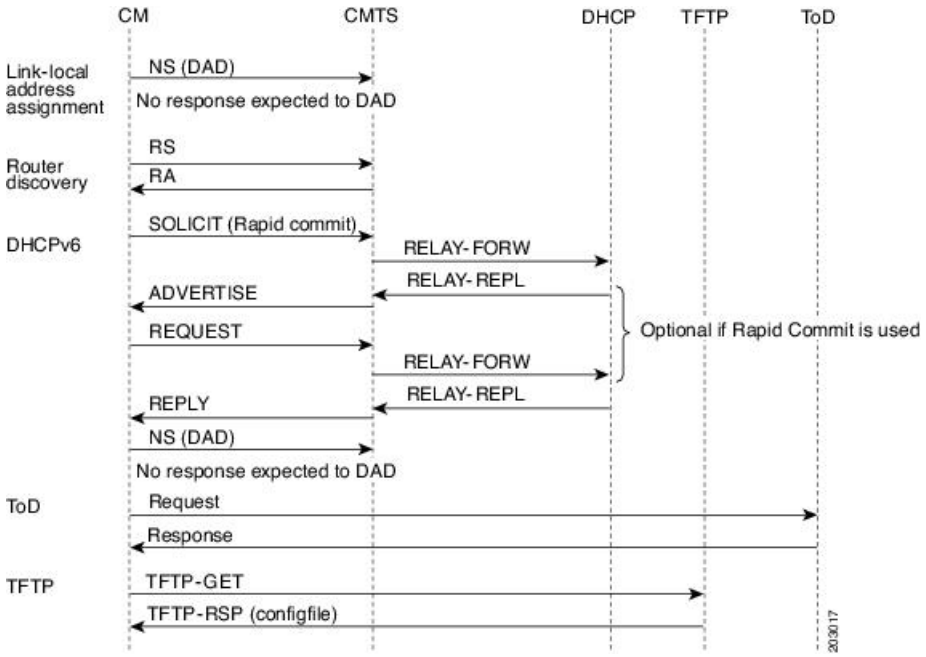
The MDD contains an IP initialization parameters type length value (TLV) that defines the IP version, management and alternate provisioning mode, and pre-registration downstream service ID (DSID) that is used by cable modems that are capable of downstream traffic filtering.

Note The Cisco CMTS routers do not support alternate provisioning mode or pre-registration DSID.

To support the MULPIv3.0 I04 or later version of the *DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification*, the cable modem must attempt IPv6 address acquisition first.

Figure below illustrates the message flow between a cable modem, the CMTS router, and the DHCP server when the cable modem is requesting an IPv6 address.

Figure 26: Message Flow for CM Provisioning of DHCP IPv6 Address Assignment



1. Link-local address assignment—The cable modem sends a Neighbor Solicit (NS) message with its link-local address (LLA) to the CMTS router, which starts the duplicate address detection (DAD) process for that LLA. The cable modem expects no response to the NS message.
2. Router discovery—The cable modem listens to the downstream to detect periodical Router Advertise (RA) messages. When an RA message is detected, the cable modem uses the data in the RA message to

configure the default route. If an RA is not detected in a specified period, the cable modem sends a Router Solicit (RS) message to find the router on the link (all nodes multicast). The CMTS router responds with a Router Advertise (RA) message with theM and O bits set to 1 to instruct the CM to perform stateful address configuration.

Note Cisco CMTS routers do not support SLAAC address assignment.

- DHCPv6—The cable modem sends a DHCPv6 Solicit message to the CMTS router to request an IPv6 address. The CMTS router relays this message to the DHCPv6 servers. The DHCPv6 servers send an Advertise message indicating the server's availability.

If the Rapid-Commit option is not used by the cable modem, then the cable modem responds to the Advertise message of the server with a Request message to select the server that the CMTS router relays to the DHCPv6 server. If the Rapid-Commit option is used, then multiple DHCPv6 servers that could assign different addresses to the same CPE must not be used.

The cable modem starts the DAD process to verify the uniqueness of the IPv6 address that the DHCPv6 server assigns to it.

- TFTP and Time of Day (ToD)—Once the CM establishes IP connectivity, it sends a request to the TFTP server to download a configuration file and requests the current time from the ToD server to complete its boot process.

Overview of IPv6 Dual Stack CPE Support on the CMTS

Most operating systems (OS) deployed at homes support dual stack operation. Cisco CMTS supports dual stack, which is both IPv4 and IPv6 addressing on the CPE.

Overview of IPv6 over Subinterfaces

Cisco CMTS supports IPv6 over bundle subinterfaces. To configure IPv6 on bundle subinterfaces, see the Implementing IPv6 Addressing and Basic Connectivity for Cable Interfaces and Bundles, on page 666 section. For a CMTS bundle configuration example, see the Example: IPv6 over Subinterfaces , on page 681 section.

To enable IPv6 on subinterfaces, configure IPv6 on bundle subinterfaces and not the bundle. Reset the CMs after the subinterface is configured.

Note MPLS VPN over subinterfaces for IPv6 is not supported.

Overview of High Availability on IPv6

Cisco cBR Series routers support IPv6 HA for the Supervisor card.

Note IPv6 DOCSIS HA and HCCP is supported on the Cisco CMTS routers.

The IPv6 HA feature support in Cisco CMTS routers covers the following capabilities:

- DOCSIS PRE HA
- DOCSIS line card HA
- Dynamic Channel Change (DCC)

DOCSIS PRE HA

The DOCSIS PRE HA has the following behavior restrictions and prerequisites on the Cisco CMTS routers:

- The CMs and CPEs should not go offline after a PRE switchover.
- The data structures of the IPv6 CM and CPE should be synchronized to the standby PRE before the PRE switchover. Both dynamic and bulk synchronization is supported.
- Single stack, dual stack, and APM are supported for the CM.
- Single stack and dual stack provisioning modes are supported on the CPE.
- After a PRE switchover, the IPv6 neighbor entries are rebuilt by Neighbor Discovery (ND) messages on the standby PRE, and the IPv6 routes are rebuilt after converging the routing protocol.

DOCSIS Line Card HA

The DOCSIS line card HA has the following behavior restrictions and prerequisites on the Cisco CMTS routers:

- The data structures of the IPv6 CM and CPE should be synchronized to the standby line card before the line card switchover. Both dynamic and bulk synchronization is supported.
- The CMs and CPEs should not fall offline after a line card switches over and reverts; the CMs and CPEs should behave the same as before the switchover.
- The DOCSIS line card HA supports both 4+1 and 7+1 redundancy.
- Traffic outages in IPv6 may be longer because traffic recovery occurs only after converging the routing protocol.

Dynamic Channel Change

The Dynamic Channel Change (DCC) feature is supported on Cisco CMTS routers.

Note The behavior of the DCC for single stack IPv6 CM and CPE, or dual stack CM and CPE is the same as that of a single stack IPv4 CM and CPE.

The IPv6 and IPv4 DCC functionality has the following behavior restrictions and prerequisites on the Cisco CMTS routers:

Narrowband Cable Modem

- If the source and destination MAC domains of the CM are on the same line card, DCC initialization techniques 0, 1, 2, 3, and 4 are used to move the CM and its associated CPE from one upstream or downstream to another; or move the CM and CPE from one upstream and downstream combination to another.
- If the source and destination MAC domains of the CM are on different line cards, you can use only the DCC initialization technique 0 to move the CM and its associated CPE across line cards.

Wideband Cable Modem

- If the source and destination MAC domains of the CM are on the same line card, DCC initialization techniques 0, 1, 2, 3, and 4 are used to move the CM and its associated CPE from one upstream to another.
- If the primary downstream of a CM is changed after DCC, you can use only the DCC initialization technique 0 to move the CM and its associated CPE across line cards.

Overview of IPv6 VPN over MPLS

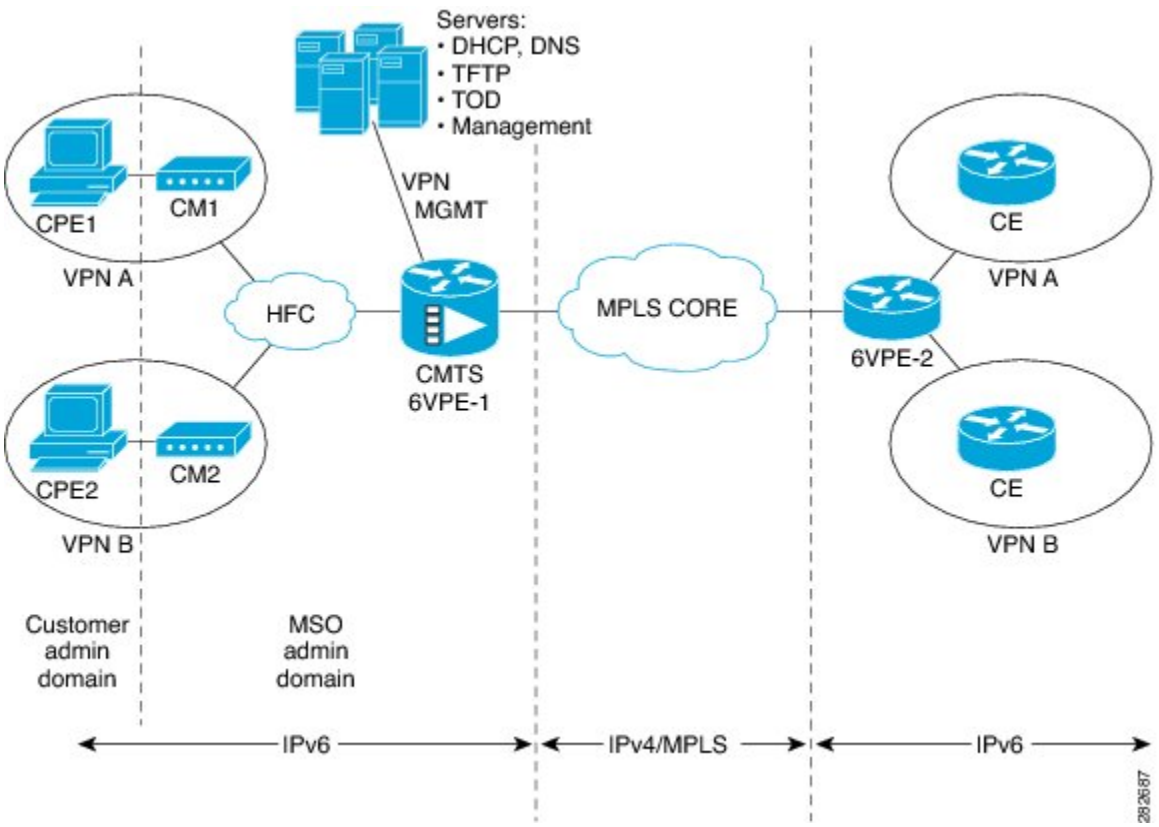
The Multiprotocol Label Switching (MPLS) VPN feature represents an implementation of the provider edge (PE) based VPN model. This document describes the IPv6 VPN over MPLS (6VPE) feature.

The 6VPE feature allows Service Providers to provide an IPv6 VPN service that does not require an upgrade or reconfiguration of the PE routers in the IPv4 MPLS Core. The resulting IPv6 VPN service has a configuration and operation which is virtually identical to the current IPv4 VPN service.

In principle, there is no difference between IPv4 and IPv6 VPNs. In both IPv4 and IPv6, the multiprotocol BGP is the core of the MPLS VPN for IPv6 (VPNv6) architecture. It is used to distribute IPv6 routes over the service provider backbone using the same procedures to work with overlapping addresses, redistribution policies, and scalability issues.

Figure below illustrates the 6PE/6VPE reference architecture diagram.

Figure 27: 6PE/6VPE Reference Architecture



Cable Monitor

The Cable Monitor and Intercept features for Cisco CMTS routers provide a software solution for monitoring and intercepting traffic coming from a cable network. These features give service providers Lawful Intercept capabilities.

For more information, see Cable Monitor and Intercept Features for the Cisco CMTS Routers guide at: http://www.cisco.com/en/US/docs/ios/cable/configuration/guide/cmts_mon_intrcpt.html

Overview of IPv6 CPE Router Support on the Cisco CMTS

The IPv6 CPE router support is provided on the Cisco CMTS. The IPv6 CPE router is a node primarily for home or small office use that connects the end-user network to a service provider network. It is also referred to as the home router.

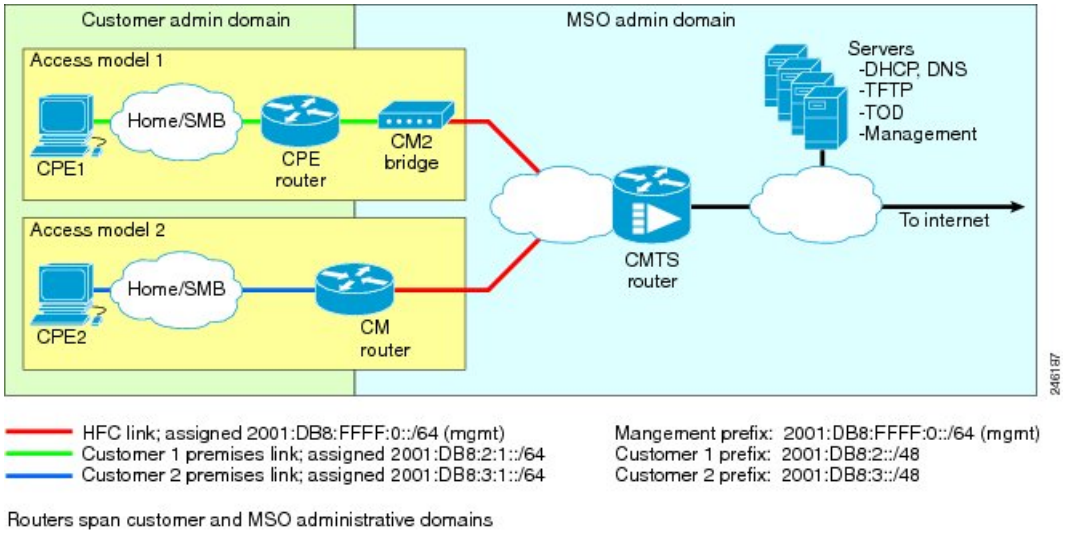
The IPv6 CPE router is responsible for implementing IPv6 routing; that is, the IPv6 CPE router looks up the IPv6 destination address in its routing table and decides to which interface the packet should be sent.

The IPv6 CPE router performs the following functions:

- Provisions its WAN interface automatically.
- Acquires IP address space for provisioning of its LAN interfaces.
- Fetches other configuration information from the service provider network.

Figure below illustrates the CPE router reference architecture diagram between the CPE router, the CMTS, and the DHCPv6 server (CNR) when the CM is requesting an IPv6 address.

Figure 28: IPv6 CPE Router Reference Architecture



As part of the IPv6 CPE Router Support feature, the following enhancements are introduced:

- Support to IPv6 router devices.
- IPv6 Prefix Delegation (PD) High Availability.
- Prefix awareness support in IPv6 cable source-verify, Cable DOCSIS filters code, and packet intercepts.

Support for IPv6 Prefix Stability on the CMTS

IPv6 prefix stability is supported on the Cisco CMTS as specified in DOCSIS 3.0 MULPI CM-SP-MULPIv3.0-I15-110210 standard. The IPv6 prefix stability allows an IPv6 home router to move from one Cisco CMTS to another while retaining the same prefix.

The multiple service operators (MSOs) can use this feature to allow their business customers (with IPv6 routers) to retain the same IPv6 prefix during a node split.

Configurable DHCPv6 Relay Address

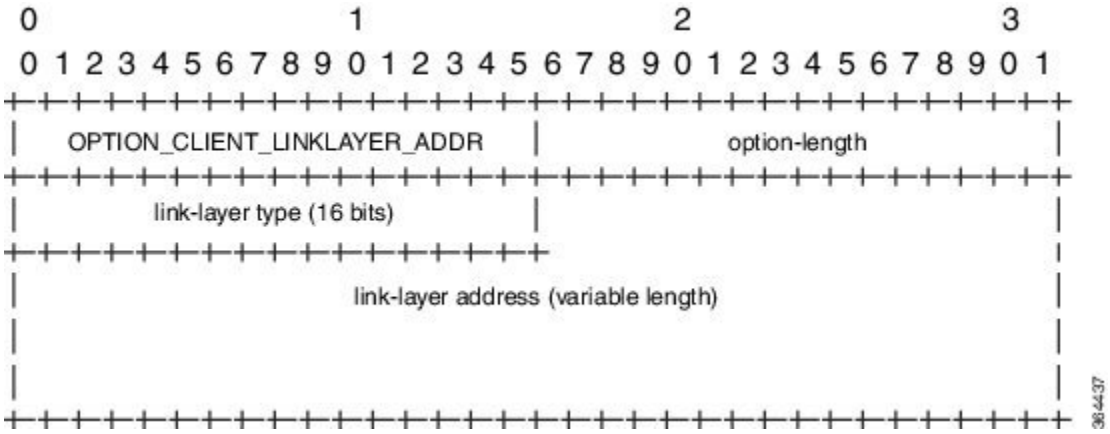
The DHCPv6 Cisco IOS relay agent on the Cisco CMTS router sends relay-forward messages from a source address to all configured relay destinations. The source address is either an IPv6 address provisioned on the network interface or a Cisco CMTS WAN IPv6 address. The relay destination can be a unicast address of a server, another relay agent, or a multicast address. The relay-forward messages contain specific DHCPv6 link-addresses.

A DHCP relay agent is used to relay messages between the client and server. A client locates a DHCP server using a reserved, link-scoped multicast address.

DHCPv6 Client Link-Layer Address Option (RFC 6939)

Cisco IOS-XE Releases support DHCPv6 Client Link-Layer Address Option (RFC 6939). It defines an optional mechanism and the related DHCPv6 option to allow first-hop DHCPv6 relay agents (relay agents that are connected to the same link as the client) to provide the client's link-layer address in the DHCPv6 messages being sent towards the server.

The format of the DHCPv6 Client Link-Layer Address option is shown below.



Name	Description
option-code	OPTION_CLIENT_LINKLAYER_ADDR (79)
option-length	2 + length of MAC address
link-layer type	CPE or CM MAC address type. The link-layer type MUST be a valid hardware type assigned by the IANA, as described in RFC0826.

Name	Description
link-layer address	MAC address of the CPE or CM.

Note RFC6939 is enabled by default. It can not be enabled/disabled by any CLI command.

To configure DHCPv6 Relay Address on the Cisco CMTS bundle subinterfaces, see the Configuring DHCPv6 Relay Agent, on page 678 section.

For more information about the DHCPv6 client, server, and relay functions, see the "Implementing DHCP for IPv6" chapter in the IPv6 Implementation Guide, Cisco IOS XE Release 3S.

Support for Multiple IAPDs in a Single Advertise

Assignment of multiple IA_NA and IAPD to CPEs behind a CM is supported on Cisco CMTS routers. This feature includes support for link-local addresses and IA_NA and IAPD. However, a CM can be assigned only one IA_NA. This IA_NA can be either static or DHCP-assigned.

The CPEs behind the CM can request for multiple DHCPv6 IA_NAs and IAPDs. Each CPE is assigned multiple IA_NAs and IAPDs in a single Advertise/Reply message. Each CPE request for IA_NA and IAPD is treated as a separate Advertise/Reply message.

IPv6 Neighbor Discovery Gleaning

The IPv6 Neighbor Discovery (ND) Gleaning feature enables Cisco CMTS routers to automatically recover lost IPv6 CPE addresses and update the CPE records in the Cisco CMTS subscriber database. The Cisco CMTS router gleans only the solicited neighbor advertise (NA) messages transmitted in the upstream direction. IPv6 ND gleaning is similar to Address Resolution Protocol (ARP) gleaning for IPv4 CPE recovery.

The IPv6 ND Gleaning feature is configured by default on Cisco CMTS routers. To disable this feature, use the **no** form of the **cable nd** command in bundle interface configuration mode. The **cable nd** command adds a CPE (host behind a cable modem) to the Cisco CMTS subscriber database. This command does not impact the IPv6 ND protocol operation on the router.

Note The IPv6 ND Gleaning feature does not support gleaning of NA messages transmitted in the downstream direction.

How to Configure IPv6 on Cable

This section includes the following tasks:

Configuring IPv6 Switching Services

The CMTS routers support forwarding of unicast and multicast IPv6 traffic using either Cisco Express Forwarding for IPv6 (CEFv6) or distributed CEFv6 (dCEFv6):

- CEFv6—All CMTS platforms
- dCEFv6—Cisco uBR10012 universal broadband router only

The CMTS routers also support Unicast Reverse Path Forwarding (RPF), as long as you enable Cisco Express Forwarding switching or distributed Cisco Express Forwarding switching globally on the router. There is no need to configure the input interface for Cisco Express Forwarding switching. As long as Cisco Express Forwarding is running on the router, individual interfaces can be configured with other switching modes.

To configure forwarding of IPv6 traffic using Cisco Express Forwarding or distributed Cisco Express Forwarding (supported on the Cisco uBR10012 universal broadband router only) on the CMTS routers, you must configure forwarding of IPv6 unicast datagrams using the **ipv6 unicast-routing** global configuration command, and you must configure an IPv6 address on the bundle interface using the **ipv6 address** command.

The **show ipv6 cef platform** command is supported on the Cisco CMTS platform. You can use the **show ipv6 cef platform** command for debugging purposes.

Before you begin

- You must enable Cisco Express Forwarding for IPv4 globally on the router by using the **ip cef** or **ip cef distributed** command before configuring Cisco Express Forwarding v6 or distributed Cisco Express Forwarding v6.

Note The **ip cef** command is enabled by default on all Cisco CMTS routers. Therefore, you only must configure the command if it has been disabled. However, you must explicitly configure the **ip cef distributed** command on a Cisco uBR10012 universal broadband router if you want to run distributed CEF switching services for IPv4 or IPv6.

- You must configure forwarding of IPv6 unicast datagrams using the **ipv6 unicast-routing** global configuration command.
- You must configure IPv6 addressing on the cable bundle interface.
- CEF switching is required for Unicast RPF to work.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	Do one of the following:	Enables Cisco Express Forwarding.

	Command or Action	Purpose

	• **ip cef**
	• **ip cef distributed**
	Example:
	`Router(config)# ip cef`
	`or`
	`Router(config)# ip cef distributed`
		or
		Enables distributed Cisco Express Forwarding for IPv4 datagrams.
		Note For CMTS routers, distributed Cisco Express Forwarding is supported only on a Cisco uBR10012 universal broadband router.

Step 4	Do one of the following:
	• **ipv6 cef**
	• **ipv6 cef distributed**
	Example:
	`Router(config)# ipv6 cef`
	`or`
	`Router(config)# ipv6 cef distributed`
		Enables Cisco Express Forwarding v6.
		or
		Enables distributed Cisco Express Forwarding v6 for IPv6 datagrams.
		Note For CMTS routers, distributed Cisco Express Forwarding v6 is supported only on a Cisco uBR10012 universal broadband router.

Step 5	**ipv6 unicast-routing** **Example:** `Router(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.

What to do next

- (Optional) Enable IPv6 multicast routing using the **ipv6 multicast-routing** command in global configuration mode and configure other multicast features.

Implementing IPv6 Addressing and Basic Connectivity for Cable Interfaces and Bundles

Configuring the Cable Virtual Bundle Interface

The only required IPv6 configuration on a cable line card interface is the IP provisioning mode. The remainder of the IPv6 features are configured at the virtual bundle interface, which is then associated with a particular cable line card interface to establish its configuration.

Most of the IPv6 features that are supported in interface configuration mode (both cable-specific as well as platform-independent IPv6 features) are configured at a cable bundle interface.

The Cisco CMTS routers support IPv6 routing on the bundle interface and map both IPv6 unicast and multicast addresses into the cable bundle forwarding table, for packet forwarding.

Each bundle interface has a unique link-local address (LLA) to support link-local traffic when IPv6 is enabled. Cisco CMTS routers can support a maximum of 40 active bundle interfaces, which also translates to a maximum of 40 active IPv6-enabled bundle interfaces.

IPv6 commands can be configured on multiple bundle subinterfaces.

Before you begin

The **cable ipv6 source-verify** and **cable nd** commands are not compatible with each other in Cisco IOS release 12.2(33)SCE and later. You must disable IPv6 ND gleaning using the **no** form of the **cable nd** command before using the **cable ipv6 source-verify** command to ensure that only DHCPv6 and SAV-based CPEs can send traffic on the router.

Restriction All multicast traffic is flooded onto bundle member interfaces.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface bundle** *n* **Example:** `Router(config)# interface bundle 1`	Specifies the cable bundle interface and enters interface configuration mode, where *n* specifies the number of the bundle interface.
Step 4	**ipv6 address***ipv6-prefix* / *prefix-length* [**eui-64**] **Example:** `Router(config-if)# ipv6 address 2001:DB8::/32 eui-64`	Specifies an IPv6 network assigned to the interface and enables IPv6 processing on the interface. The ipv6 address eui-64 command configures site-local and global IPv6 addresses with an interface identifier (ID) in the low-order 64 bits of the IPv6 address. You need to specify only the 64-bit network prefix for the address; the last 64 bits are automatically computed from the interface ID.
Step 5	**ipv6 address***ipv6-prefix* / *prefix-length* **link-local** **Example:** `Router(config-if)# ipv6 address 2001:DB8::/32 link-local`	(Optional) Specifies an IPv6 address assigned to the interface and enables IPv6 processing on the interface. The **ipv6 address link-local** command configures a link-local address on the interface that is used instead of the link-local address that is automatically configured, when IPv6 is enabled on the interface (using the **ipv6 enable** command).

	Command or Action	Purpose
Step 6	**ipv6 enable** **Example:** `Router(config-if)# ipv6 enable`	Automatically configures an IPv6 link-local address on the interface while also enabling the interface for IPv6 processing. The link-local address can be used only to communicate with nodes on the same link.
Step 7	cable ipv6 source-verify **Example:** `Router(config-if)# cable ipv6 source-verify`	(Optional) Enables source verification of MAC address-MD-SID-IPv6 address binding packets received by a cable interface upstream on Cisco CMTS routers.

What to do next

- Configure the desired platform-independent IPv6 features on the bundle interface, such as Neighbor Discovery and DHCPv6 features.
- Configure the IP provisioning mode and bundle on the cable interface.

Configuring the IP Provisioning Mode and Bundle on the Cable Interface

The CMTS routers allow you to configure cable interfaces to support cable modems provisioned for both IPv4 and IPv6 addressing support (known as "dual stack"), only IPv4 addressing, or only IPv6 addressing. Prior to cable modem registration, the CMTS router sends its supported provisioning mode to the cable modem in the MDD message.

In addition to configuring the provisioning mode on the cable interface, you must also associate the cable interface with a cable bundle. You perform most of the other IPv6 feature configuration at the bundle interface.

Note This section describes only the commands associated with establishing IPv6 support on a CMTS router. Other cable interface commands that apply but are optional are not shown, such as to configure upstream and downstream features.

Before you begin

Configuration of a bundle interface is required.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**interface cable** {*slot* / *port* | *slot* / *subslot* /*port* } **Example:** `Router(config)# interface cable 5/0/1`	Specifies the cable interface line card, where: The valid values for these arguments are dependent on your CMTS router and cable interface line card. Refer to the hardware documentation for your router chassis and cable interface line card for supported slot and port numbering.
Step 4	**cable ip-init** {**apm | dual-stack** | **ipv4** | **ipv6**} **Example:** `Router(config-if)# cable ip-init ipv6`	Specifies the IP provisioning mode supported by the cable interface, where:
Step 5	**cable bundle***n* **Example:** `Router(config)# cable bundle 1`	Associates the cable interface with a configured virtual bundle interface, where *n* specifies the number of the bundle interface.

What to do next

- Proceed to configuring any other cable interface features that you want to support, such as upstream and downstream features. For more information about the other cable interface features, refer to the *Cisco IOS CMTS Cable Software Configuration Guide* .
- Proceed to configure other optional IPv6 cable features.

Configuring IPv6 Cable Filter Groups

The Cisco CMTS router supports IPv6 cable filter group capability with IPv6 filter options.

Configuring IPv6 Cable Filter Groups

The Cisco CMTS router supports IPv6 cable filter group capability with IPv6 filter options.

Cable Filter Groups and the DOCSIS Subscriber Management MIB

Cable subscriber management is a DOCSIS 1.1 specification, which can be established using the following configuration methods:

- CMTS router configuration (via CLI)
- SNMP configuration
- DOCSIS 1.1 configuration file (TLVs 35, 36, and 37)

This section describes the IPv6 cable filter group feature support of the packet filtering portion of the DOCSIS Subscriber Management MIB (DOCS-SUBMGMT-MIB) using configuration commands on the CMTS routers. This IPv6 cable filter group support extends filter classifiers with IPv6 addressing options for CM and CPE traffic, but is independent of DOCSIS IPv6 classifiers, which are used to match packets to service flows.

Configuration of IPv6 cable filter groups on the CMTS routers is supported according to the following guidelines:

- A cable filter group consists of a set of **cable filter group** commands that share the same group ID.
- Separate indexes can be used to define different sets of filters for the same group ID. This can be used to define both IPv4 and IPv6 filters to the same filter group.
- CMs can be associated with one upstream and one downstream filter group.
  - Upstream traffic—All traffic coming from CMs is evaluated against the assigned upstream filter group that is configured by the **cable submgmt default filter-group cm upstream** command.
  - Downstream traffic—All traffic going to CMs is evaluated against the assigned downstream filter group that is configured by the **cable submgmt default filter-group cm downstream** command.
- CPEs can be associated with one upstream and one downstream filter group.
  - Upstream traffic—All traffic coming from CPEs is evaluated against the assigned upstream filter group that is configured by the **cable submgmt default filter-group cpe upstream** command.
  - Downstream traffic—All traffic going to CPEs is evaluated against the assigned downstream filter group that is configured by the **cable submgmt default filter-group cpe downstream** command.

Note Because TLVs 35, 36, and 37 do not apply to DOCSIS 1.0 CM configuration files, the only way to enable cable subscriber management for a DOCSIS 1.0 CM is to configure it explicitly on the Cisco CMTS router and activate it by using the **cable submgmt default active** global configuration command.

Before you begin

You must create the cable filter group before you assign it to a CM or CPE upstream or downstream.

Restriction

- Chained IPv6 headers are not supported.
- An individual filter group index cannot be configured to support both IPv4 and IPv6 versions at the same time. If you need to support IPv4 and IPv6 filters for the same filter group, then you must use a separate index number with the same filter group ID, and configure one index as **ip-version ipv4**, and the other index as **ip-version ipv6**.
- Only a single upstream and a single downstream filter group can be assigned for CM traffic.
- Only a single upstream and a single downstream filter group can be assigned to CPEs attached to a CM such that all CPEs behind a CM share a common filter group.
- For the filter group to work for CMs, a CM must re-register after the CMTS router is configured for the filter group.
- If parallel eXpress forwarding (PXF) is configured on the Cisco uBR10012 router, either the **cable filter group** commands or the interface ACL (**ip access-list**) command can be configured.
- If you do not provision TLVs 35, 36, and 37 in the DOCSIS CM configuration file, then you must activate the functionality by specifying the **cable submgmt default active** global configuration command on the CMTS router.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable filter group***group-id* **index***index-num***dest-port***port-num* **Example:** `Router(config)# cable filter group 1 index 1 dest-port 69`	(Optional) Specifies the TCP/UDP destination port number that should be matched. The valid range is from 0 to 65535. The default value matches all TCP/UDP port numbers (IPv4 and IPv6 filters).
Step 4	**cable filter group** *group-id* **index** *index-num* **ip-proto** *proto-type* **Example:** `Router(config)# cable filter group 1 index 1 ip-proto 17`	(Optional) Specifies the IP protocol type number that should be matched. The valid range is from 0 to 256, with a default value of 256 that matches all protocols (IPv4 and IPv6 filters). Some commonly used values are:
Step 5	**cable filter group** *group-id* **index** *index-num* **ip-tos** *tos-mask tos-value* **Example:** `Router(config)# cable filter group 1 index 1 ip-tos 0xff 0x80`	(Optional) Specifies a ToS mask and value to be matched (IPv4 and IPv6 filters): The *tos-mask* is logically ANDed with the *tos-value* and compared to the result of ANDing the *tos-mask* with the actual ToS value of the packet. The filter considers it a match if the two values are the same. The default values for both parameters matches all ToS values.
Step 6	**cable filter group** *group-id* **index** *index-num* **ip-version ipv6** **Example:** `Router(config)# cable filter group 1 index 1 ip-version ipv6`	Specifies that this filter group is an IPv6 filter group.
Step 7	**cable filter group** *group-id* **index** *index-num* **match-action** {**accept** | **drop**} **Example:**	(Optional) Specifies the action that should be taken for packets that match this filter (IPv4 and IPv6 filters):

	Command or Action	Purpose
	`Router(config)# cable filter group 1 index 1 match-action drop`	
Step 8	**cable filter group** *group-id* **index** *index-num* **src-port** *port-num* **Example:** `Router(config)# cable filter group 1 index 1 src-port 50`	(Optional) Specifies the TCP/UDP source port number that should be matched. The valid range is from 0 to 65535. The default value matches all TCP/UDP port numbers (IPv4 and IPv6 filters).
Step 9	**cable filter group** *group-id* **index** *index-num* **status** {**active** | **inactive**} **Example:** `Router(config)# cable filter group 1 index 1 status inactive`	(Optional) Enables or disables the filter (IPv4 and IPv6 filters): **Note** You must create a filter group using at least one of the other options before you can use this command to enable or disable the filter.
Step 10	**cable filter group** *group-id* **index** *index-num* **tcp-flags** *flags-mask flags-value* **Example:** `Router(config)# cable filter group 1 index 1 tcp-flags 0 0`	(Optional) Specifies the TCP flag mask and value to be matched (IPv4 and IPv6 filters):
Step 11	**cable filter group** *group-id* **index** *index-num* **v6-dest-address** *ipv6-address* **Example:** `Router(config)# cable filter group 1 index 1 v6-dest-address 2001:DB8::/32`	(Optional) Specifies the IPv6 destination address that should be matched using the format X:X:X:X::X (IPv6 filters only).
Step 12	**cable filter group** *group-id* **index** *index-num* **v6-dest-pfxlen** *prefix-length* **Example:** `Router(config)# cable filter group 1 index 1 v6-dest-pfxlen 64`	(Optional) Specifies the length of the network portion of the IPv6 destination address. The valid range is from 0 to 128.
Step 13	**cable filter group** *group-id* **index** *index-num* **v6-src-address** *ipv6-address* **Example:** `Router(config)# cable filter group 1 index 1 v6-src-address 2001:DB8::/32`	(Optional) Specifies the IPv6 source address that should be matched using the format X:X:X:X::X (IPv6 filters only).
Step 14	**cable filter group** *group-id* **index** *index-num* **v6-src-pfxlen** *prefix-length* **Example:**	(Optional) Specifies the length of the network portion of the IPv6 source address. The valid range is from 0 to 128 (IPv6 filters only).

	Command or Action	Purpose
	`Router(config)# cable filter group 1 index 1 v6-src-pfxlen 48`	
Step 15	**cable submgmt default filter-group** {**cm** | **cpe**} {**downstream** | **upstream**} *group-id* **Example:** `Router(config)# cable submgmt default filter-group cm upstream 1`	Applies a defined filter group (by specifying its *group-id)* to either a CM or its CPE devices, for downstream or upstream traffic.
Step 16	**cable submgmt default active** **Example:** `Router(config)# cable submgmt default active`	(Required if you do not provision TLVs 35, 36, and 37 in the DOCSIS 1.1 CM configuration file) Enables filters and allows the CMTS to manage the CPE devices for a particular CM (sets the docsSubMgtCpeActiveDefault attribute to TRUE).

Example

The following example shows how to create an IPv6 filter group with ID 254 and an index number of 128. The **ip-version ipv6** keywords must be configured to create the IPv6 filter group; otherwise, the default is an IPv4 filter group:

```
configure terminal
cable filter group 254
 index 128 v6-src-address 2001:DB8::/32
cable filter group 254
 index 128 v6-src-pfxlen 48
cable filter group 254
 index 128 v6-dest-address 2001:DB8::/32
cable filter group 254
 index 128 v6-dest-pfxlen 64
cable filter group 254
 index 128 ip-version ipv6
cable filter group 254
 index 128 match-action drop
cable submgmt default filter-group cm upstream 254
```

This group filters CM upstream traffic and drops any packets with an IPv6 source address of 2001:33::20B:BFFF:FEA9:741F (with network prefix of 128) destined for an IPv6 address of 2001:DB8::/32 (with network prefix of 128).

All of the **cable filter group** commands are associated by their group ID of 254 (and index of 128), and the **cable submgmt default filter-group** command applies the corresponding filter group ID of 254 to CM upstream traffic.

To monitor your cable filter group configuration, use forms of the **show cable filter** command as shown in the following examples. In these output examples, the output from the **show cable filter**, **show cable filter group 254**, and **show cable filter group 254 index 128** commands all display the same information because there is currently only a single filter group and index defined.

Note The "Use Verbose" string appears in the output area of the SrcAddr/mask and DestAddr/Mask fields suggesting use of the **show cable filter group verbose** form of the command to display the complete IPv6 address.

```
Router# show cable filter
Filter    SrcAddr/Mask       DestAddr/Mask      Prot ToS  SPort DPort TCP   Action Status
Grp Id v6                                                             Flags
254 128Y  Use Verbose
        Use Verbose
                                  drop   active
Router# show cable filter group 254
Filter    SrcAddr/Mask       DestAddr/Mask      Prot ToS  SPort DPort TCP   Action Status
Grp Id v6                                                             Flags
254 128Y  Use Verbose        Use Verbose                                    drop   active
Router# show cable filter group 254 index 128
Filter    SrcAddr/Mask       DestAddr/Mask      Prot ToS  SPort DPort TCP   Action Status
Grp Id v6                                                             Flags
254 128Y  Use Verbose        Use Verbose                                    drop   active
Router# show cable filter group 254 index 128 verbose
Filter Group                          : 254
Filter Index                          : 128
Filter Version                        : IPv6
Matches                               : 0
    Source IPv6 address               : 2001:DB8::/32
    Destination IPv6 address          : 2001:DB8::/32
    Match action                      : drop
    Status                            : active
```

Troubleshooting Tips

You should configure the **cable filter group** commands prior to applying a filter group using the **cable submgmt default filter-group** command. Failure to do so results in the following message, and an association to a filter group that is undefined:

```
Router(config)# cable submgmt default filter-group cm upstream 100
Default value set to a nonexistent filter-group 100.
```

Configuring IPv6 Domain Name Service

Cisco IOS releases support the domain name service (DNS) capability for devices using IPv6 addressing on the Cisco CMTS routers.

DNS simplifies the identification of cable devices by associating a hostname with what can often be a complex 128-bit IPv6 address. The hostname can then be used in place of the IPv6 address within the CMTS router CLI that supports use of hostnames.

There are two separate DNS caches supported on a CMTS router—an IOS DNS cache and a cable-specific DNS cache that stores IPv6 addresses learned by the CMTS router for CMs and CPEs.

In this phase of the IPv6 DNS service on cable, the DNS server is queried for domain name information as needed when you use the **show cable modem domain-name** command. When you use this command, the following actions take place:

1. The CMTS router checks whether CMs are online. If a CM is online, the CMTS router uses the corresponding IPv6 address assigned to the CM and looks up its domain name from the IOS DNS cache.
2. If no match is found, the CMTS router sends a DNS-QUERY message with the IPv6 address of the CM to the DNS server, which tries to resolve the domain name.
3. When the DNS reply is received, the CMTS router stores the domain name in the IOS DNS cache for each IPv6 address.
4. The CMTS router also stores the fully-qualified domain name (FQDN) that is replied by the DNS server in the cable-specific DNS cache.

Note Running the **no ip domain lookup** command turns off the DNS resolution.

The following platform-independent Cisco IOS-xe software commands are supported using host names by the CMTS router for IPv6 DNS on cable:

- **connect**
- **ping ipv6**
- **show hosts**
- **telnet**
- **traceroute**

Before you begin

- A DNS server must be configured.
- You must identify and assign the host names to the IPv6 addresses. If you are using the Cisco DNS server, use the **ip host** global configuration command to map hostnames to IP addresses.
- You must configure the DNS server using the **ip name-server** global configuration command before use of DNS host names (or domains) are available in the supported commands.
- The **show cable modem domain-name** command must be run first on the Route Processor (RP) of the CMTS router before any domain name can be used as part of a cable command.

Restriction

- DNS for cable devices using IPv4 addressing is not supported.
- Due to column size limitations within the command-line interface (CLI), the domain name display is limited to 32 characters. Therefore, the entire domain name cannot always be seen in CMTS router command output.
- Only those cable devices where IPv6 address learning takes place are supported, such as acquiring an IPv6 address through DHCPv6 or the IPv6 (ND) process.
- The cable-specific DNS cache is only updated when you use the **show cable modem domain-name** command on the Route Processor (RP). A DNS-QUERY can only be sent on the RP using this command, therefore the DNS cache cannot update if you use the **show cable modem domain-name** command on a line card console. The output is displayed on the RP only.
- The cable-specific DNS cache does not store partially qualified domain names, only FQDNs are stored.
- The cable-specific DNS cache is not associated with the timeouts that apply to the IOS DNS cache. Therefore, a cable-specific DNS cache entry is not removed when an IOS DNS cache timeout occurs for that device. The cable-specific DNS cache is only updated when you use the **show cable modem domain-name** command.
- The CMTS router supports storage of only one domain name per IPv6 address in the cable-specific DNS cache.
- Domain names for the link local address are not supported.
- The **no ip domain-name** command disables DNS lookup.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip name-server** [**vrf** *vrf-name*] *server-address1* [*server-address2...server-address6*] **Example:** `Router(config)# ip name-server 2001:DB8::/32`	Specifies the address of one or more name servers to use for name and address resolution.
Step 4	**exit** **Example:** `Router(config)# exit`	Leaves global configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 5	**show cable modem domain-name** **Example:** `Router# show cable modem domain-name`	Updates the cable-specific DNS cache and displays the domain name for all CMs and the CPE devices behind a CM.

Configuring IPv6 Source Verification

Typically, the IPv6 source verification feature is enabled on a cable bundle interface. From there, the cable interface is associated with the virtual bundle interface to acquire its configuration.

When you enable IPv6 source verification on a cable line card interface, the source verification routine verifies the MAC address-MD-SID-IP binding of the packet. If the source verification succeeds, the packet is forwarded. If the verification fails, the packet is dropped.

When a CM is operating as a bridge modem device, then the CMTS router verifies all the IPv6 addresses related to that CM and the CPEs behind that CM.

The **cable ipv6 source-verify** command controls only the source verification of IPv6 packets. For IPv4-based source verification, use the **cable source-verify** command, which also supports different options.

For more information about how to configure IPv6 source verification on a bundle interface, see the Configuring the Cable Virtual Bundle Interface, on page 666.

Restrictions

Source verification of IPv6 packets occurs only on packets in the process-switched path of the Route Processor (RP).

Configuring IPv6 VPN over MPLS

The Cisco CMTS routers support the IPv6 VPN over MPLS (6VPE) feature. Implementing this feature includes the following configuration tasks.

- Configuring a VRF instance for IPv6
- Binding a VRF to an interface
- Creating a subinterface
- Configuring a static route for PE-to-CE-routing
- Configuring eBGP PE-to-CE routing sessions
- Configuring the IPv6 VPN address family for iBGP
- Configuring route reflectors for improved scalability
- Configuring Internet access

For detailed information about the configuration examples, see Configuration Examples for IPv6 on Cable, on page 681.

Note The IPv6 address of the sub-bundle interface (to which the CM is connected) is used in the DHCPv6 relay packet of the CPE DHCPv6 request. If the DHCPv6 packet has to go from one VRF interface to another, the IPv6 address of each VRF interface should be configured on the Cisco CMTS to establish connectivity.

Configuring DHCPv6 Relay Agent

The Cisco CMTS router supports DHCPv6 relay agent to forward relay-forward messages from a specific source address to client relay destinations.

Perform the steps given below to enable the DHCPv6 relay agent function and specify relay destination addresses on an interface.

Before you begin

The relay-forward messages should contain specific source IPv6 address. This is required because the firewall deployed between the Cisco CMTS DHCPv6 relay agent and the DHCPv6 server expects only one source address for one Cisco CMTS bundle interface.

Restriction If you change one or more parameters of the **ipv6 dhcp relay destination** command, you have to disable the command using the **no** form, and execute the command again with changed parameters.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface type** *number* **Example:** `Router(config)# interface ethernet 4/2`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	**ipv6 dhcp relay destination** *ipv6-address*[*interface*] [**link-address** *link-address*] [**source-address** *source-address*] **Example:** `Router(config-if) ipv6 dhcp relay`	Specifies a destination address to which client packets are forwarded and enables DHCPv6 relay service on the interface.

	Command or Action	Purpose
	`destination 2001:db8:1234::1 ethernet 4/2 link-address 2001:db8::1 source-address 2001:db8::2`	
Step 5	**end** **Example:** `Router(config-if) end`	Exits interface configuration mode and enters privileged EXEC mode.

Disabling IPv6 ND Gleaning

You must disable IPv6 ND gleaning before configuring IPv6 source verification using DHCPv6 leasequery.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interfacebundle** *bundle-no* **Example:** `Router(config)# interface bundle 1`	Specifies a bundle interface number and enters bundle interface configuration mode. • *bundle-no* —Bundle interface number. The valid range is from 1 to 255.
Step 4	**no cable nd** **Example:** `Router(config-if) no cable nd`	Disables IPv6 ND gleaning on the Cisco CMTS router.
Step 5	**end** **Example:** `Router(config-if) end`	Returns to privileged EXEC mode.

How to Verify IPv6 Dual Stack CPE Support

This section describes how to use **show** commands to verify the configuration of the IPv6 Dual Stack CPE Support on the CMTS feature.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**show cable modem** [*ip-address* | *mac-address*] **ipv6**[**cpe** | **prefix** | **registered** | **unregistered]**
	Example:
	`Router# show cable modem ipv6 registered`
	Example:
	`Router# show cable modem 0019.474a.c14a ipv6 cpe`
		Displays IPv6 information for specified CMs and CPEs behind a CM on a Cisco CMTS router. You can specify the following options:

Step 3	**show cable modem** [ip-address | mac-address] **registered** **Example:** `Router# show cable modem 0019.474e.e4DF registered`	Displays a list of the CMs that have registered with the Cisco CMTS. You can specify the following options:
Step 4	**show cable modem** {ip-address | mac-address} **cpe** **Example:** `Router# show cable modem 0019.474a.c14a cpe`	Displays the CPE devices accessing the cable interface through a particular CM. You can specify the following options:

Examples

Use the **show cable modem ipv6** command to display the IPv6 portion of a dual stack CPE and use the **show cable modem cpe** command to display the IPv4 mode of a dual stack CPE. Both **show cable modem ipv6 registered** and **show cable modem registered** commands display CPE count as one for a dual stack CPE.

The following example shows the output of the **show cable modem ipv6** command:

```
Router# show cable modem ipv6 registered
Interface   Prim Online       CPE IP Address                                MAC Address
            Sid  State
C4/0/U2     1    online       0   ---                                       0019.474a.c18c
C4/0/U2     3    online(pt)   1   2001:420:3800:809:EDA4:350C:2F75:4779   0019.474a.c14a
Router# show cable modem 0019.474a.c14a ipv6 cpe
MAC Address    IP Address                          Domain Name
0005.0052.2c1d 2001:420:3800:809:48F7:3C33:B774:9185
```

The following example shows the output of the **show cable modem ipv6** command:

```
Router# show cable modem
 0023.bed9.4c8e ipv6 cpe
Load for five secs: 0%/0%; one minute: 1%; five minutes: 1%
Time source is hardware calendar, *06:37:20.439 UTC Thu Aug 2 2012
MAC Address    IP Address
0023.bed9.4c91 2001:40:3:4:200:5EB7:BB6:C759
2001:40:3:4:210:D73B:7A50:2D05
```

The following example shows the output of the **show cable modem registered** command:

```
Router# show cable modem registered
Interface   Prim Online      Timing Rec   QoS CPE IP address      MAC address
            Sid  State       Offset Power
C4/0/U2     3    online      1022   0.00  2   1   50.3.37.12      0019.474a.c14a
```

The following example shows the output of the **show cable modem cpe** command:

```
Router# show cable modem 0019.474a.c14a cpe

IP address        MAC address      Dual IP
 50.3.37.3 0005.0052.2c1d Y
```

Configuration Examples for IPv6 on Cable

This section includes the following examples:

Example: IPv6 over Subinterfaces

The following example shows the CMTS bundle configuration that can be used with subinterfaces:

```
Router# show cable modem ipv6
Device Type: B - CM Bridge, R - CM Router
IP Assignment Method: D - DHCP
MAC Address    Type Interface    Mac State    D/IP IP Address
0019.474a.c18c B/D  C4/0/U2      online        Y  2001:420:3800:809:4C7A:D518:91
C6:8A18
Router# show run interface bundle2
Building configuration...
Current configuration : 138 bytes
!
interface Bundle2
 no ip address
 cable arp filter request-send 3 2
 cable arp filter reply-accept 3 2
 no cable ip-multicast-echo
end
Router#

show run interface bundle2.1
Building configuration...
Current configuration : 382 bytes
!
interface Bundle2.1
 ip address 50.3.37.1 255.255.255.0
 no cable ip-multicast-echo
 cable helper-address 10.10.0.12
 ipv6 address 2001:DB8::/32
 ipv6 enable
```

```
 ipv6 nd prefix default no-advertise
 ipv6 nd managed-config-flag
 ipv6 nd other-config-flag
 ipv6 nd ra interval msec 2000
 ipv6 dhcp relay destination 2001:420:3800:800:203:BAFF:FE11:B644
 arp timeout 240
end
```

Example: Basic IPv6 Cable Filter Groups

The following example shows the configuration of an IPv6 filter group that drops traffic from a specific IPv6 host (with source address 2001:DB8::1/48) behind a cable router to an IPv6 host on the network (with destination address 2001:DB8::5/64):

```
configure terminal
!
! Specify the filter group criteria using a common group ID
!
cable filter group 254 index 128 v6-src-address 2001:DB8::1
cable filter group 254 index 128 v6-src-pfxlen 128
cable filter group 254 index 128 v6-dest-address 2001:DB8::5
cable filter group 254 index 128 v6-dest-pfxlen 128
!
! Specify that the filter group is IP version 6
!
cable filter group 254 index 128 ip-version ipv6
!
! Specify the drop action for matching packets
!
cable filter group 254 index 128 match-action drop
!
! Apply the filter group with ID 254 to all CM upstream traffic
!
cable submgmt default filter-group cm upstream 254
```

Example: Complete Cable Configuration with IPv6

The following example shows a complete cable configuration example; it also displays the configuration of multiple cable filter groups using both IPv4 and IPv6 and separate indexes to associate the filter definitions with the same group ID.

```
Router# show running-config
Building configuration...
Current configuration : 15010 bytes
!
! Last configuration change at 08:32:14 PST Thu Nov 8 2007
!
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service compress-config
!
hostname router
!
boot-start-marker
```

```
boot-end-marker
!
enable password password1
!
no aaa new-model
clock timezone PST -9
clock summer-time PDT recurring
clock calendar-valid
facility-alarm core-temperature major 53
facility-alarm core-temperature minor 45
facility-alarm core-temperature critical 85
facility-alarm intake-temperature major 49
facility-alarm intake-temperature minor 40
facility-alarm intake-temperature critical 67
!
!
card 1/0 2jacket-1
card 1/0/0 24rfchannel-spa-1
card 5/0 5cable-mc520h-d
cable admission-control preempt priority-voice
cable modem vendor 00.18.68 SA-DPC2203
cable modem vendor 00.19.47 SA-DPC2505
no cable qos permission create
no cable qos permission update
cable qos permission modems
!
cable filter group 1 index 1 src-ip 0.0.0.0
cable filter group 1 index 1 src-mask 0.0.0.0
cable filter group 1 index 1 dest-ip 0.0.0.0
cable filter group 1 index 1 dest-mask 0.0.0.0
cable filter group 2 index 1 src-ip 0.0.0.0
cable filter group 2 index 1 src-mask 0.0.0.0
cable filter group 2 index 1 dest-ip 0.0.0.0
cable filter group 2 index 1 dest-mask 0.0.0.0
cable filter group 3 index 1 src-ip 0.0.0.0
cable filter group 3 index 1 src-mask 0.0.0.0
cable filter group 3 index 1 dest-ip 0.0.0.0
cable filter group 3 index 1 dest-mask 0.0.0.0
cable filter group 4 index 1 src-ip 0.0.0.0
cable filter group 4 index 1 src-mask 0.0.0.0
cable filter group 4 index 1 dest-ip 0.0.0.0
cable filter group 4 index 1 dest-mask 0.0.0.0
cable filter group 5 index 1 src-ip 0.0.0.0
cable filter group 5 index 1 src-mask 0.0.0.0
cable filter group 5 index 1 dest-ip 0.0.0.0
cable filter group 5 index 1 dest-mask 0.0.0.0
cable filter group 6 index 1 src-ip 0.0.0.0
cable filter group 6 index 1 src-mask 0.0.0.0
cable filter group 6 index 1 dest-ip 0.0.0.0
cable filter group 6 index 1 dest-mask 0.0.0.0
cable filter group 7 index 1 src-ip 0.0.0.0
cable filter group 7 index 1 src-mask 0.0.0.0
cable filter group 7 index 1 dest-ip 0.0.0.0
cable filter group 7 index 1 dest-mask 0.0.0.0
cable filter group 8 index 1 src-ip 0.0.0.0
cable filter group 8 index 1 src-mask 0.0.0.0
cable filter group 8 index 1 dest-ip 0.0.0.0
cable filter group 8 index 1 dest-mask 0.0.0.0
cable filter group 9 index 1 src-ip 0.0.0.0
cable filter group 9 index 1 src-mask 0.0.0.0
cable filter group 9 index 1 dest-ip 0.0.0.0
cable filter group 9 index 1 dest-mask 0.0.0.0
cable filter group 10 index 1 src-ip 0.0.0.0
cable filter group 10 index 1 src-mask 0.0.0.0
```

```
cable filter group 10 index 1 dest-ip 0.0.0.0
cable filter group 10 index 1 dest-mask 0.0.0.0
cable filter group 12 index 1 src-ip 0.0.0.0
cable filter group 12 index 1 src-mask 0.0.0.0
cable filter group 12 index 1 dest-ip 0.0.0.0
cable filter group 12 index 1 dest-mask 0.0.0.0
cable filter group 16 index 1 src-ip 0.0.0.0
cable filter group 16 index 1 src-mask 0.0.0.0
cable filter group 16 index 1 dest-ip 0.0.0.0
cable filter group 16 index 1 dest-mask 0.0.0.0
ip subnet-zero
ip domain name cisco.com
ip host host1 239.192.254.254
ip host host2 239.192.254.253
ip name-server 10.39.26.7
ip name-server 2001:0DB8:4321:FFFF:0:800:20CA:D8BA
!
!
!
!
ipv6 unicast-routing
ipv6 cef
packetcable multimedia
packetcable
!
!
!
redundancy
 mode sso
!
!
controller Modular-Cable 1/0/0
 annex B modulation 64qam 0 23
 ip-address 10.30.4.175
 modular-host subslot 5/0
 rf-channel 0 cable downstream channel-id 24
 rf-channel 1 cable downstream channel-id 25
 rf-channel 2 cable downstream channel-id 26
 rf-channel 3 cable downstream channel-id 27
 rf-channel 4 cable downstream channel-id 28
 rf-channel 5 cable downstream channel-id 29
 rf-channel 6 cable downstream channel-id 30
 rf-channel 7 cable downstream channel-id 31
 rf-channel 8 cable downstream channel-id 32
 rf-channel 9 cable downstream channel-id 33
 rf-channel 10 cable downstream channel-id 34
 rf-channel 11 cable downstream channel-id 35
 rf-channel 12 cable downstream channel-id 36
 rf-channel 13 cable downstream channel-id 37
 rf-channel 14 cable downstream channel-id 38
 rf-channel 15 cable downstream channel-id 39
 rf-channel 16 cable downstream channel-id 40
 rf-channel 17 cable downstream channel-id 41
 rf-channel 18 cable downstream channel-id 42
 rf-channel 19 cable downstream channel-id 43
 rf-channel 20 cable downstream channel-id 44
 rf-channel 21 cable downstream channel-id 45
 rf-channel 22 cable downstream channel-id 46
 rf-channel 23 cable downstream channel-id 47
!
!
policy-map foo
policy-map 1
policy-map cos
```

```
policy-map qpolicy
policy-map shape
policy-map dscp
!
!
!
!
!
!
interface Loopback0
 ip address 127.0.0.1 255.255.255.255
!
interface FastEthernet0/0/0
 ip address 10.39.21.10 255.255.0.0
 speed 100
 half-duplex
 ipv6 address 2001:DB8::/32
 ipv6 enable
!
interface Wideband-Cable1/0/0:0
 no cable packet-cache
 cable bonding-group-id 1
!
interface Wideband-Cable1/0/0:1
 no cable packet-cache
 cable bonding-group-id 2
!
interface Wideband-Cable1/0/0:2
 no cable packet-cache
 cable bonding-group-id 3
!
interface Wideband-Cable1/0/0:3
 no cable packet-cache
 cable bonding-group-id 4
!
interface Wideband-Cable1/0/0:4
 no cable packet-cache
 cable bundle 1
 cable bonding-group-id 5
 cable rf-channel 1 bandwidth-percent 60
!
interface Wideband-Cable1/0/0:5
 no cable packet-cache
 cable bundle 1
 cable bonding-group-id 6
 cable rf-channel 0 bandwidth-percent 40
 cable rf-channel 2
 cable rf-channel 3
!
interface Wideband-Cable1/0/0:6
 no cable packet-cache
 cable bonding-group-id 7
!
interface Wideband-Cable1/0/0:7
 no cable packet-cache
 cable bonding-group-id 8
!
interface Wideband-Cable1/0/0:8
 no cable packet-cache
 cable bonding-group-id 9
!
interface Wideband-Cable1/0/0:9
 no cable packet-cache
 cable bonding-group-id 33
```

```
!
interface Wideband-Cable1/0/0:10
 no cable packet-cache
 cable bonding-group-id 34
!
interface Wideband-Cable1/0/0:11
 no cable packet-cache
 cable bonding-group-id 35
!
interface Cable5/0/0
 no cable packet-cache
 cable bundle 1
 cable downstream channel-id 119
 cable downstream annex B
 cable downstream modulation 256qam
 cable downstream interleave-depth 32
 cable downstream frequency 99000000
 no cable downstream rf-shutdown
 cable upstream max-ports 4
 cable upstream 0 connector 0
 cable upstream 0 frequency 6000000
 cable upstream 0 ingress-noise-cancellation 200
 cable upstream 0 docsis-mode tdma
 cable upstream 0 channel-width 1600000 1600000
 cable upstream 0 minislot-size 4
 cable upstream 0 range-backoff 3 6
 cable upstream 0 modulation-profile 21
 no cable upstream 0 shutdown
 cable upstream 1 connector 1
 cable upstream 1 ingress-noise-cancellation 200
 cable upstream 1 docsis-mode tdma
 cable upstream 1 channel-width 1600000 1600000
 cable upstream 1 minislot-size 4
 cable upstream 1 range-backoff 3 6
 cable upstream 1 modulation-profile 21
 cable upstream 1 shutdown
 cable upstream 2 connector 2
 cable upstream 2 ingress-noise-cancellation 200
 cable upstream 2 docsis-mode tdma
 cable upstream 2 channel-width 1600000 1600000
 cable upstream 2 minislot-size 4
 cable upstream 2 range-backoff 3 6
 cable upstream 2 modulation-profile 21
 cable upstream 2 shutdown
 cable upstream 3 connector 3
 cable upstream 3 ingress-noise-cancellation 200
 cable upstream 3 docsis-mode tdma
 cable upstream 3 channel-width 1600000 1600000
 cable upstream 3 minislot-size 4
 cable upstream 3 range-backoff 3 6
 cable upstream 3 modulation-profile 21
 cable upstream 3 shutdown
!
interface Cable5/0/1
 cable ip-init ipv6
 no cable packet-cache
 cable bundle 1
 cable downstream channel-id 120
 cable downstream annex B
 cable downstream modulation 64qam
 cable downstream interleave-depth 32
 cable downstream frequency 705000000
 no cable downstream rf-shutdown
 cable upstream max-ports 4
```

```
 cable upstream 0 connector 4
 cable upstream 0 frequency 6000000
 cable upstream 0 ingress-noise-cancellation 200
 cable upstream 0 docsis-mode tdma
 cable upstream 0 channel-width 1600000 1600000
 cable upstream 0 minislot-size 4
 cable upstream 0 range-backoff 3 6
 cable upstream 0 modulation-profile 21
 no cable upstream 0 shutdown
 cable upstream 1 connector 5
 cable upstream 1 ingress-noise-cancellation 200
 cable upstream 1 docsis-mode tdma
 cable upstream 1 channel-width 1600000 1600000
 cable upstream 1 minislot-size 4
 cable upstream 1 range-backoff 3 6
 cable upstream 1 modulation-profile 21
 cable upstream 1 shutdown
 cable upstream 2 connector 6
 cable upstream 2 ingress-noise-cancellation 200
 cable upstream 2 docsis-mode tdma
 cable upstream 2 channel-width 1600000 1600000
 cable upstream 2 minislot-size 4
 cable upstream 2 range-backoff 3 6
 cable upstream 2 modulation-profile 21
 cable upstream 2 shutdown
 cable upstream 3 connector 7
 cable upstream 3 ingress-noise-cancellation 200
 cable upstream 3 docsis-mode tdma
 cable upstream 3 channel-width 1600000 1600000
 cable upstream 3 minislot-size 4
 cable upstream 3 range-backoff 3 6
 cable upstream 3 modulation-profile 21
 cable upstream 3 shutdown
!
interface Cable5/0/2
 no cable packet-cache
 cable downstream channel-id 121
 cable downstream annex B
 cable downstream modulation 64qam
 cable downstream interleave-depth 32
 cable downstream rf-shutdown
 cable upstream max-ports 4
 cable upstream 0 connector 8
 cable upstream 0 ingress-noise-cancellation 200
 cable upstream 0 docsis-mode tdma
 cable upstream 0 channel-width 1600000 1600000
 cable upstream 0 minislot-size 4
 cable upstream 0 range-backoff 3 6
 cable upstream 0 modulation-profile 21
 cable upstream 0 shutdown
 cable upstream 1 connector 9
 cable upstream 1 ingress-noise-cancellation 200
 cable upstream 1 docsis-mode tdma
 cable upstream 1 channel-width 1600000 1600000
 cable upstream 1 minislot-size 4
 cable upstream 1 range-backoff 3 6
 cable upstream 1 modulation-profile 21
 cable upstream 1 shutdown
 cable upstream 2 connector 10
 cable upstream 2 ingress-noise-cancellation 200
 cable upstream 2 docsis-mode tdma
 cable upstream 2 channel-width 1600000 1600000
 cable upstream 2 minislot-size 4
 cable upstream 2 range-backoff 3 6
```

```
 cable upstream 2 modulation-profile 21
 cable upstream 2 shutdown
 cable upstream 3 connector 11
 cable upstream 3 ingress-noise-cancellation 200
 cable upstream 3 docsis-mode tdma
 cable upstream 3 channel-width 1600000 1600000
 cable upstream 3 minislot-size 4
 cable upstream 3 range-backoff 3 6
 cable upstream 3 modulation-profile 21
 cable upstream 3 shutdown
!
interface Cable5/0/3
 no cable packet-cache
 cable downstream channel-id 122
 cable downstream annex B
 cable downstream modulation 64qam
 cable downstream interleave-depth 32
 cable downstream rf-shutdown
 cable upstream max-ports 4
 cable upstream 0 connector 12
 cable upstream 0 ingress-noise-cancellation 200
 cable upstream 0 docsis-mode tdma
 cable upstream 0 channel-width 1600000 1600000
 cable upstream 0 minislot-size 4
 cable upstream 0 range-backoff 3 6
 cable upstream 0 modulation-profile 21
 cable upstream 0 shutdown
 cable upstream 1 connector 13
 cable upstream 1 ingress-noise-cancellation 200
 cable upstream 1 docsis-mode tdma
 cable upstream 1 channel-width 1600000 1600000
 cable upstream 1 minislot-size 4
 cable upstream 1 range-backoff 3 6
 cable upstream 1 modulation-profile 21
 cable upstream 1 shutdown
 cable upstream 2 connector 14
 cable upstream 2 ingress-noise-cancellation 200
 cable upstream 2 docsis-mode tdma
 cable upstream 2 channel-width 1600000 1600000
 cable upstream 2 minislot-size 4
 cable upstream 2 range-backoff 3 6
 cable upstream 2 modulation-profile 21
 cable upstream 2 shutdown
 cable upstream 3 connector 15
 cable upstream 3 ingress-noise-cancellation 200
 cable upstream 3 docsis-mode tdma
 cable upstream 3 channel-width 1600000 1600000
 cable upstream 3 minislot-size 4
 cable upstream 3 range-backoff 3 6
 cable upstream 3 modulation-profile 21
 cable upstream 3 shutdown
!
interface Cable5/0/4
 no cable packet-cache
 cable downstream channel-id 123
 cable downstream annex B
 cable downstream modulation 64qam
 cable downstream interleave-depth 32
 cable downstream rf-shutdown
 cable upstream max-ports 4
 cable upstream 0 connector 16
 cable upstream 0 ingress-noise-cancellation 200
 cable upstream 0 docsis-mode tdma
 cable upstream 0 channel-width 1600000 1600000
```

```
 cable upstream 0 minislot-size 4
 cable upstream 0 range-backoff 3 6
 cable upstream 0 modulation-profile 21
 cable upstream 0 shutdown
 cable upstream 1 connector 17
 cable upstream 1 ingress-noise-cancellation 200
 cable upstream 1 docsis-mode tdma
 cable upstream 1 channel-width 1600000 1600000
 cable upstream 1 minislot-size 4
 cable upstream 1 range-backoff 3 6
 cable upstream 1 modulation-profile 21
 cable upstream 1 shutdown
 cable upstream 2 connector 18
 cable upstream 2 ingress-noise-cancellation 200
 cable upstream 2 docsis-mode tdma
 cable upstream 2 channel-width 1600000 1600000
 cable upstream 2 minislot-size 4
 cable upstream 2 range-backoff 3 6
 cable upstream 2 modulation-profile 21
 cable upstream 2 shutdown
 cable upstream 3 connector 19
 cable upstream 3 ingress-noise-cancellation 200
 cable upstream 3 docsis-mode tdma
 cable upstream 3 channel-width 1600000 1600000
 cable upstream 3 minislot-size 4
 cable upstream 3 range-backoff 3 6
 cable upstream 3 modulation-profile 21
 cable upstream 3 shutdown
!
interface Bundle1
 ip address 10.46.2.1 255.255.0.0 secondary
 ip address 10.46.1.1 255.255.0.0
 cable arp filter request-send 3 2
 cable arp filter reply-accept 3 2
 cable dhcp-giaddr policy strict
 cable helper-address 10.39.26.8
 ipv6 address 2001:DB8::/32
 ipv6 enable
 ipv6 nd managed-config-flag
 ipv6 nd other-config-flag
 ipv6 nd ra interval 5
 ipv6 dhcp relay destination 2001:0DB8:4321:FFFF:0:800:20CA:D8BA
!
ip default-gateway 10.39.0.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.39.26.12
ip route 192.168.254.253 255.255.255.255 10.39.0.1
ip route 192.168.254.254 255.255.255.255 10.39.0.1
!
!
no ip http server
no ip http secure-server
!
logging cmts cr10k log-level errors
cpd cr-id 1
nls resp-timeout 1
cdp run
!
tftp-server bootflash:docs10.cm alias docs10.cm
tftp-server bootflash:rfsw_x373.bin alias rfsw_x373.bin
snmp-server community private RW
snmp-server enable traps cable
snmp-server manager
!
```

```
!
control-plane
!
!
line con 0
 logging synchronous
 stopbits 1
line aux 0
line vty 0 4
 password lab
 login
!
!
cable fiber-node 1
  downstream Modular-Cable 1/0/0 rf-channel 1
  upstream Cable 5/0 connector 0
!
cable fiber-node 2
  downstream Modular-Cable 1/0/0 rf-channel 0 2-3
  upstream Cable 5/0 connector 4
!
end
```

Example: BGP Configuration for 6VPE

The following example shows a sample BGP configuration on CMTS 6VPE.

```
Router# router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor 11.1.1.5 remote-as 1
 neighbor 11.1.1.5 update-source Loopback1
 no auto-summary
 !
 address-family vpnv6              --- Enable vpnv6 AF
  neighbor 11.1.1.5 activate      --- Activate neighbor 6VPE-2
  neighbor 11.1.1.5 send-community extended
 exit-address-family
 !
 address-family ipv6 vrf vrf_mgmt
  redistribute connected            ---- Publish directly connected route
  redistribute static
  no synchronization
 exit-address-family
!
 address-family ipv6 vrf vrfa   --- Enable IPv6 vrf AF for each VRF
  redistribute connected
  no synchronization
 exit-address-family
 !
 address-family ipv6 vrf vrfb --- Enable IPv6 vrf AF for each VRF
  redistribute connected
  no synchronization
 exit-address-family
!
```

Example: Subinterface Configuration for 6VPE

The following example shows how to define a subinterface on virtual bundle interface 1.

When configuring IPv6 VPNs, you must configure the first subinterface created as a part of the management VRF. In the following example, Bundle 1.10 is the first sub-interface, which is configured into management VRF. Make sure the CNR server is reachable in management VRF.

```
interface Bundle1.10                --- Management VRF
 vrf forwarding vrf_mgmt
 cable dhcp-giaddr primary
 ipv6 address 2001:40:3:110::1/64
 ipv6 enable
 ipv6 nd managed-config-flag
 ipv6 nd other-config-flag
 ipv6 dhcp relay destination 2001:10:74:129::2
interface Bundle1.11          --- VRF A
 vrf forwarding vrfa
 cable dhcp-giaddr primary
 ipv6 address 2001:40:3:111::1/64
 ipv6 enable
 ipv6 dhcp relay destination 2001:10:74:129::2
interface Bundle1.12          --- VRFB
 vrf forwarding vrfb
 cable dhcp-giaddr primary
 ipv6 address 2001:40:3:112::1/64
 ipv6 enable
 ipv6 dhcp relay destination 2001:10:74:129::2
```

Example: Cable Interface Bundling

The following example shows how to bundle a group of physical interfaces.

```
int C5/0/4 and int c5/0/3 are bundled.
int c5/0/4
cable bundle 1
int c5/0/3
cable bundle 1
```

Example: VRF Configuration for 6VPE

The following example shows how to create VRFs for each VPN.

```
vrf definition vrf_mgmt
 rd 1:1
 !
 address-family ipv4
 route-target export 1:1
 route-target import 1:1
 route-target import 2:2
 route-target import 2:1
 exit-address-family
 !
 address-family ipv6
 route-target export 1:1
 route-target import 1:1
 route-target import 2:1  -- import route of vrfa
 route-target import 2:2  -- import route of vrfb
 exit-address-family
```

Verifying IPv6 on Cable

This section explains how to verify IPv6 on cable configuration and it contains the following topics:

Verifying IPv6 VRF Configuration

To verify the IPv6 VRF configuration, use the show vrf ipv6 command in privileged EXEC mode.

```
Router# show vrf ipv6 vrfa
  Name                             Default RD          Protocols   Interfaces
  vrfa                             2:1                 ipv4,ipv6   Bu1.11
Router# show vrf ipv6 interfaces
Interface              VRF                              Protocol   Address

Bu1.10                 vrf_mgmt                          up         2001:40:3:110::1

Fa0/0/0                vrf_mgmt                          up         2001:20:4:1::38

Bu1.11                 vrfa                              up         2001:40:3:111::1

Bu1.12                 vrfb                              up         2001:40:3:112::1

CMTS#
```

Verifying IPv6 BGP Status

To verify the IPv6 BGP status, use the show ip bgp command in privileged EXEC mode.

```
Router# show ip bgp vpnv6 unicast all neighbors

BGP neighbor is 11.1.1.5,  remote AS 1, internal link
  BGP version 4, remote router ID 11.1.1.5
  Session state = Established, up for 00:35:52
  Last read 00:00:37, last write 00:00:14, hold time is 180, keepalive interval is 60 seconds

  BGP multisession with 2 sessions (2 established), first up for 00:40:07
  Neighbor sessions:
    2 active, is multisession capable
  Neighbor capabilities:
    Route refresh: advertised and received(new) on session 1, 2
    Address family IPv4 Unicast: advertised and received
    Address family VPNv6 Unicast: advertised and received
......
```

Verifying MPLS Forwarding Table

To verify the output of the MPLS forwarding table, use the show mpls forwarding-table command in the privileged EXEC mode.

```
Router# show mpls forwarding-table

Local  Outgoing      Prefix            Bytes Label   Outgoing   Next Hop
Label  Label or VC   or Tunnel Id      Switched      interface
......
```

```
19     No Label      2001:40:3:110::/64[V]   \                                ---Route in
vrf_mgmt
                                           0             aggregate/vrf_mgmt
21     No Label      2001:40:3:111::/64[V]   \                                ---Route in
vrfa
                                           0             aggregate/vrfa
22     No Label      2001:40:3:112::/64[V]   \                                ---Route in
vrfb
                                           0             aggregate/vrfb
......
```

Verifying IPv6 Cable Modem and its Host State

To verify IPv6 addresses and connected host states of cable modems and CPEs, use the **show interface cable modem** command in the privileged EXEC mode:

```
Router# show interface cable 7/0/0 modem ipv6
SID  Type State         IPv6 Address                              M MAC address
11   CM  online         2001:420:3800:809:3519:5F9C:B96A:D31      D 0025.2e2d.743a
11   CPE unknown        2001:420:3800:809:3DB2:8A6C:115F:41D8     D 0011.2544.f33b
```

Verifying Multiple IAPDs in a Single Advertise

To verify the multiple IPv6 prefixes assigned to devices on a network, use the show cable modem ipv6 prefix command in privileged EXEC mode:

```
Router# show cable modem ipv6 prefix
Load for five secs: 1%/0%; one minute: 1%; five minutes: 1%
Time source is hardware calendar, *06:36:53.075 UTC Thu Aug 2 2012
Device Type: B - CM Bridge, R - CM Router
IP Assignment Method: D - DHCP
MAC Address    Type IPv6 prefix
0023.bed9.4c91 R/D  2001:40:1012::/64
               R/D  2001:40:2012:1::/64
0000.002e.074c R/D  2001:40:1012:8::/64
               R/D  2001:40:2012:1D::/64
0000.002e.074b R/D  2001:40:1012:23::/64
               R/D  2001:40:2012:1C::/64
0000.002e.074a R/D  2001:40:1012:22::/64
               R/D  2001:40:2012:1B::/64
```

To verify the multiple IPv6 prefixes assigned to CPEs behind a CM with a specific MAC address, use the **show cable modem** *mac-address* **ipv6 prefix** command in privileged EXEC mode:

```
Router# show cable modem 0023.bed9.4c8e ipv6 prefix
Load for five secs: 0%/0%; one minute: 1%; five minutes: 1%
Time source is hardware calendar, *06:37:22.335 UTC Thu Aug 2 2012
Device Type: B - CM Bridge, R - CM Router
IP Assignment Method: D - DHCP
MAC Address    Type IPv6 prefix
0023.bed9.4c91 R/D  2001:40:1012::/64
               R/D  2001:40:2012:1::/64
```

To verify the IPv6 information of CPEs behind a CM with a specific MAC address, use the show cable modem *mac-address* ipv6 **cpe** command in privileged EXEC mode:

```
Router# show cable modem 0023.bed9.4c8e ipv6 cpe
```

```
Load for five secs: 0%/0%; one minute: 1%; five minutes: 1%
Time source is hardware calendar, *06:37:20.439 UTC Thu Aug 2 2012
MAC Address    IP Address
0023.bed9.4c91 2001:40:3:4:200:5EB7:BB6:C759
               2001:40:3:4:210:D73B:7A50:2D05
```

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

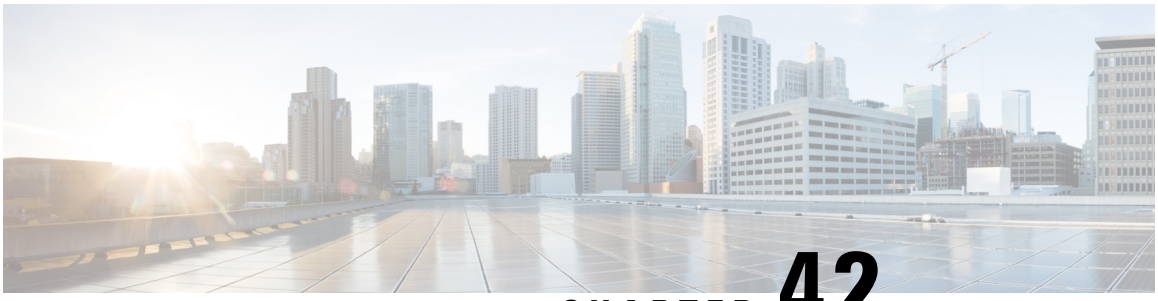
Feature Information for IPv6 on Cable

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 110: Feature Information for Downstream Interface Configuration

Feature Name	Releases	Feature Information
IPv6 on cable	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 42

Cable DHCP Leasequery

This document describes the Dynamic Host Configuration Protocol (DHCP) Leasequery feature on the Cisco cable modem termination system (CMTS) router.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 695
- Prerequisites for Cable DHCP Leasequery, on page 696
- Restrictions for Cable DHCP Leasequery, on page 696
- Information About Cable DHCP Leasequery, on page 697
- How to Configure Filtering of Cable DHCP Leasequery Requests, on page 698
- Configuration Examples for Filtering of DHCP Leasequery , on page 702
- Additional References, on page 703
- Feature Information for Cable DHCP Leasequery, on page 704

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 111: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Cable DHCP Leasequery

- You must configure a cable interface with the **cable source-verify dhcp** command and the **no cable arp** command before the Cisco CMTS router can enable DHCP Leasequery. Lease queries are sent to the DHCP server or to a configured alternate server.

To divert DHCP Leasequeries to a specific server, you must use the cable **source-verify dhcp server** ipaddress command and the **no cable arp** command before the Cisco CMTS router is enabled for DHCP Leasequery. Only one alternate server may be configured.

- You must configure the **ipv6 route** command when IPv6 Customer Premise Equipment (CPE) routers are deployed on the Cisco CMTS router.

Restrictions for Cable DHCP Leasequery

- Leasequeries are sent to the DHCP server unless an alternate server is configured.
- Only one alternate server can be configured.
- Users are responsible for the synchronization of the DHCP server and the configured alternate server.
- If the configured alternate server fails, leasequery requests are *not* returned to the DHCP server.
- Only one IA_IADDR is supported per client. If the leasequery returns multiple results, only the IA_ADDR matching the query is added to the Cisco CMTS subscriber database.

- The Cisco CMTS will not verify the source of the IPv6 link-local address of a CPE.

Information About Cable DHCP Leasequery

Problems can occur when viruses, denial of service (DoS) attacks, and theft-of-service attacks begin scanning a range of IP addresses, in an attempt to find unused addresses. When the Cisco CMTS router is verifying unknown IP addresses, this type of scanning generates a large volume of DHCP leasequeries, which can result in the following problems:

- High CPU utilization on the Cisco CMTS router PRE card.
- High utilization on the DHCP servers, resulting in a slow response time or no response at all.
- Packets can be dropped by the Cisco CMTS router or DHCP server (or configured alternate server).
- Lack of available bandwidth for other customers on the cable interface.

To prevent such a large volume of leasequery requests on cable interfaces, you can enable filtering of these requests on upstream interfaces, downstream interfaces, or both. When the Cable DHCP Leasequery feature is enabled, the Cisco CMTS allows only a certain number of DHCP leasequery requests for each service ID (SID) on an interface within the configured interval time period. If an SID generates more Leasequeries than the maximum, the router drops the excess number of requests until the next interval period begins.

You can configure both the number of allowable DHCP leasequery requests and the interval time period, so as to match the capabilities of your DHCP server (or configured alternate server) and cable network.

To configure the Cisco CMTS router to send DHCP leasequery requests to the DHCP server, use the **cable source-verify dhcp** and **no cable arp** commands. Unknown IP addresses that are found in packets for customer premises equipment (CPE) devices that use the cable modems on the cable interface are verified. The DHCP server returns a DHCP ACK message with the DHCP relay information and lease information of the CPE device that has been assigned this IP address, if any.

When **cable source-verify dhcp** and **no cable arp** commands are configured, DHCP leasequery is sent for downstream packets to verify unknown IP addresses within the IP address range configured on the cable bundle interface.

For DHCP leasequery to work in the downstream direction, the Cisco Network Registrar (CNR) should be made aware of the DHCP Option 82. This is required to make the CMTS map the CPE IP address to the correct CM. To do this, configure the **ip dhcp relay information option** command on the bundle interface to insert service class relay agent option into the DHCP DISCOVER messages. When the configuration is in place, during DHCP DISCOVER the values of DHCP Option 82 is cached by the CNR and is returned to the CMTS on any subsequent DHCP leasequery for that IP address.

To configure the Cisco CMTS router to divert DHCP leasequery requests to a server other than the DHCP server, use the **cable source-verify dhcp server ipaddress** and **no cable arp** commands.

The Cisco CMTS supports two types of DHCP leasequery implementation, Cisco standard compliant DHCP leasequery and RFC 4388 standard compliant DHCP leasequery. These two standards differ mostly in the identifiers used to query or respond to the DHCP Server. You can choose between these two implementations depending on which standard is supported on your DHCP Server.

Use the **ip dhcp compatibility lease-query client** {**cisco** | **standard**} command to configure the Cisco CMTS in either Cisco mode or RFC 4388 standard mode.

DHCP MAC Address Exclusion List

This feature enables the ability to exclude trusted MAC addresses from the standard DHCP source verification checks for the Cisco CMTS. The DHCP MAC Address Exclusion List feature enables packets from trusted MAC addresses to pass when otherwise packets would be rejected with standard DHCP source verification. This feature overrides the cable source-verify command on the Cisco CMTS for the specified MAC address, yet maintains overall support for standard and enabled DHCP source verification processes. This feature is supported on the Performance Routing Engine 1 (PRE1), PRE2, and PRE4 modules on the Cisco cBR router chassis.

To enable packets from trusted source MAC addresses in DHCP to pass without source verification checks, use the cable trust command in global configuration mode. To remove a trusted MAC address from the MAC exclusion list, use the no form of this command. Removing a MAC address from the exclusion list subjects all packets from that source to standard DHCP source verification.

For more information on the cable trust command, see the Cisco IOS CMTS Cable Command Reference Guide .

Unitary DHCPv6 Leasequery

This feature supports unitary DHCPv6 leasequery protocol (RFC 5007) on the Cisco CMTS routers for upstream IPv6 source verification. This protocol verifies the authenticity of the IPv6 CPE behind a home or small office cable deployment.

If the IPv6 source verification fails on the router and the **cable ipv6 source-verify dhcp** and **no cable nd** commands are configured on the bundle interface or subinterface, the Cisco CMTS triggers a unitary DHCPv6 leasequery to the Cisco Network Registrar (CNR). If a valid leasequery response is received from the CNR, the Cisco CMTS adds the CPE to its subscriber database and allows future traffic for the CPE.

The primary use of the unitary DHCPv6 leasequery protocol on the Cisco CMTS router is to recover lost CPE data including the Prefix Delegation (PD) route. The IPv6 CPE data can be lost from the Cisco CMTS in several ways. For example, PD route loss can occur during a Cisco CMTS reload.

The unitary DHCPv6 leasequery protocol also supports the following:

- DHCPv6 leasequery protocol.
- Rogue client database for failed source-verify clients.
- DHCPv6 leasequery filters.
- DHCPv6 leasequeries to a specific DHCPv6 server.

How to Configure Filtering of Cable DHCP Leasequery Requests

Use the following procedures to configure the filtering of DHCP Leasequery requests on the Cisco CMTS downstreams and upstreams:

Enabling DHCP Leasequery Filtering on Downstreams

Use the following procedure to start filtering DHCP leasequeries on all downstreams of a cable interface.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable source-verify leasequery-filter downstream** *threshold interval* **Example:** `Router(config)# cable source-verify leasequery-filter downstream 5 10`	Enables leasequery filtering on all downstreams on the specified bundle interface, using the specified *threshold* and *interval* values.
Step 4	**end** **Example:** `Router(config)# end`	Exits configuration mode and returns to privileged EXEC mode.

Enabling DHCP Leasequery Filtering on Upstreams

Use the following procedure to start filtering DHCP Leasequeries on all upstreams on a bundle interface.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface bundle** *bundle-no* **Example:**	Enters interface configuration mode for the specified bundle interface.

	Command or Action	Purpose
	`Router(config)# interface bundle 1`	
Step 4	**cable source-verify leasequery-filter upstream** *threshold interval* **Example:** `Router(config-if)# cable source-verify leasequery-filter upstream 2 5`	Enables leasequery filtering on all upstreams on the specified bundle interface, using the specified *threshold* and *interval* values. **Note** The **cable source-verify leasequery-filter upstream** command can only be configured under bundle interface. **Note** Repeat step 3 and step 4 to enable the filtering of DHCP Leasequeries on the upstreams for other bundle interfaces. Primary and secondary interfaces in a cable bundle must be configured separately.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Unitary DHCPv6 Leasequery Filtering

Use the following procedure to configure the Cisco CMTS router to send Leasequeries to a DHCP server to verify the authenticity of the IPv6 CPE. You can also enable filtering of these requests to prevent large volumes of Leasequery requests on the bundle interfaces. Similarly, the number of allowable Leasequery requests and the interval time period can also be configured.

Note When the leasequery timer expires, only the IPv4 static CPE is automatically removed from the host database.

Before you begin

- Disable the IPv6 Neighbor Discovery (ND) Gleaning feature using the **no** form of the **cable nd** command in bundle interface configuration mode before configuring the unitary DHCPv6 leasequery protocol. For details on IPv6 ND gleaning, see IPv6 on Cable feature guide.
- Configure the **cable ipv6 source-verify dhcp** command under the Cisco CMTS bundle or bundle subinterface to enable the unitary DHCPv6 leasequery protocol.
- Use the **cable ipv6 source-verify dhcp** [**server** *ipv6-address]* command for a single DHCP server.
- *Use the* **cable ipv6 source-verify dhcp command without any keywords** *for multiple DHCP servers.*

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface bundle** *bundle-no* **Example:** `Router(config)# interface bundle 1`	Enters interface configuration mode for the specified bundle interface.

Step 4	**cable ipv6 source-verify** or**cable ipv6 source-verify dhcp** [server ipv6-address]
	Example:
	`Router(config-if)# cable ipv6 source-verify`
	`or`
	`Router(config-if)# cable ipv6 source-verify dhcp server 2001:DB8:1::1`
		Enables leasequery filtering on the specified bundle interface and verifies the IP address with multiple DHCPv6 servers. or Enables leasequery filtering on the specified bundle interface and verifies the IP address with a specified DHCPv6 server.

Step 5	**cable ipv6 source-verify leasetimer** *value* **Example:** `Router(config-if)# cable ipv6 source-verify leasetimer 200`	Enables leasequery timer on the specified bundle interface, for the Cisco CMTS to check its internal CPE database for IPv6 addresses whose lease time has expired.
Step 6	**cable ipv6 source-verify leasequery-filter** *threshold interval* **Example:** `Router(config-if)# cable ipv6 source-verify leasetimer 5 10`	Enables filtering of the IPv6 leasequery requests.
Step 7	**end** **Example:**	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config-if)# end`	

Enabling DHCPv6 Leasequery Filtering on Downstreams

Use the following procedure to start filtering DHCP Leasequeries on all downstreams of a cable interface.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable ipv6 source-verify leasequery-filter downstream** *threshold interval* **Example:** `Router(config-if)# cable ipv6 source-verify leasetimer 5 10`	Enables leasequery filtering on all downstreams on the specified bundle interface, using the specified threshold and interval values:
Step 4	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Filtering of DHCP Leasequery

This section provides the following examples on how to configure the DHCP leasequery filtering feature:

Example: DHCP Leasequery Filtering

The following example shows an excerpt from a typical configuration of a bundle interface that is configured for filtering DHCP leasequery requests on both its upstream and downstream interfaces:

Note If an alternate server has been configured to receive leasequery requests, the **cable source-verify dhcp server** *ipaddress command* would display in place of the **cable source-verify dhcp** command below.

```
.
.
.
cable source-verify leasequery-filter downstream 5 20
.
.
.
interface bundle 1
.
.
.
 cable source-verify dhcp
 cable source-verify leasequery-filter upstream 1 5
 no cable arp
.
.
```

Example: Unitary DHCPv6 Leasequery Filtering

The following example shows how to display the total number of DHCPv6 leasequery requests that have been filtered on the router in Cisco IOS Release 12.2(33)SCF1:

```
Router# show cable leasequery-filter
IPv4 Lease Query Filter statistics for Unknown Sid
  Requests Sent : 0 total. 0 unfiltered, 0 filtered
IPv6 Lease Query Filter statistics for Unknown Sid
  Requests Sent : 0 total. 0 unfiltered, 0 filtered
```

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

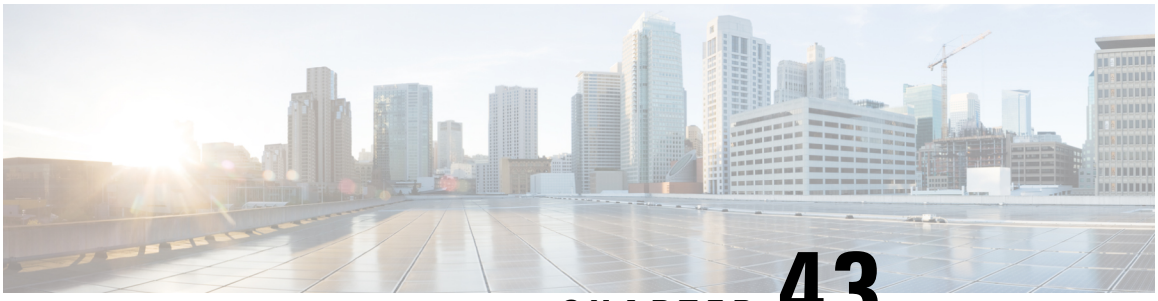
Feature Information for Cable DHCP Leasequery

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 112: Feature Information for Cable DHCP Leasequery

Feature Name	Releases	Feature Information
Cable DHCP leasequery	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 43

Layer 3 CPE Mobility

Layer 3 CPE Mobility feature is introduced to allows the mobility CPE devices to move between cable modems with as less disruption of traffic as possible.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 705
- Prerequisites for Layer 3 CPE Mobility , on page 706
- Restrictions for Layer 3 CPE Mobility , on page 706
- Information About Layer 3 CPE Mobility , on page 707
- How to Configure Layer 3 Mobility, on page 708
- Configuration Examples for Layer 3 Mobility, on page 711
- Additional References, on page 712
- Feature Information for Layer 3 CPE Mobility , on page 712

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 113: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Layer 3 CPE Mobility

No special equipment or software is needed to use the Layer 3 CPE Mobility feature.

Restrictions for Layer 3 CPE Mobility

- Layer 3 CPE Mobility feature allows CPE devices to move only in the same bundle or sub-bundle interface.
- The IPv4 or IPv6 subnets that are configured with mobility must match with the IPv4 or IPv6 subnets already configured on bundle or sub-bundle interface. Otherwise, configuration will not be accepted and the following message will be displayed:

```
Please remove the previous online CPEs or reset CMs,
```

- If you remove the IPv4 or IPv6 address on bundle or sub-bundle interface, it also removes the relative mobility subnets at the same time.
- Multicast packets will not trigger the Layer 3 CPE Mobility feature.
- VRF configured under bundle or sub-bundle interface is not supported for CPE mobility feature.
- In Layer 3 CPE Mobility feature, the packet lost time period during mobility will be unpredictable, depending on how many CPE devices move at the same time and system loading conditions.

- For CPE devices, which have multiple IPv4 or IPv6 addresses, all of IPv4 or IPv6 addresses will be rebuilt with new source information.
- Layer 3 CPE Mobility may be failed during line card or SUP HA and the trigger upstream packet will be dropped.
- If CPE mobility is turned on, mobility behavior will become effective before cable Ipv4 or IPv6 source verify.
- If Layer 3 CPE Mobility is enabled, some of the security checks will be skipped for the mobility subnets to achieve faster movement of the CPE devices.

Information About Layer 3 CPE Mobility

The Layer 3 CPE Mobility feature allows CPE devices to move from cable modem to other by trigger of any unicast upstream packets of IPv4 or IPV6.

Each cable modem would be situated at a business hotspot location and the CPE devices move from one business location to another, where the service provider is the same and the head end CMTS is the same. This mobility is allowed for selected IP subnets.

The IPv4 or IPv6 subnets that are configured with mobility must match with the IPv4 or IPv6 subnets already configured on bundle or sub-bundle interface. Otherwise, configuration will not be accepted and the following message will be displayed:

```
Please remove the previous online CPEs or reset CMs,
```

When you remove mobility subnets under bundle or sub-bundle interface. The following warning message will be displayed after mobility subnets is configured or removed.

```
Warning: Please remove the previous online CPEs or reset CMs, to make the mobility scope
change works for every device !!!
```

Note If you have enabled mobility configuration for a subnet, the existing online CPE devices will be updated to aware of the mobility subnets, and the CPU usage will rise up during that time. So it's better to configure the mobility subnets before CM and CPE come online.

Enabling the Layer 3 CPE Mobility feature may, in certain situations, cause excessive punted packets. By default, the Source-Based Rate-Limiting (SBRL) feature rate-limits these punted packets to avoid CPU overload.

Benefits of Layer 3 CPE Mobility

The feature provides the movement of CPE devices from one cable modem to another without change in the IP address and the TCP or UDP sessions established are maintained.

How to Configure Layer 3 Mobility

Configuring CPE Mobility

This section describes how to enable mobility on a particular IP subnet on a interface or subinterface bundle.

Before you begin

Mobility subnets should match the IPv4 or IPv6 address configured on the bundle or sub-bundle interface.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**interface bundle bundle number| bundle-subif-number**
	Example:
	`Router(config)# interface bundle 1`
	or
	`Router(config)# interface Bundle 1.1`
		Enters interface configuration or subinterface mode.

Step 4	**cable l3-mobility** *IP-address mask | IPv6 prefix*
	Example:
	`Router(config-if)# cable l3-mobility 2001:DB:22:1::1/64`
	Example:
	`Router(config-subif)# cable l3-mobility 192.0.3.1 255.255.255.0`
	Example:
	`Router(config-subif)#cable l3-mobility 2001:DB:22:1::1/64`
		Enables mobility for a particular IPv4 or IPv6 subnet.
		Note This command can be configured on a interface or a subinterface bundle.

Step 5	**exit** **Example:**	Exits interface configuration mode.

	Command or Action	Purpose
	`Router(config-if)# exit`	

What to do next

Troubleshooting Tips

If the mobility IP address does not match with the mobility subnet, the following warning message is displayed:

```
Mobility IP should match the IDB subnet!
```

If you remove the IPv4 or IPv6 address from the interface, the mobility scope is removed for the IP address and the following warning message is displayed.

```
IPv6 2001:DBB:3:111::1 removed from Mobility subnets on Bundle1
```

.

Configure Source-Based Rate Limit (SBRL) for L3-mobility

This section describes how to configure Source-Based Rate Limit (SBRL) for the L3-mobility feature. This procedure is optional and if not configured, the default SBRL configuration will apply.

Note SBRL for L3-mobility is enabled by default, so this configuration is optional.

Subscriber-side SBRL has a global and per-punt-cause configuration. L3-mobility punts are only subject to the per-punt-cause configuration. Traffic streams are identified by hashing the punt-cause and the source-MAC-address. This value is used as the index for rate-limiting. There is no special processing for hash-collisions, so hash-colliding streams are treated as if they are the same stream.

The default rate for L3-mobility punts is 4 packets per second.

Before you begin

Note All punted packets are subject to CoPP and the punt-policer.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**platform punt-sbrl subscriber punt-cause** *punt-cause* **rate** *rate* **Example:** `Router(config)# platform punt-sbrl subscriber punt-cause 99 rate 8`	Configures Subscriber-MAC-address SBRL.
Step 4	**exit** **Example:** `Router(config-if)# exit`	Exits global configuration mode.

Disabling CPE Mobility

This section describes how to disable mobility on a particular IP subnet.

Before you begin

The CPE mobility should be enabled on a particular IP subnet before you complete this procedure.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**interface bundle** *bundle number* | *bundle-subif-number*
	Example:
	`Router(config)# interface bundle 1`
	or
	`Router(config)# interface Bundle 1.1`
		Enters interface configuration or subinterface mode.

	Command or Action	Purpose

Step 4	**no cable l3-mobility** *IP-address mask* | *IPv6 prefix*
	Example:
	`Router(config-if)# cable l3-mobility 192.0.3.1 255.255.255.0`
	`Router(config-if)# cable l3-mobility 2001:DB:22:1::1/64`
		Disbles mobility for a particular IPv4 or IPv6 subnet.
		Note This command can be configured on a interface or a subinterface bundle

Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.

Verifying Layer 3 Mobility Configuration

To verify the layer 3 mobility configuration, use the **show cable bundle** command.

Configuration Examples for Layer 3 Mobility

This section provides the following configuration examples:

Example: Configuring CPE Layer 3 Mobility

The following example shows how to configure the layer 3 CPE mobility on a interface bundle:

```
Router#show running interface bundle 10
Building configuration...
Current configuration : 1247 bytes
!
interface Bundle10
ip address 192.0.3.1 255.255.255.0 secondary
ip address 192.2.21.1 255.255.255.0 secondary
ip address 192.3.23.1 255.255.255.0
ip pim sparse-dense-mode
ip igmp static-group 231.1.1.1
no cable arp filter request-send
no cable arp filter reply-accept
cable l3-mobility 192.0.3.1 255.255.255.0
cable l3-mobility 192.2.21.1 255.255.255.0
cable l3-mobility 192.3.23.1 255.255.255.0
cable l3-mobility 2001:DB:26:1::1/64
cable l3-mobility 2001:DB:27:1::1/96
cable dhcp-giaddr primary
cable helper-address 20.1.0.3
ipv6 address 2001:DB:26:1::1/64
ipv6 address 2001:DB:27:1::1/96
ipv6 enable
ipv6 nd reachable-time 3600000
```

```
ipv6 nd managed-config-flag
ipv6 nd other-config-flag
ipv6 dhcp relay destination 2001:DB:1:1:214:4FFF:FEA9:5863
end
```

Example: Configuring SBRL for L3-mobility

The following example shows how SBRL is configured for L3-mobility:

```
Router# show run | i punt-sbrl
platform punt-sbrl subscriber punt-cause 99 rate 8
```

Additional References

The following sections provide references related to Layer 3 CPE Mobility feature for the Cisco CMTS routers.

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

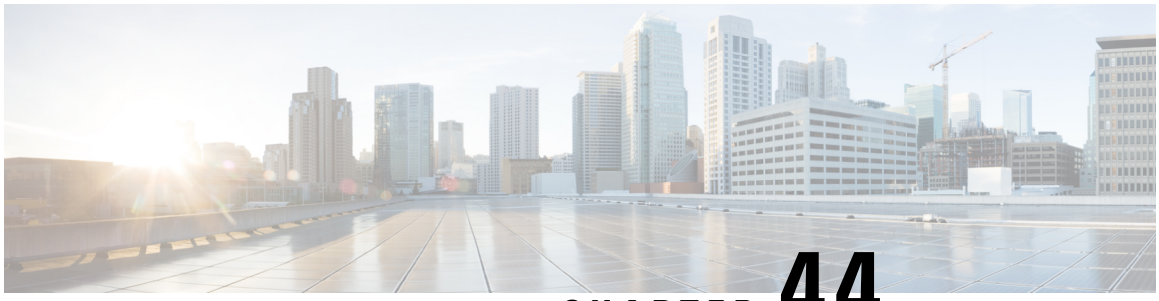
Feature Information for Layer 3 CPE Mobility

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 114: Feature Information for Layer 3 CPE Mobility

Feature Name	Releases	Feature Information
Layer 3 Mobility	Cisco IOS-XE 16.5.1	This feature was integrated into Cisco <polaris release label> on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 44

DOCSIS 3.0 Multicast Support

The Cisco cBR Series Routers support multicast improvements based on Data-over-Cable Service Interface Specifications (DOCSIS) 3.0. DOCSIS 3.0 multicast support improves bandwidth efficiency and allows service providers to offer differentiated quality of service for different types of traffic.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 713
- Prerequisites for the DOCSIS 3.0 Multicast Support, on page 714
- Restrictions for the DOCSIS 3.0 Multicast Support, on page 714
- Information About the DOCSIS 3.0 Multicast Support, on page 715
- How to Configure the DOCSIS 3.0 Multicast Support, on page 719
- How to Monitor the DOCSIS 3.0 Multicast Support, on page 726
- Configuration Examples for DOCSIS 3.0 Multicast Support, on page 729
- Additional References, on page 731
- Feature Information for DOCSIS 3.0 Multicast Support , on page 732

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 115: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for the DOCSIS 3.0 Multicast Support

- DOCSIS 3.0-compliant Cisco CMTS and DOCSIS 3.0-enabled cable modems are required.
- Cisco CMTS must be MDF-enabled by default.
- Quality of service (QoS) parameters must be configured for various multicast sessions.

Restrictions for the DOCSIS 3.0 Multicast Support

- You cannot disable explicit tracking.
- For multicast QoS, you must define three objects and templates, Service-Class, Group-QoS-Config (GQC), and Group-Config, and associate them to a particular bundle or forwarding interface.
- You must define a default service class and GQC before defining objects and templates.
- Static multicast feature is always enabled and you cannot disable it.
- The service flow attribute-based selection will be ignored if the group configuration is configured on the default forwarding interface.
- The multicast DSID feature is supported only on DOCSIS 3.0-compliant cable modems.

- The cable multicast mdf-disable wb-incapable-cm command disables multicast downstream service identifier (DSID) forwarding capability on the cable modem, which impacts the DSID capability between the Cisco CMTS and the cable modem.
- The multicast traffic to CPE increases two-fold after changing the multicast QoS configuration or the service-flow attribute during an active session. The traffic replication will continue till the default session timeout period (180 seconds). After the session timeout, the multicast DSID is removed from both Cisco CMTS and CM, and normal multicast traffic flow is resumed.
- For the DOCSIS 3.0 Multicast support feature to function properly, the CPE and the CM must be in the same virtual routing and forwarding (VRF) interface.

Information About the DOCSIS 3.0 Multicast Support

IP multicast, an integral technology in networked applications, is the transmission of the same information to multiple recipients. Any network application, including cable networks, can benefit from the bandwidth efficiency of multicast technology. Two new technologies—Channel Bonding and Single Source Multicast (SSM)—are expected to dramatically accelerate multicast deployment.

The channel bonding and SSM technologies dramatically increase the operational efficiency of the existing hybrid fiber-coaxial (HFC) network. Using the multicast improvements, the cable operators can seamlessly deliver advanced services like video on demand (VoD), internet protocol television (IPTV), and facilitate interactive video and audio, and data services.

The following sections explain the benefits of DOCSIS 3.0 Multicast Support:

Multicast DSID Forwarding

DOCSIS 3.0 multicast support introduces centralized control at the Cisco CMTS to provide flexibility and scalability to support a large array of multicast protocols. It replaces the Internet Group Management Protocol (IGMP), version 2 snooping infrastructure, which was part of the DOCSIS 1.1 and 2.0 models. Now, the Cisco CMTS allocates an unique Downstream Service Identifier (DSID) to identify every multicast stream. These DSIDs are sent to the CMs that use these DSIDs to filter and forward Multicast traffic to the CPEs.

The multicast DSID forwarding (MDF) provides the following benefits:

- Unique identification of packet stream across bonding group within a MAC domain.
- Designation of packet stream as either Any Source Multicast (ASM) or Source Specific Multicast (SSM) per multicast channel.
- Implementation of multicast DSID management on the Route Processor (RP) makes it operate on a standalone basis.
- Snooping of all upstream signal control packets by the Cisco CMTS to find the customer premises equipment (CPE) on the Multicast DSID-based Forwarding (MDF) enabled CM and allocates DSID from the pool.
- Transmission of allocated DSIDs to the CM through Dynamic Bonding Change (DBC) message.
- Reuse of DSIDs on other MDF-enabled CMs in the same bonding group, joining the multicast session.
- Removal of DSIDs from the CM through a DBC message by the Cisco CMTS after a multicast session leave event.
- Release of DSID to the pool by the Cisco CMTS when the last member leaves the bonding group.
- The following DSIDs are preallocated for each primary downstream (modular and integrated cable interfaces) to forward general query messages. These DSIDs form part of the multicast group signaling protocol. Other multicast groups, do no use these DSIDs.

- IGMPv2 general query (IPv4)
- IGMPv3 general query (IPv4)
- MLDv1 general query (IPv6)
- MLDv2 general query (IPv6)
- Preregistration of DSID (IPv6)

- Allocation of DSID ensures traffic segregation between virtual private networks (VPNs) for DOCSIS 3.0 MDF-enabled CMs. For example, two clients from two VPNs joining the same multicast will get two distinct DSIDs.

Multicast Forwarding on Bonded CM

Multicast packets to the DOCSIS 3.0-enabled CMs are transmitted as bonded packets with DSID extension header on the primary bonding group if the Secondary Multicast Bonding Group is disabled. Multicast packets for MDF-disabled or pre-DOCSIS 3.0 CMs are transmitted as non-bonded without DSID extension header. For more information on this feature, refer to Multicast Secondary Bonding Group, on page 717.

In a network, where only MDF-enabled or MDF-disabled CMs exist, the traffic is segregated using field types. The MDF-enabled CM forwards the frame with the field type and the MDF-disabled CM drops it. The DSID labeling ensures that MDF-enabled CM gets a copy of the multicast session to prevent “cross talk”.

For hybrid CMs (MDF-enabled and MDF-disabled CMs) that do not support field type forwarding, you should configure per session encryption or security association identifier (SAID) isolation to ensure traffic segregation. DOCSIS 3.0 mandates that if the hybrid CM fails to forward field type frames, the Cisco CMTS should employ multicast security association identifier (MSAID) isolation. This isolation is achieved by assigning different MSAID to each replication, one to bonded CM and another to the non-bonded or hybrid CM. This helps to prevent CMs from receiving duplicate traffic.

Static TLV Forwarding

As per DOCSIS 3.0 specifications, the Cisco CMTS must support Static Multicast. When the CM tries to register with the Cisco CMTS, the Cisco CMTS checks whether Static Multicast Encoding is present in the CM configuration file. If the Static Multicast Encoding is present, the Cisco CMTS sends a DSID corresponding to each Static Multicast channel in the Registration-Response (REG-RSP) message.

The Multicast DSID management is located at Supervisor and the interface card has to contact the Supervisor for proper DSID assignment. The interface card also caches the response from Supervisor to eliminate the need to communicate to the Supervisor for subsequent Static Multicast encoding.

Explicit Tracking

The Cisco CMTS can perform explicit tracking with IGMPv3 support. The IGMPv3 removes the report suppression feature associated with the IGMPv2 specification enabling the Cisco CMTS to get the complete information on session and host information. This benefits the IGMP Fast Leave processing and DSID management for each CM.

A host or session database is used to track hosts (IP/MAC) joining a particular multicast session. From the host, you can track the CM based on the SID and cable downstream interface. This database also helps to determine whether the Cisco CMTS should remove the DSID from a particular CM when the multicast session is over.

Multicast Quality of Service Enhancement

DOCSIS 3.0 mandates that the CMTS should not admit any flow exceeding the session limit. Though the current Multicast QoS (MQoS) session limit admits the session, it fails to provide any QoS for sessions exceeding the session limit.

Note Multicast packets are sent using the default Group Service Flows (GSF) when the Multicast QoS feature is disabled.

As part of DOCSIS 3.0 requirements for Multicast QoS, Group Classifier Rules (GCR) is supported. The Cisco CMTS determines the set of Group Configurations (GCs) whose session range matches the multicast group address. For SSM, the source address is also used to identify the matching GCs. A GCR is created for each matching GC and linked to the multicast session. The GCR is assigned also with an unique identifier, SAID, and Group Service Flow (GSF).

The following conditions are used to select the GC entries:

- The GC entry with the highest rule priority is selected, if more than one GC entry matches.
- All matching GC entries are selected, when multiple GCs have the same highest rule priority.

The GCR classification is done based on type of service (TOS) fields. The TOS specifier in the GCR is used to choose the correct GCR when multiple GCRs match a single multicast session.

Note When two multicast group configurations (GCs) have the same session range and configuration (under global or bundle configuration), then the same forwarding interface selection is not guaranteed.

Non-IP multicasts and broadcast packets use GSF. They are similar to individual service flows and are shared by all the CMs on a particular Digital Command Signal (DCS) matching the same GCR. A single GSF is used for multicast sessions matching different GCs using the same aggregate GQC.

Multicast Secondary Bonding Group

The DOCSIS 3.0-compliant CM can receive multicast packets from non-primary (or bonded) channels using the MDF support at the CMTS.

The multicast secondary bonding group is defined as a shared bonding group or RF channel that feeds more than one fiber node through an optical split. This allows CMs from different primary bonding groups and channels to listen to one or more shared sets. The multicast packets are replicated only to the shared downstream channel set, which helps conserve the downstream bandwidth.

DOCSIS 3.0 defines attribute-based service flow creation, which allows the Cisco CMTS to make more “intelligent” decisions on the selection of bonding group or individual channel for unicast and multicast forwarding.

The Multicast Secondary Bonding Group provides the following benefits:

- New MQoS and attribute-based forwarding for Multicast Secondary Bonding Group.
- The primary downstream interface acts as a forwarding interface for narrowband CMs.
- The following algorithm is used to select a forwarding interface for wideband CMs:

- A primary bonding group is selected if a group-config matching the session is present in it. MQoS parameters are taken from the group-config.
- A primary bonding group is selected if a group-config is not present at the bundle level or at the global level.
- A group-config found at the bundle level or global level is used to find the Group-QoS-Config (GQC) and eventually the attribute and forbidden bit-masks, which are then used to find the interface.
- All Wideband Cable Modems (WCMs) in a bundle use the same secondary bonding group if a bundle-level group-config or global-level group-config is configured.

- The IGMP report ignores a source if the given source address fails to find a matching interface.
  - If a matching interface is found, that interface is used for forwarding and the MQoS parameters are taken from the matching group-config from the forwarding interface or bundle interface or global level.
  - If a matching interface is not found, then the IGMP report is ignored.
- For a static join, attribute-based forwarding is not supported, and only the primary downstream is used.

Load Balancing

The Load Balancing feature does not load balance a CM while a multicast stream is going on for that particular CM. It utilizes the Explicit Tracking Database, which holds complete information on the CM subscription to achieve this.

Multicast DSID Forwarding Disabled Mode

For any application that needs the cable modem to perform IGMP snooping, the MDF on the cable modem must be disabled. Cable modems registered in MDF-enabled mode by the Cisco CMTS do not perform IGMP snooping because MDF forwarding is based on DSID filtering. The **cable multicast mdf-disable** command disables the MDF capability on the cable modem.

This command is configured on the route processor and is downloaded to the cable line card via the configuration update. The configuration does not change the Cisco CMTS forwarding mechanism or DSID allocation. The Cisco CMTS allocates the DSID and the multicast packet is encapsulated with the DSID header. This does not affect traffic forwarding on the MDF-disabled cable modem. According to DOCSIS3.0 specification, pre-DOCSIS2.0 or MDF-disabled cable modems ignore the DSID header and continue multicast forwarding based on the Group Media Access Control (GMAC) from IGMP snooping. When the cable modem runs in MDF-disabled mode, only IGMPv2 is supported and the Cisco CMTS drops IGMPv3 and MLD messages.

Multicast encryption based on BPI+ is not supported on non-MDF cable modems, if IGMP SSM mapping is used. A non-MDF cable modem is either a pre-DOCSIS 3.0 cable modem or a DOCSIS 3.0 cable modem running in MDF-disabled mode.

MDF1 Support for DOCSIS 2.0 Hybrid Cable Modems

The Cisco CMTS router enables MDF capability for DOCSIS 2.0 hybrid cable modems, IPv6, and other cable modems that advertise MDF capability to allow IPv6 packet forwarding. The **wb-incapable-cm** keyword in the **cable multicast mdf-disable** command disables MDF on all DOCSIS 2.0 hybrid cable modems including DOCSIS Set-Top Gateway (DSG) hybrid embedded cable modems to support IGMP snooping.

DSG Disablement for Hybrid STBs

The **cable multicast mdf-disable** command with the **wb-incapable-cm** keyword prevents all DOCSIS 2.0 DSG embedded cable modems from receiving DSG multicast traffic besides disabling MDF support.

The **wb-incapable-cm** keyword disables MDF capability only on non-DSG DOCSIS 2.0 hybrid cable modems. To disable MDF capability on all DSG embedded cable modems (DOCSIS 3.0 DSG and DOCSIS 2.0 DSG hybrid), a new keyword, DSG, is introduced.

Note After disabling MDF capability, you must run **clear cable modem reset** command to bring all DSG embedded cable modems online.

Benefits of MDF1 Support

- Supports IPv6 on different known cable modem firmware types.
- Disables the MDF capability on the Cisco CMTS.
- Supports In-Service Software Upgrade (ISSU) and line card high availability.

How to Configure the DOCSIS 3.0 Multicast Support

This section describes the following tasks that are required to implement DOCSIS 3.0 Multicast Support on Cisco CMTS Routers:

Configuring Basic Multicast Forwarding

To configure a basic multicast forwarding profile that can be applied to a DOCSIS 3.0 multicast configuration, use the **ip multicast-routing** command. You must configure a multicast routing profile before you can proceed with a multicast group.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**IP multicast-routing** [**vrf**] **Example:** `Router(config)# IP multicast-routing vrf`	Enables multicast routing globally or on a particular virtual routing and forwarding (VRF) interface.

	Command or Action	Purpose
Step 4	**interface bundle** *number* **Example:** `Router(config)# interface bundle 1`	Configures the interface bundle and enters interface configuration mode.
Step 5	**IP pim sparse-mode** **Example:** `Router(config-if)# IP pim sparse-mode`	Configures sparse mode of operation. **Note** The Cisco CMTS router must have a Protocol Independent Multicast (PIM) rendezvous point (RP) configured for the PIM sparse mode. The Supervisor is configured using the **ip pim rp-address** command or Auto- Supervisor configuration protocol.
Step 6	**IP pim sparse-dense-mode** **Example:** `Router(config-if)# IP pim sparse-dense-mode`	Configures the interface for either sparse mode or dense mode of operation, depending on the mode in which the multicast group is operating.
Step 7	**IP igmp version version-number** **Example:** `Router(config-if)# IP igmp version 3`	Configures the interface to use IGMP version 3.

Configuring Multicast DSID Forwarding

The multicast DSID forwarding is enabled by default. You cannot configure this feature.

Configuring Explicit Tracking

The Explicit Tracking feature is enabled by default. You cannot configure it.

Configuring Multicast QoS

To configure a Multicast QoS profile that can be applied to a DOCSIS 3.0 configuration, use the **cable multicast group-qos** command. You must configure a Multicast QoS profile before you can add a Multicast QoS profile to a QoS multicast group.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configureterminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable service class** *class-index* **name** *service-class-name* **Example:** `Router(config)# cable service class 1 name MQOS_DEFAULT`	Configures the name of the cable service class.
Step 4	**cable service class** *class-index* **downstream** **Example:** `Router(config)# cable service class 1 downstream`	Configures the downstream for the cable service class.
Step 5	**cable service class** *class-index* **max-rate** *maximum-bandwidth-allowed* **Example:** `Router(config)# cable service class 1 max-rate 10000000`	Configures the maximum allowed bandwidth for the cable service class.
Step 6	**cable service class** *class-index* **min-rate** *cir* **Example:** `Router(config)# cable service class 1 min-rate 1000000`	Configures the minimum committed information rate for the cable service class.
Step 7	**cable multicast group-qos default scn** *service-class-name* **aggregate** **Example:** `Router(config)# cable multicast group-qos default scn MQOS_DEFAULT aggregate`	Specifies the default service class name for the QoS profile.
Step 8	**cable multicast qos group** *number* **priority** *value* **Example:** `Router(config)# cable multicast qos group 20 priority 1`	Configures a multicast QoS group and enters multicast QoS configuration mode, and specifies the priority of the cable multicast QoS group.

	Command or Action	Purpose
Step 9	**application-id** *app-id* **Example:** `Router(config-mqos)# application-id 10`	Specifies the application identification number of the multicast QoS group. This value is configured to enable admission control to the multicast QoS group.
Step 10	**session-range ip-address ip-mask** **Example:** `Router(config-mqos)# session-range 230.0.0.0 255.0.0.0`	Specifies the session range IP address and IP mask of the multicast QoS group. You can configure multiple session ranges.
Step 11	**cable multicast qos group** *number priority value* [**global**] **Example:** `Router(config)#cable multicast qos group 20 priority 63 global`	Specifies the multicast QoS group identifier.

Selecting a Forwarding Interface Based on Service Flow Attribute

The Service Flow Attribute feature allows a bonded CM to listen to multiple bonding groups, and using the interface-specific bit-masks, the CM can select the best route to receive multicast traffic.

The Service Flow Attribute feature allows selection of a forwarding interface based on the DOCSIS 3.0 construct named “service flow attribute mask.” Every interface has an attribute bit-mask depicting attributes of that interface. The multicast service class specified in the group QoS configuration contains required and forbidden attribute bit-masks. If a bonded CM can listen to multiple bonding groups (wideband interfaces), using specific bit-masks in the service class as well as on the bonding group, then one of these bonding groups can be selected for forwarding of multicast traffic.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable service class** *class-index* **name** *name* **Example:**	Configures the service class name.

	Command or Action	Purpose
	`Router(config)# cable service class 10 name mcast10`	
Step 4	**cable service class** *class-index* **downstream** **Example:** `Router(config)# cable service class 10 downstream`	Configures the downstream for the selected service class.
Step 5	**cable service class** *class-index* **max-rate** *maximum-rate* **Example:** `Router(config)# cable service class 10 max-rate 1000000`	Configures the maximum rate for the selected service class.
Step 6	**cable service class** *class-index* **min-rate** *minimum-rate* **Example:** `Router(config)#  cable service class 10 min-rate 100000`	Configures the minimum rate for the selected service class.
Step 7	**cable service class** *class-index* **req-attr-mask** *required-attribute-mask* **Example:** `Router(config)# cable service class 10 req-attr-mask 8000000F`	Configures the required attribute mask for the selected service class.
Step 8	**cable service class** *class-index* **forb-attr-mask** *forbidden-attribute-mask* **Example:** `Router(config)# cable service class 10 forb-attr-mask 7FFFFFF0`	Configures the forbidden attribute mask for the selected service class name.
Step 9	**cable multicast group-qos** *number* **scn** *service-class-name* **aggregate** **Example:** `Router(config)# cable multicast group-qos 1 scn 10 mcast10 aggregate`	Configures the cable multicast group QoS identifier, service class name, and multicast value.
Step 10	**cable multicast qos group** *group* **priority** *priority* **Example:**	Configures the cable MQoS group and enters MQoS configuration mode.

	Command or Action	Purpose
	`Router(config)# cable multicast qos group 1 priority 1`	
Step 11	**session-range** *session-range mask* **Example:** `Router(config-mqos)# session-range 230.1.1.1 255.255.255.255`	Specifies session range.
Step 12	**group-qos** *qos* **Example:** `Router(config-mqos)# group-qos 1`	Specifies the group QoS.
Step 13	**exit** **Example:** `Router(config-mqos)# exit`	Returns to global configuration mode.

Step 14	**interface bundle** *number*
	• **ip address** *ip mask*
	• **ip pim sparse-mode**
	• **ip helper-address** *helper-address*
	• **cable multicast-qos group** *group*
	Example:
	`Router(config)# interface Bundle1`
	`Router(config-if)#ip address 40.1.1.1 255.255.255.0`
	`Router(config-if)#ip pim sparse-mode`
	`Router(config-if)#ip helper-address 2.39.16.1`
	`Router(config-if)#cable multicast-qos group 1`
		Configures the interface bundle with the IP address, helper address, and MQoS group.

Step 15	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 16	**interface wideband-cable** *slot*/*subslot*/*port*:*wideband-channel* • **description** *description* • **cable bundle** *number* • **cable rf-channel channel-list** *grouplist* **bandwidth-percent** *bw-percent* • **cable downstream attribute-mask attribute-***mask*	Selects the interface for forwarding based on the bit-masks specified in the service class and on the wideband interface.

	Command or Action	Purpose

	Example:
	
	`Router(config)# interface Wideband-Cable1/0/0:0`
	`Router(config-if)# description cable rf-channels channel-list 0-7 bandwidth-percent 20`
	`Router(config-if)# cable bundle 1`
	`Router(config-if)# cable rf-channels channel-list 0-7 bandwidth-percent 20`
	`Router(config-if)# cable downstream attribute-mask 8000000F`	

Step 17	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring Multicast DSID Forwarding Disabled Mode

To disable MDF on the cable modem, use the **cable multicast mdf-disable** command in global configuration mode.

Note Multicast encryption based on BPI+ is not supported on non-MDF cable modems, if IGMP SSM mapping is used.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable multicast mdf-disable [wb-incapable-cm]** **Example:** `Router(config)# cable multicast mdf-disable`	Disables MDF capability on the cable modem.

	Command or Action	Purpose

Step 4	exit
	Example:
	`Router(config)# exit`
	`Router#`
		Exits the global configuration mode.

How to Monitor the DOCSIS 3.0 Multicast Support

To monitor the DOCSIS 3.0 Multicast Support feature, use the following procedures:

Verifying the Basic Multicast Forwarding

To verify the configuration parameters for basic multicast forwarding, use the **show ip mroute** command as shown in the following example:

```
Router# show ip mroute

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 230.1.1.1), 00:00:03/00:02:55, RP 30.1.1.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Bundle1, Forward/Sparse, 00:00:03/00:02:55, H
(*, 224.0.1.40), 00:12:02/00:02:19, RP 30.1.1.1, flags: SJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Bundle1, Forward/Sparse, 00:12:02/00:02:19
```

To verify the multicast information for the specified virtual interface bundle, based on IGMPv3, use the **show cable bundle multicast** command as shown in the following example:

```
Router# show cable bundle 1 multicast

CableBundle Interface Source IP  Multicast IP    MAC Address
1    Bundle1.1 *       230.1.1.1    0100.5e00.0001
```

To verify the MAC forwarding table for the specified virtual interface bundle, based on IGMPv3, use the **show cable bundle forwarding** command as shown in the following example:

```
Router# show cable bundle 1 forwarding

MAC address Interface Flags Location link sublink
00c0.5e01.0203 Cable8/0/0 3 64E5BF60 0 64E5BE00
```

```
00c0.5e01.0203 Cable7/0/0 3 64E5BE00 0 0
00c0.5e01.0101 Cable8/0/0 3 64E5BEE0 0 64E5BE40
```

Verifying the Multicast DSID Forwarding

To verify the entire DSID database content, use the **show cable multicast dsid** command as shown in the following example:

```
Router# show cable multicast dsid
Multicast Group    : 230.1.2.3
         Source    : *
         IDB       : Bu2         Interface: Mo1/1/0:0   Dsid: 0x1F078
         StatIndex : 2           SAID: DEFAULT
Multicast Group    : 230.1.2.3
         Source    : *
         IDB       : Bu2         Interface: Mo1/1/0:0   Dsid: 0x1F078
         StatIndex : 3           SAID: 8196
Multicast Group    : 230.1.2.3
         Source    : *
         IDB       : Bu2         Interface: Mo1/1/0:0   Dsid: 0x1F078
```

StatIndex : 4 SAID: 8197

To verify the entire database content, use the **show cable multicast db** command as shown in the following example:

Router# **show cable multicast db**

```
interface : Bundle1
Session (S,G) : (*,230.1.1.1)
Fwd Intfc Sub Intfc Host Intfc CM Mac Hosts
Wi1/1/0:0 Bundle1 Ca5/0/0 0018.6852.8056 1
```

To verify the information for the registered and unregistered CMs, use the **show cable modem verbose** command as shown in the following example:

Router# **show cable modem 0010.7bb3.fcd1 verbose**

```
MAC Address : 00C0.7bb3.fcd1
IP Address : 10.20.113.2
Prim Sid : 1
QoS Profile Index : 6
Interface : C5/0/U5
sysDescr : Vendor ABC DOCSIS 2.0 Cable Modem
Upstream Power : 0 dBmV (SNR = 33.25 dBmV)
Downstream Power : 0 dBmV (SNR = ----- dBmV)
Timing Offset : 1624
Initial Timing Offset : 2812
Received Power : 0.25
MAC Version : DOC1.0
Qos Provisioned Mode : DOC1.0
Enable DOCSIS2.0 Mode : Y
Phy Operating Mode : atdma
Capabilities : {Frag=N, Concat=N, PHS=N, Priv=BPI}
Sid/Said Limit : {Max Us Sids=0, Max Ds Saids=0}
Optional Filtering Support : {802.1P=N, 802.1Q=N}
Transmit Equalizer Support : {Taps/Symbol= 0, Num of Taps= 0}
Number of CPE IPs : 0(Max CPEs = 1)
CFG Max-CPE : 1
```

```
Flaps : 373(Jun 1 13:11:01)
Errors : 0 CRCs, 0 HCSes
Stn Mtn Failures : 0 aborts, 3 exhausted
Total US Flows : 1(1 active)
Total DS Flows : 1(1 active)
Total US Data : 1452082 packets, 171344434 bytes
Total US Throughput : 0 bits/sec, 0 packets/sec
Total DS Data : 1452073 packets, 171343858 bytes
Total DS Throughput : 0 bits/sec, 0 packets/sec
Active Classifiers : 0 (Max = NO LIMIT)
DSA/DSX messages : reject all
Dynamic Secret : A3D1028F36EBD54FDCC2F74719664D3F
Spoof attempt : Dynamic secret check failed
Total Time Online : 16:16
```

Verifying the Explicit Tracking Feature

To verify explicit tracking information, use the **show cable multicast db** command as shown in the following example:

Router# **show cable multicast db**

```
Interface : Bundle1
Session (S,G) : (*,230.1.1.1)
Fwd Intfc  Sub Intfc  Host Intfc   CM Mac           Hosts
Mo1/1/0:0  Bundle1    Ca5/0/0      0018.6852.8056       1
```

Verifying the Multicast QoS Feature

To verify the cable MQoS details, use the **show cable multicast qos** commands as shown in the following example:

```
Router# show cable multicast qos ?
group-config Display Multicast Group Config information
group-encryption Display Multicast Group Encryption information
group-qos Display Multicast Group QOS information
Router# show cable multicast qos group-config
Multicast Group Config 1 : Priority 1
Group QOS - 1
Group Encryption - 1
Session Range - Group Prefix 230.0.0.0 Mask 255.0.0.0 Source Prefix 0.0.0.0 Mask 0.0.0.0
Router# show cable multicast qos group-encryption
Multicast Group Encryption 1 : Algorithm 56bit-des
Router# show cable multicast qos group-qos
Group QOS Index Service Class Control Igmp Limit Override
DEFAULT MQOS_DEFAULT Aggregate NO-LIMIT 1 MQOS Aggregate NO-LIMIT
```

To verify the DOCSIS service flows on a given cable interface, use the **show interface service-flow** command as shown in the following example:

```
Router# show interface cable 6/0 service-flow

Sfid  Sid   Mac Address     QoS Param Index Type    Dir    Curr   Active
BG/CH
                            Prov  Adm  Act                 State  Time
4     8193  ffff.ffff.ffff   3     3    3   sec(S)  DS     act    21h57m
5     8196  ffff.ffff.ffff   4     4    4   sec(S)  DS     act    00:17
```

Verifying the Service Flow Attributes

To verify the configuration of service flow attributes on the service class configuration, use the show cable service-class verbose command as shown in the following example:

```
Router# show cable service-class 10 verbose
Index:                                  10
Name:                                   mcast10
Direction:                              Downstream
Traffic Priority:                       0
Maximum Sustained Rate:                 1000000 bits/sec
Max Burst:                              3044 bytes
Minimum Reserved Rate:                  1000000 bits/sec
Minimum Packet Size                     0 bytes
Admitted QoS Timeout                    200 seconds
Active QoS Timeout                      0 seconds
Required Attribute Mask                 8000000F
Forbidden Attribute Mask                7FFFFFF0
Scheduling Type:                        Undefined
Max Latency:                            0 usecs
Parameter Presence Bitfield:            {0x3148, 0x0}
```

To verify the configuration of SF attributes on the Wideband interface configuration, use the **show running-config interface** command as shown in the following example:

```
Router# show running-config interface Wideband-Cable 1/0/0:2
interface Wideband-Cable1/0/0:2
 cable bundle 1
 cable bonding-group-id 3
 cable rf-channel 3
 cable downstream attribute-mask 8000000F
end
```

Verifying the Multicast Group Classifiers

To verify the details of the Group Classifier Rule, use the **show interface wideband-cable multicast-gcr** command as shown in the following example:

```
Router# show interface wideband-cable 1/1/0:0 multicast-gcr
Group Classifier Rules on Wideband-Cable1/1/0:0:
Classifier_id  Group_id  Group_Qos_id  Sid   SFID   ref_count
7              1         1             8196  10     1
8              2         1             8197  11     1
```

Troubleshooting Tips

Make sure that CM can listen to the RF-frequencies specified for the Wideband interfaced chosen for forwarding multicast traffic.

Configuration Examples for DOCSIS 3.0 Multicast Support

This section provides the following configuration examples:

Example: Configuring Basic Multicast Forwarding

Note The commands given below are required to enable the Cisco CMTS to forward multicast packets. However, Multicast QoS, and Authorization features are all optional for multicast packets to be forwarded correctly.

In the following example, a basic multicast forwarding profile is configured.

```
ip multicast-routing
interface TenGigabitEthernet4/1/0
  ip pim sparse-dense-mode
interface Bundle 1
  ip pim sparse-mode
  ip igmp version 3
```

Example: Configuring Multicast QoS

Note A default service class and GQC must be defined before proceeding with configuring Multicast QoS.

In the following example, Multicast QoS is configured. You should define three objects and templates and then associate these to a particular bundle or forwarding interface. The objects are Service-Class, Group-QoS-Config (GQC), and Group-Config.

```
cable service class 1 name MQOS_DEFAULT
cable service class 1 downstream
cable service class 1 max-rate 10000000
cable service class 1 min-rate 1000000
cable multicast group-qos default scn MQOS_DEFAULT aggregate
cable multicast group-qos 10 scn MQOS single
cable multicast qos group 20 priority 1
application-id 10
session-range 230.0.0.0 255.0.0.0
tos 1 6 15
vrf name1
cable multicast qos group 20 priority 63 global
```

Example: Configuring Forwarding Interface Selection Based on Service Flow Attribute

In the following example, the service flow attribute-based Forwarding Interface Selection is configured. To send multicast traffic for group 230.1.1.1, interface W6/0/0:0 is selected. The multicast QoS parameters are taken from group qos 1 (effectively from service class “mcast10”).

```
cable service class 10 name mcast10
cable service class 10 downstream
cable service class 10 max-rate 1000000
cable service class 10 min-rate 1000000
cable service class 10 req-attr-mask 8000000F
cable service class 10 forb-attr-mask 7FFFFFF0
cable multicast group-qos 1 scn mcast10 aggregate
```

```
cable multicast qos group 1 priority 1
session-range 230.1.1.1 255.255.255.255
 group-qos 1
interface Bundle1
 ip address 40.1.1.1 255.255.255.0
 ip pim sparse-mode
 ip helper-address 2.39.16.1
 cable multicast-qos group 1
end
interface Wideband-Cable6/0/0:0
cable bundle 10
cable rf-channels channel-list 0-7 bandwidth-percent 20
cable downstream attribute-mask 8000000F
end
```

Additional References

The following sections provide references related to the DOCSIS 3.0 Multicast Support on the CMTS Routers.

Related Documents

Related Topic	Document Title
CMTS cable commands	http://www.cisco.com/en/US/docs/ios/cable/command/reference/cbl_book.htmlCisco IOS CMTS Cable Command Reference
Multicast VPN and DOCSIS 3.0 Multicast QoS	Multicast VPN and DOCSIS 3.0 Multicast QoS Support
DOCSIS 3.0 QoS Support	DOCSIS WFQ Scheduler on the Cisco CMTS Routers

Standards

Standard	Title
CM-SP-CMCIv3-I01-080320	Cable Modem to Customer Premise Equipment Interface Specification
CM-SP-MULPIv3.0-I08-080522	MAC and Upper Layer Protocols Interface Specification
CM-SP-OSSIv3.0-I07-080522	Operations Support System Interface Specification
CM-SP-PHYv3.0-I07-080522	Physical Layer Specification
CM-SP-SECv3.0-I08-080522	Security Specification

MIBs

MIB[3]	MIBs Link
• DOCS-MCAST-AUTH-MIB • DOCS-MCAST-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

[3] Not all supported MIBs are listed.

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

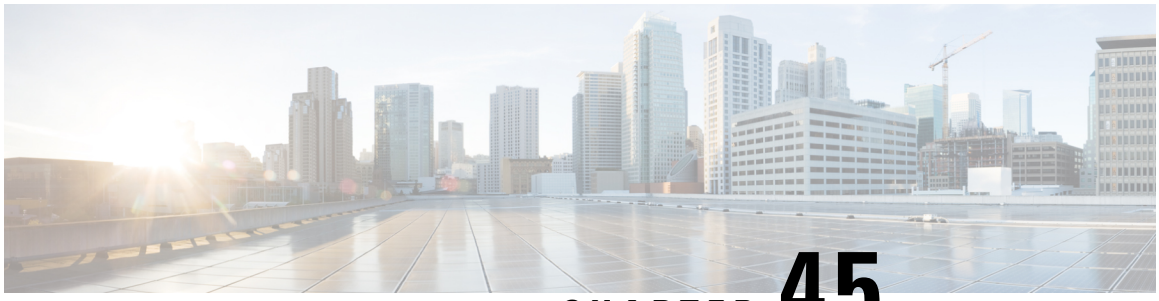
Feature Information for DOCSIS 3.0 Multicast Support

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 116: Feature Information for DOCSIS 3.0 Multicast Support

Feature Name	Releases	Feature Information
DOCSIS 3.0 Multicast Support	Cisco IOS-XE Release 3.15.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 45

IPv6 Segment Routing on Cisco cBR

In Cisco Converged Broadband Router, IPv6 Segment Routing is available as a sub mode of IPv6 address configuration.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 733
- Information about IPv6 Segment Routing, on page 734
- How to Configure IPv6 Segment Routing, on page 735
- Configuration Examples, on page 737
- Feature Information for IPv6 Segment Routing, on page 738

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 117: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information about IPv6 Segment Routing

IPv6 Segment Routing (SR) is an SDN technology supporting IPv6 forwarding. In SR, a source or edge router performs source routing of traffic and encodes it as a segment list in an IPv6 routing extension header. The network is not required to maintain a per-application or per-flow state.

Any IPv6 capable node in a network may forward IPv6 traffic with an SR extension header to the first segment in the segment list without supporting IPv6 Segment Routing (SRv6).

At the node that hosts the current segment in the segment list, SRv6 is configured to modify the destination address of the traffic containing the SR extension header and destined to that segment ID. As part of SRv6 final processing, the next segment ID in the SR extension header is written to the destination address of the packet and a lookup is performed to forward the traffic to the new destination address.

The forwarding and SRv6 end processing continues at nodes hosting the segment IDs in the SR extension header until the last segment in the list is removed and the traffic is delivered to its ultimate destination.

Restriction for Configuring IPv6 Segment Routing

Configuring duplicate IPv6 addresses on the same interface is not allowed.

How to Configure IPv6 Segment Routing

Configuring IPv6 Segment Routing on cBR

To configure IPv6 segment routing, use the following procedure.

1. Enter segment-routing sub mode when configuring an IPv6 address on an interface.

```
enable
configure terminal
interface type [slot_#/]port_#
ipv6 address ipv6_address_prefix/prefix_length
ipv6 address ipv6_address_prefix/prefix_length segment-routing
```

2. Define a local prefix as an SID

```
ipv6-sr prefix-sid
exit
```

Verifying IPv6 Segment Routing Configuration

The following example shows how to verify SRv6 configuration:c

```
Router#sh run
*Oct 17 13:13:23.975: %SYS-5-CONFIG_I: Configured from console by console
Router#sh run | sec Ether
interface Ethernet0/0
 no ip address
 shutdown
 ipv6 address 2001::2001/64 segment-routing >>>>>>>
  ipv6-sr prefix-sid >>>>>>>
```

Configure Multiple IPv6 Addresses for Segment Routing

To configure multiple IPv6 addresses for SRv6 under the same interface, use the following commands.

1. Enter segment-routing sub mode when configuring an IPv6 address on an interface.

```
enable
configure terminal
interface type [slot_#/]port_#
ipv6 address ipv6_address_prefix/prefix_length segment-routing
```

2. Define a local prefix as an SID.

```
ipv6-sr prefix-sid
exit
```

Verifying IPv6 Segment Routing Configuration on Multiple IPv6 Addresses

The following example shows how to verify SRv6 configuration for multiple IPv6 addresses:

```
Router#sh run | sec Ether
interface Ethernet0/0
```

```
 no ip address
 shutdown
 ipv6 address 2001:db8:110::/64 segment-routing  >>> submode 1
  ipv6-sr prefix-sid
 ipv6 address 2001:db9:111::/64 segment-routing  >>> submode 2
  ipv6-sr prefix-sid
interface Ethernet0/1
 no ip address
 shutdown
interface Ethernet0/2
 no ip address
 shutdown
interface Ethernet0/3
 no ip address
 shutdown
interface Ethernet1/0
 no ip address
 shutdown
interface Ethernet1/1
 no ip address
 shutdown
interface Ethernet1/2
```

Disabling Prefix SID

To disable the local prefix SID associated to the segment ID, use the following commands.

```
enable
configure terminal
interface type [slot_#/]port_#
ipv6 address ipv6_address_prefix/prefix_length segment-routing
no ipv6-sr prefix-sid
end
```

Verifying whether Prefix SID is Disabled

The following example shows how to verify whether the prefix SID is disabled:

```
Router#sh run | sec Ether
interface Ethernet0/0
 no ip address
 shutdown
 ipv6 address 110::110/64 segment-routing >>> "ipv6-sr prefix sid" is no longer present
ipv6 address 111::111/64 segment-routing
ipv6-sr prefix-sid
```

Disabling SRv6 for a Prefix-SID

To disable SRv6 configuration for an IPv6 address and remove the IPv6 address, use the following command:

```
enable
configure terminal
interface type [slot_#/]port_#
no ipv6 address ipv6_address_prefix/prefix_length segment-routing
end
```

Verifying whether SRv6 is Disabled and Prefix SID Removed

The following example shows how to verify whether SRv6 is disabled and the prefix SID is removed for the prefix SID.

```
Router#sh run |
*Oct 17 13:17:51.523: %SYS-5-CONFIG_I: Configured from console by console
Router#sh run | sec Ether
interface Ethernet0/0
 no ip address
 shutdown
 ipv6 address 110::110/64 segment-routing
ipv6 address 111::111/64 segment-routing is entirely removed from ethernet0/0
```

Configuration Examples

This section provides examples for IPv6 Segment Routing.

Example: Configuring IPv6 Segment Routing on Cisco cBR

```
Router#conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#inter Ether0/0
Router(config-if)#ipv6 address 110::110/64 ?
  anycast
  eui-64
  segment-routing
  <cr>
Router(config-if)#ipv6 address 110::110/64 segment-routing
Router(config-if-sr-ipv6)#?
ipv6 address segment-routing mode configuration commands:
  default  Set a command to its defaults
  exit     Exit from SR submode
  ipv6-sr  Request options specific to IPV6 segment-routing
  no       Negate a command or set its defaults
Router(config-if-sr-ipv6)#ipv6-sr ?
  prefix-sid  Set host prefix as IPv6 SR identifier prefix-sid
Router(config-if-sr-ipv6)#ipv6-sr prefix-sid
Router(config-if-sr-ipv6)#exit
Router(config-if)#exit
Router(config)#exit
Router#
```

Example: Configure Multiple IPv6 Addresses for SRv6

```
Router#conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#inter Ether 0/0
Router(config-if)# ipv6 address 110::110/64 segment-routing
Router(config-if)# ipv6 address 111::111/64 segment-routing
Router(config-if-sr-ipv6)#ipv6-sr prefix-sid
Router(config-if-sr-ipv6)#end
```

Example: Disabling Prefix SID

```
Router#conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#inter Ether0/0
```

```
Router(config-if)#inter Ether0/0
Router(config-if)#ipv6 address 110::110/64 segment-routing
Router(config-if-sr-ipv6)#no ipv6-sr prefix-sid
Router(config-if-sr-ipv6)#end
```

Example: Disabling SR with an Active Prefix SID

```
Router#conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#inter Ether0/0
Router(config-if)#no ipv6 address 111::111/64 segment-routing
Router(config-if)#end
```

Feature Information for IPv6 Segment Routing

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

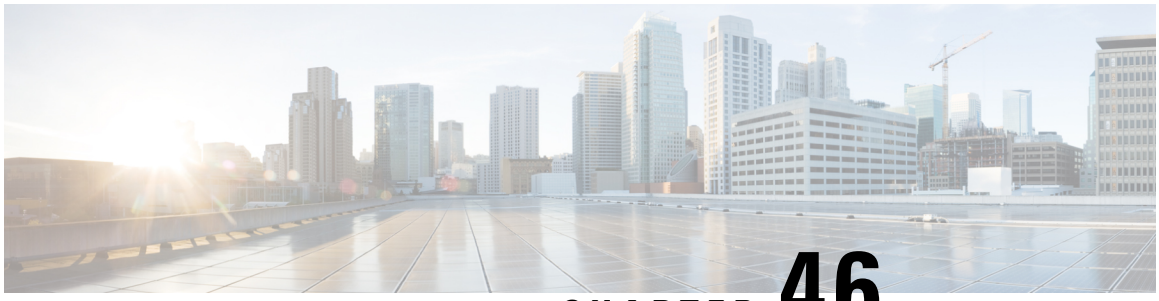
Table 118: Feature Information for IPv6 Segment Routing

Feature Name	Releases	Feature Information
IPv6 Segment Routing	Cisco IOS XE Everest 16.5.1	This feature was introduced in Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.

PART VII

Layer 3 Configuration—IP Access Control Lists

- IP Access Control Lists, on page 741
- Creating an IP Access List and Applying It to an Interface, on page 753
- Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, on page 771
- Refining an IP Access List , on page 791
- IP Named Access Control Lists, on page 805
- IPv4 ACL Chaining Support , on page 815
- IPv6 ACL Chaining with a Common ACL , on page 821
- Commented IP Access List Entries, on page 827
- Standard IP Access List Logging , on page 833
- IP Access List Entry Sequence Numbering, on page 839
- ACL IP Options Selective Drop , on page 851
- ACL Syslog Correlation , on page 857
- IPv6 Access Control Lists, on page 869
- IPv6 Template ACL , on page 879
- IPv6 ACL Extensions for Hop by Hop Filtering, on page 885



CHAPTER 46

IP Access Control Lists

Access control lists (ACLs) perform packet filtering to control which packets move through a network and to where. The packet filtering provides security by helping to limit the network traffic, restrict the access of users and devices to a network, and prevent the traffic from leaving a network. IP access lists reduce the chance of spoofing and denial-of-service attacks and allow dynamic, temporary user-access through a firewall.

IP access lists can also be used for purposes other than security, such as bandwidth control, restrict the content of routing updates, redistribute routes, trigger dial-on-demand (DDR) calls, limit debug output, and identify or classify traffic for quality of service (QoS) features. This module provides an overview of IP access lists.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 741
- Information About IP Access Lists, on page 742
- Additional References, on page 750
- Feature Information for IP Access Lists, on page 751

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 119: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About IP Access Lists

Benefits of IP Access Lists

Access control lists (ACLs) perform packet filtering to control the flow of packets through a network. Packet filtering can restrict the access of users and devices to a network, providing a measure of security. Access lists can save network resources by reducing traffic. The benefits of using access lists are as follows:

- Authenticate incoming rsh and rcp requests—Access lists can simplify the identification of local users, remote hosts, and remote users in an authentication database that is configured to control access to a device. The authentication database enables Cisco software to receive incoming remote shell (rsh) and remote copy (rcp) protocol requests.
- Block unwanted traffic or users—Access lists can filter incoming or outgoing packets on an interface, thereby controlling access to a network based on source addresses, destination addresses, or user authentication. You can also use access lists to determine the types of traffic that are forwarded or blocked at device interfaces. For example, you can use access lists to permit e-mail traffic to be routed through a network and to block all Telnet traffic from entering the network.
- Control access to vty—Access lists on an inbound vty (Telnet) can control who can access the lines to a device. Access lists on an outbound vty can control the destinations that the lines from a device can reach.

- Identify or classify traffic for QoS features—Access lists provide congestion avoidance by setting the IP precedence for Weighted Random Early Detection (WRED) and committed access rate (CAR). Access lists also provide congestion management for class-based weighted fair queueing (CBWFQ), priority queueing, and custom queueing.
- Limit debug command output—Access lists can limit debug output based on an IP address or a protocol.
- Provide bandwidth control—Access lists on a slow link can prevent excess traffic on a network.
- Provide NAT control—Access lists can control which addresses are translated by Network Address Translation (NAT).
- Reduce the chance of DoS attacks—Access lists reduce the chance of denial-of-service (DoS) attacks. Specify IP source addresses to control traffic from hosts, networks, or users from accessing your network. Configure the TCP Intercept feature to can prevent servers from being flooded with requests for connection.
- Restrict the content of routing updates—Access lists can control routing updates that are sent, received, or redistributed in networks.
- Trigger dial-on-demand calls—Access lists can enforce dial and disconnect criteria.

Border Routers and Firewall Routers Should Use Access Lists

There are many reasons to configure access lists; for example, you can use access lists to restrict contents of routing updates or to provide traffic flow control. One of the most important reasons to configure access lists is to provide a basic level of security for your network by controlling access to it. If you do not configure access lists on your router, all packets passing through the router could be allowed onto all parts of your network.

An access list can allow one host to access a part of your network and prevent another host from accessing the same area. In the figure below, by applying an appropriate access list to the interfaces of the router, Host A is allowed to access the Human Resources network and Host B is prevented from accessing the Human Resources network.

Access lists should be used in firewall routers, which are often positioned between your internal network and an external network such as the Internet. You can also use access lists on a router positioned between two parts of your network, to control traffic entering or exiting a specific part of your internal network.

To provide some security benefits of access lists, you should at least configure access lists on border routers--routers located at the edges of your networks. Such an access list provides a basic buffer from the outside network or from a less controlled area of your own network into a more sensitive area of your network. On these border routers, you should configure access lists for each network protocol configured on the router interfaces. You can configure access lists so that inbound traffic or outbound traffic or both are filtered on an interface.

Access lists are defined on a per-protocol basis. In other words, you should define access lists for every protocol enabled on an interface if you want to control traffic flow for that protocol.

Definition of an Access List

Access control lists (ACLs) perform packet filtering to control the movement of packets through a network. Packet filtering provides security by limiting the access of traffic into a network, restricting user and device access to a network, and preventing traffic from leaving a network. IP access lists reduce the chance of spoofing and denial-of-service attacks, and allow dynamic, temporary user-access through a firewall.

IP access lists can also be used for purposes other than security, such as to control bandwidth, restrict the content of routing updates, redistribute routes, trigger dial-on-demand (DDR) calls, limit debug output, and identify or classify traffic for quality of service (QoS) features.

An access list is a sequential list that consists of at least one **permit** statement and possibly one or more **deny** statements. In the case of IP access lists, these statements can apply to IP addresses, upper-layer IP protocols, or other fields in IP packets.

Access lists are identified and referenced by a name or a number. Access lists act as packet filters, filtering packets based on the criteria defined in each access list.

After you configure an access list, for the access list to take effect, you must either apply the access list to an interface (by using the **ip access-group** command), a vty (by using the **access-class** command), or reference the access list by any command that accepts an access list. Multiple commands can reference the same access list.

In the following configuration, an IP access list named branchoffices is configured on Ten Gigabit Ethernet interface 4/1/0 and applied to incoming packets. Networks other than the ones specified by the source address and mask pair cannot access Ten Gigabit Ethernet interface 4/1/0. The destinations for packets coming from sources on network 172.16.7.0 are unrestricted. The destination for packets coming from sources on network 172.16.2.0 must be 172.31.5.4.

```
ip access-list extended branchoffices
 10 permit 172.16.7.0 0.0.0.3 any
 20 permit 172.16.2.0 0.0.0.255 host 172.31.5.4
!
interface tengigabitethernet 4/1/0
 ip access-group branchoffices in
```

Access List Rules

The following rules apply to access lists:

- Only one access list per interface, per protocol, and per direction is allowed.
- An access list must contain at least one **permit** statement or all packets are denied entry into the network.
- The order in which access list conditions or match criteria are configured is important. While deciding whether to forward or block a packet, Cisco software tests the packet against each criteria statement in the order in which these statements are created. After a match is found, no more criteria statements are checked. The same **permit** or **deny** statements specified in a different order can result in a packet being passed under one circumstance and denied in another circumstance.
- If an access list is referenced by a name, but the access list does not exist, all packets pass. An interface or command with an empty access list applied to it permits all traffic into the network.
- Standard access lists and extended access lists cannot have the same name.
- Inbound access lists process packets before the packets are routed to an outbound interface. Inbound access lists that have filtering criteria that deny packet access to a network saves the overhead of routing lookup. Packets that are permitted access to a network based on the configured filtering criteria are processed for routing. For inbound access lists, when you configure a **permit** statement, packets are processed after they are received, and when you configure a **deny** statement, packets are discarded.
- Outbound access lists process packets before they leave the device. Incoming packets are routed to the outbound interface and then processed by the outbound access list. For outbound access lists, when you

configure a **permit** statement, packets are sent to the output buffer, and when you configure a **deny** statement, packets are discarded.

- An access list can control traffic arriving at a device or leaving a device, but not traffic originating at a device.

Helpful Hints for Creating IP Access Lists

The following tips will help you avoid unintended consequences and help you create more efficient, useful access lists.

- Create the access list before applying it to an interface (or elsewhere), because if you apply a nonexistent access list to an interface and then proceed to configure the access list, the first statement is put into effect, and the implicit **deny** statement that follows could cause you immediate access problems.
- Another reason to configure an access list before applying it is because an interface with an empty access list applied to it permits all traffic.
- All access lists need at least one **permit** statement; otherwise, all packets are denied and no traffic passes.
- Use the statement **permit any any** if you want to allow all other packets not already denied. Using the statement **permit any any** in effect avoids denying all other packets with the implicit deny statement at the end of an access list. Do not make your first access list entry **permit any any** because all traffic will get through; no packets will reach the subsequent testing. In fact, once you specify **permit any any** , all traffic not already denied will get through.
- Although all access lists end with an implicit **deny** statement, we recommend use of an explicit **deny** statement (for example, **deny ip any any**). On most platforms, you can display the count of packets denied by issuing the **show access-list** command, thus finding out more information about who your access list is disallowing. Only packets denied by explicit **deny** statements are counted, which is why the explicit **deny** statement will yield more complete data for you.
- While you are creating an access list or after it is created, you might want to delete an entry.
  - You cannot delete an entry from a numbered access list; trying to do so will delete the entire access list. If you need to delete an entry, you need to delete the entire access list and start over.
  - You can delete an entry from a named access list. Use the **no permit** or **no deny** command to delete the appropriate entry.
- In order to make the purpose of individual statements more scannable and easily understood at a glance, you can write a helpful remark before or after any statement by using the **remark** command.
- If you want to deny access to a particular host or network and find out if someone from that network or host is attempting to gain access, include the **log** keyword with the corresponding **deny** statement so that the packets denied from that source are logged for you.
- This hint applies to the placement of your access list. When trying to save resources, remember that an inbound access list applies the filter conditions before the routing table lookup. An outbound access list applies the filter conditions after the routing table lookup.

Named or Numbered Access Lists

All access lists must be identified by a name or a number. Named access lists are more convenient than numbered access lists because you can specify a meaningful name that is easier to remember and associate with a task. You can reorder statements in or add statements to a named access list.

Named access lists support the following features that are not supported by numbered access lists:

- IP options filtering
- Noncontiguous ports
- TCP flag filtering
- Deleting of entries with the **no permit** or **no deny** command

Note Not all commands that accept a numbered access list will accept a named access list. For example, vty uses only numbered access lists.

Standard or Extended Access Lists

All access lists are either standard or extended access lists. If you only intend to filter on a source address, the simpler standard access list is sufficient. For filtering on anything other than a source address, an extended access list is necessary.

- Named access lists are specified as standard or extended based on the keyword **standard** or **extended** in the **ip access-list** command syntax.
- Numbered access lists are specified as standard or extended based on their number in the **access-list** command syntax. Standard IP access lists are numbered 1 to 99 or 1300 to 1999; extended IP access lists are numbered 100 to 199 or 2000 to 2699. The range of standard IP access lists was initially only 1 to 99, and was subsequently expanded with the range 1300 to 1999 (the intervening numbers were assigned to other protocols). The extended access list range was similarly expanded.

Standard Access Lists

Standard IP access lists test only source addresses of packets (except for two exceptions). Because standard access lists test source addresses, they are very efficient at blocking traffic close to a destination. There are two exceptions when the address in a standard access list is not a source address:

- On outbound VTY access lists, when someone is trying to telnet, the address in the access list entry is used as a destination address rather than a source address.
- When filtering routes, you are filtering the network being advertised to you rather than a source address.

Extended Access Lists

Extended access lists are good for blocking traffic anywhere. Extended access lists test source and destination addresses and other IP packet data, such as protocols, TCP or UDP port numbers, type of service (ToS), precedence, TCP flags, and IP options. Extended access lists can also provide capabilities that standard access lists cannot, such as the following:

- Filtering IP Options
- Filtering TCP flags
- Filtering noninitial fragments of packets (see the module "Refining an IP Access List")
- Time-based entries (see the module "Refining an IP Access List")

Note Packets that are subject to an extended access list will not be autonomous switched.

IP Packet Fields You Can Filter to Control Access

You can use an extended access list to filter on any of the following fields in an IP packet. Source address and destination address are the two most frequently specified fields on which to base an access list:

- Source address--Specifies a source address to control packets coming from certain networking devices or hosts.
- Destination address--Specifies a destination address to control packets being sent to certain networking devices or hosts.
- Protocol--Specifies an IP protocol indicated by the keyword **eigrp** , **gre** , **icmp** , **igmp** , **ip** , **ipinip** , **nos** , **ospf** , **tcp** , or **udp** , or indicated by an integer in the range from 0 to 255 (representing an Internet protocol). If you specify a transport layer protocol (**icmp** , **igmp** , **tcp** , or **udp**), the command has a specific syntax.
  - Ports and non-contiguous ports--Specifies TCP or UDP ports by a port name or port number. The port numbers can be noncontiguous port numbers. Port numbers can be useful to filter Telnet traffic or HTTP traffic, for example.
  - TCP flags--Specifies that packets match any flag or all flags set in TCP packets. Filtering on specific TCP flags can help prevent false synchronization packets.
- IP options--Specifies IP options; one reason to filter on IP options is to prevent routers from being saturated with spurious packets containing them.

Wildcard Mask for Addresses in an Access List

Address filtering uses wildcard masking to indicate to the software whether to check or ignore corresponding IP address bits when comparing the address bits in an access list entry to a packet being submitted to the access list. By carefully setting wildcard masks, you can specify one or more IP addresses for permit or deny tests.

Wildcard masking for IP address bits uses the number 1 and the number 0 to specify how the software treats the corresponding IP address bits. A wildcard mask is sometimes referred to as an inverted mask because a 1 and 0 mean the opposite of what they mean in a subnet (network) mask.

- A wildcard mask bit 0 means check the corresponding bit value; they must match.
- A wildcard mask bit 1 means ignore that corresponding bit value; they need not match.

If you do not supply a wildcard mask with a source or destination address in an access list statement, the software assumes an implicit wildcard mask of 0.0.0.0, meaning all values must match.

Unlike subnet masks, which require contiguous bits indicating network and subnet to be ones, wildcard masks allow noncontiguous bits in the mask.

The table below shows examples of IP addresses and masks from an access list, along with the corresponding addresses that are considered a match.

Table 120: Sample IP Addresses, Wildcard Masks, and Match Results

Address	Wildcard Mask	Match Results
0.0.0.0	255.255.255.255	All addresses will match the access list conditions.
172.18.0.0/16	0.0.255.255	Network 172.18.0.0
172.18.5.2/16	0.0.0.0	Only host 172.18.5.2 matches
172.18.8.0	0.0.0.7	Only subnet 172.18.8.0/29 matches
172.18.8.8	0.0.0.7	Only subnet 172.18.8.8/29 matches
172.18.8.15	0.0.0.3	Only subnet 172.18.8.15/30 matches
10.1.2.0	0.0.252.255 (noncontiguous bits in mask)	Matches any even-numbered network in the range of 10.1.2.0 to 10.1.254.0

Access List Sequence Numbers

The ability to apply sequence numbers to IP access list entries simplifies access list changes. Prior to the IP Access List Entry Sequence Numbering feature, there was no way to specify the position of an entry within an access list. If you wanted to insert an entry in the middle of an existing list, all of the entries after the desired position had to be removed, then the new entry was added, and then all the removed entries had to be reentered. This method was cumbersome and error prone.

This feature allows users to add sequence numbers to access list entries and resequence them. When you add a new entry, you specify the sequence number so that it is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced to create room to insert the new entry.

Access List Logging

The Cisco IOS software can provide logging messages about packets permitted or denied by a single standard or extended IP access list entry. That is, any packet that matches the entry will cause an informational logging message about the packet to be sent to the console. The level of messages logged to the console is controlled by the **logging console** global configuration command.

The first packet that triggers the access list entry causes an immediate logging message, and subsequent packets are collected over 5-minute intervals before they are displayed or logged. The logging message includes the access list number, whether the packet was permitted or denied, the source IP address of the packet, and the number of packets from that source permitted or denied in the prior 5-minute interval.

However, you can use the **ip access-list log-update** command to set the number of packets that, when match an access list (and are permitted or denied), cause the system to generate a log message. You might want to do this to receive log messages more frequently than at 5-minute intervals.

Caution If you set the *number-of-matches* argument to 1, a log message is sent right away, rather than caching it; every packet that matches an access list causes a log message. A setting of 1 is not recommended because the volume of log messages could overwhelm the system.

Even if you use the **ip access-list log-update** command, the 5-minute timer remains in effect, so each cache is emptied at the end of 5 minutes, regardless of the count of messages in each cache. Regardless of when the log message is sent, the cache is flushed and the count reset to 0 for that message the same way it is when a threshold is not specified.

Note The logging facility might drop some logging message packets if there are too many to be handled or if there is more than one logging message to be handled in 1 second. This behavior prevents the router from crashing due to too many logging packets. Therefore, the logging facility should not be used as a billing tool or an accurate source of the number of matches to an access list.

Alternative to Access List Logging

Packets matching an entry in an ACL with a log option are process switched. It is not recommended to use the log option on ACLs, but rather use NetFlow export and match on a destination interface of Null0. This is done in the CEF path. The destination interface of Null0 is set for any packet that is dropped by the ACL.

Additional IP Access List Features

Beyond the basic steps to create a standard or extended access list, you can enhance your access lists as mentioned below. Each of these methods is described completely in the module entitled “Refining an Access List.”

- You can impose dates and times when **permit** or **deny** statements in an extended access list are in effect, making your access list more granular and specific to an absolute or periodic time period.
- After you create a named access list, you might want to add entries or change the order of the entries, known as resequencing an access list.
- You can achieve finer granularity when filtering packets by filtering on noninitial fragments of packets.

Where to Apply an Access List

You can apply access lists to the inbound or outbound interfaces of a device. Applying an access list to an inbound interface controls the traffic that enters the interface and applying an access list to an outbound interface controls the traffic that exits the interface.

When software receives a packet at the inbound interface, the software checks the packet against the statements that are configured for the access list. If the access list permits packets, the software processes the packet. Applying access lists to filter incoming packets can save device resources because filtered packets are discarded before entering the device.

Access lists on outbound interfaces filter packets that are transmitted (sent) out of the interface. You can use the TCP Access Control List (ACL) Splitting feature of the Rate-Based Satellite Control Protocol (RBSCP) on the outbound interface to control the type of packets that are subject to TCP acknowledgment (ACK) splitting on an outbound interface.

You can reference an access list by using a **debug** command to limit the amount of debug logs. For example, based on the filtering or matching criteria of the access list, debug logs can be limited to source or destination addresses or protocols.

You can use access lists to control routing updates, dial-on-demand (DDR), and quality of service (QoS) features.

Additional References

Related Documents

Related Topic	Document Title
IP access list commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Addressing Services Command Reference
Filtering on source address, destination address, or protocol	Creating an IP Access List and Applying It to an Interface" module
Filtering on IP Options, TCP flags, noncontiguous ports, or TTL	Creating an IP Access List to Filter IP Options, TCP Flags, or Noncontiguous Ports module

Standards

Standards & RFCs	Title
None	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

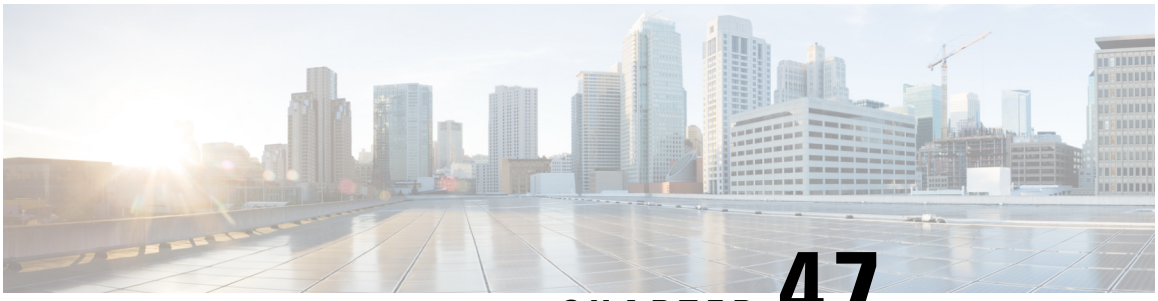
Feature Information for IP Access Lists

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 121: Feature Information for IP Access Lists

Feature Name	Releases	Feature Information
IP Access Lists	Cisco IOS-XE Release 3.15.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 47

Creating an IP Access List and Applying It to an Interface

IP access lists provide many benefits for securing a network and achieving nonsecurity goals, such as determining quality of service (QoS) factors or limiting **debug** command output. This module describes how to create standard, extended, named, and numbered IP access lists. An access list can be referenced by a name or a number. Standard access lists filter on only the source address in IP packets. Extended access lists can filter on source address, destination address, and other fields in an IP packet.

After you create an access list, you must apply it to something in order for it to have any effect. This module describes how to apply an access list to an interface. However, there are many other uses for access lists, which are mentioned in this module and described in other modules and in other configuration guides for various technologies.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 754
- Information About Creating an IP Access List and Applying It to an Interface, on page 754
- How to Create an IP Access List and Apply It to an Interface, on page 756
- Configuration Examples for Creating an IP Access List and Applying It to an Interface, on page 764
- Additional References Creating an IP Access List and Applying It to an Interface, on page 768
- Feature Information Creating an IP Access List and Applying It to an Interface , on page 769

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 122: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About Creating an IP Access List and Applying It to an Interface

Helpful Hints for Creating IP Access Lists

- Create the access list before applying it to an interface (or elsewhere), because if you apply a nonexistent access list to an interface and then proceed to configure the access list, the first statement is put into effect, and the implicit **deny** statement that follows could cause immediate access problems.
- Another reason to configure an access list before applying it is because an interface with an empty access list applied to it permits all traffic.

- All access lists need at least one **permit** statement; otherwise, all packets are denied and no traffic passes.
- Organize your access list so that more specific references in a network or subnet appear before more general ones.
- A packet will match the first ACE in the ACL. Thus, a **permit ip any any** will match all packets, ignoring all subsequent ACES.
- Although all access lists end with an implicit **deny** statement, we recommend use of an explicit **deny** statement (for example, **deny ip any any**). On most platforms, you can display the count of packets denied by issuing the **show access-list** command, thus finding out more information about who your access list is disallowing. Only packets denied by explicit **deny** statements are counted, which is why the explicit **deny** statement will yield more complete data for you.
- While you are creating an access list or after it is created, you might want to delete an entry. You can delete an entry from a named access list. Use the **no permit** or **no deny** command to delete the appropriate entry.
- In order to make the purpose of individual statements more scannable and easily understood at a glance, you can write a helpful remark before or after any statement by using the **remark** command.
- If you want to deny access to a particular host or network and find out if someone from that network or host is attempting to gain access, include the **log** keyword with the corresponding **deny** statement so that the packets denied from that source are logged for you.
- This hint applies to the placement of your access list. When trying to save resources, remember that an inbound access list applies the filter conditions before the routing table lookup. An outbound access list applies the filter conditions after the routing table lookup.

Access List Remarks

You can include comments or remarks about entries in any IP access list. An access list remark is an optional remark before or after an access list entry that describes the entry so that you do not have to interpret the purpose of the entry. Each remark is limited to 100 characters in length.

The remark can go before or after a **permit** or **deny** statement. Be consistent about where you add remarks. Users may be confused if some remarks precede the associated **permit** or **deny** statements and some remarks follow the associated statements.

The following is an example of a remark that describes function of the subsequent **deny** statement:

```
ip access-list extended telnetting
 remark Do not allow host1 subnet to telnet out
 deny tcp host 172.16.2.88 any eq telnet
```

Additional IP Access List Features

Beyond the basic steps to create a standard or extended access list, you can enhance your access lists as mentioned below. Each of these methods is described completely in the *Refining an IP Access List module*.

- You can impose dates and times when **permit** or **deny** statements in an extended access list are in effect, making your access list more granular and specific to an absolute or periodic time period.

- After you create a named or numbered access list, you might want to add entries or change the order of the entries, known as resequencing an access list.
- You can achieve finer granularity when filtering packets by filtering on noninitial fragments of packets.

How to Create an IP Access List and Apply It to an Interface

This section describes the general ways to create a standard or extended access list using either a name or a number. Access lists are very flexible; the tasks simply illustrate one **permit** command and one **deny** command to provide you the command syntax of each. Only you can determine how many **permit** and **deny** commands you need and their order.

Note The first two tasks in this module create an access list; you must apply the access list in order for it to function. If you want to apply the access list to an interface, perform the task "Applying the Access List to an Interface".

Creating a Standard Access List to Filter on Source Address

If you want to filter on source address only, a standard access list is simple and sufficient. There are two alternative types of standard access list: named and numbered. Named access lists allow you to identify your access lists with a more intuitive name rather than a number, and they also support more features than numbered access lists.

Creating a Named Access List to Filter on Source Address

Use a standard, named access list if you need to filter on source address only. This task illustrates one **permit** statement and one **deny** statement, but the actual statements you use and their order depend on what you want to filter or allow. Define your **permit** and **deny** statements in the order that achieves your filtering goals.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip access-list standard** *name* **Example:** `Device(config)# ip access-list standard R&D`	Defines a standard IP access list using a name and enters standard named access list configuration mode.
Step 4	**remark** *remark* **Example:** `Device(config-std-nacl)# remark deny Sales network`	(Optional) Adds a user-friendly comment about an access list entry. • A remark can precede or follow an access list entry. • In this example, the remark reminds the network administrator that the subsequent entry denies the Sales network access to the interface (assuming this access list is later applied to an interface).
Step 5	**deny** { *source* [*source-wildcard*] | **any** } [**log**] **Example:** `Device(config-std-nacl)# deny 172.16.0.0 0.0.255.255 log`	(Optional) Denies the specified source based on a source address and wildcard mask. • If the *source-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* to specify the source and source wildcard of 0.0.0.0 255.255.255.255. • In this example, all hosts on network 172.16.0.0 are denied passing the access list. • Because this example explicitly denies a source address and the **log** keyword is specified, any packets from that source are logged when they are denied. This is a way to be notified that someone on a network or host is trying to gain access.
Step 6	**remark** *remark* **Example:** `Device(config-std-nacl)# remark Give access to Tester's host`	(Optional) Adds a user-friendly comment about an access list entry. • A remark can precede or follow an access list entry. • This remark reminds the network administrator that the subsequent entry allows the Tester's host access to the interface.

	Command or Action	Purpose
Step 7	**permit** { *source* [*source-wildcard*] | **any** } [**log**] **Example:** `Device(config-std-nacl)# permit 172.18.5.22 0.0.0.0`	Permits the specified source based on a source address and wildcard mask. • Every access list needs at least one **permit** statement; it need not be the first entry. • If the *source-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* to specify the source and source wildcard of 0.0.0.0 255.255.255.255. • In this example, host 172.18.5.22 is allowed to pass the access list.
Step 8	Repeat some combination of Steps 4 through 7 until you have specified the sources on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.
Step 9	**end** **Example:** `Device(config-std-nacl)# end`	Exits standard named access list configuration mode and enters privileged EXEC mode.
Step 10	**show ip access-list** **Example:** `Device# show ip access-list`	(Optional) Displays the contents of all current IP access lists.

Creating a Numbered Access List to Filter on Source Address

Configure a standard, numbered access list if you need to filter on source address only and you prefer not to use a named access list.

IP standard access lists are numbered 1 to 99 or 1300 to 1999. This task illustrates one **permit** statement and one **deny** statement, but the actual statements you use and their order depend on what you want to filter or allow. Define your **permit** and **deny** statements in the order that achieves your filtering goals.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number permit* { *source* [*source-wildcard*] | **any** } [**log**] **Example:** `Device(config)# access-list 1 permit 172.16.5.22 0.0.0.0`	Permits the specified source based on a source address and wildcard mask. • Every access list needs at least one permit statement; it need not be the first entry. • Standard IP access lists are numbered 1 to 99 or 1300 to 1999. • If the source-wildcard is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address. • Optionally use the keyword any as a substitute for the source source-wildcard to specify the source and source wildcard of 0.0.0.0 255.255.255.255. • In this example, host 172.16.5.22 is allowed to pass the access list.
Step 4	**access-list** *access-list-number* **deny** { *source* [*source-wildcard*] | **any** } [**log**] **Example:** `Device(config)# access-list 1 deny 172.16.7.34 0.0.0.0`	Denies the specified source based on a source address and wildcard mask. • If the *source-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address. • Optionally use the abbreviation **any** as a substitute for the *source source-wildcard* to specify the source and source wildcard of 0.0.0.0 255.255.255.255. • In this example, host 172.16.7.34 is denied passing the access list.
Step 5	Repeat some combination of Steps 3 through 6 until you have specified the sources on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.
Step 6	**end** **Example:**	Exits global configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	
Step 7	**show ip access-list** **Example:** `Device# show ip access-list`	(Optional) Displays the contents of all current IP access lists.

Creating an Extended Access List

If you want to filter on anything other than source address, you need to create an extended access list. There are two alternative types of extended access list: named and numbered. Named access lists allow you to identify your access lists with a more intuitive name rather than a number, and they also support more features.

For details on how to filter something other than source or destination address, see the syntax descriptions in the command reference documentation.

Creating a Named Extended Access List

Create a named extended access list if you want to filter the source and destination address or filter a combination of addresses and other IP fields.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *name* **Example:** `Device(config)# ip access-list extended acl1`	Defines an extended IP access list using a name and enters extended named access list configuration mode.
Step 4	**deny** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:**	(Optional) Denies any packet that matches all of the conditions specified in the statement. • If the *source-wildcard* or *destination-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively.

	Command or Action	Purpose
	`Device(config-ext-nacl)# deny ip 172.18.0.0 0.0.255.255 host 172.16.40.10 log`	• Optionally use the keyword **any** as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255. • Optionally use the keyword **host** *source* to indicate a source and source wildcard of *source* 0.0.0.0 or the abbreviation **host** *destination* to indicate a destination and destination wildcard of *destination* 0.0.0.0. • In this example, packets from all sources are denied access to the destination network 172.18.0.0. Logging messages about packets permitted or denied by the access list are sent to the facility configured by the **logging facility** command (for example, console, terminal, or syslog). That is, any packet that matches the access list will cause an informational logging message about the packet to be sent to the configured facility. The level of messages logged to the console is controlled by the **logging console** command.
Step 5	**permit** *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# permit tcp any any`	Permits any packet that matches all of the conditions specified in the statement. • Every access list needs at least one permit statement. • If the *source-wildcard* or *destination-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255. • In this example, TCP packets are allowed from any source to any destination. • Use the **log-input** keyword to include input interface, source MAC address, or virtual circuit in the logging output.

	Command or Action	Purpose
Step 6	Repeat some combination of Steps 4 through 7 until you have specified the fields and values on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.
Step 7	**end** **Example:** `Device(config-ext-nacl)# end`	Exits standard named access list configuration mode and enters privileged EXEC mode.
Step 8	**show ip access-list** **Example:** `Device# show ip access-list`	(Optional) Displays the contents of all current IP access lists.

Creating a Numbered Extended Access List

Create a numbered extended access list if you want to filter on source and destination address, or a combination of addresses and other IP fields, and you prefer not to use a name. Extended IP access lists are numbered 100 to 199 or 2000 to 2699.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* **remark** *remark* **Example:** `Device(config)# access-list 107 remark allow Telnet packets from any source to network 172.69.0.0 (headquarters)`	(Optional) Adds a user-friendly comment about an access list entry. • A remark of up to 100 characters can precede or follow an access list entry.
Step 4	**access-list** *access-list-number* **permit** *protocol* { *source* [*source-wildcard*] | **any** } { *destination* [*destination-wildcard*] | **any** } [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]	Permits any packet that matches all of the conditions specified in the statement. • Every access list needs at least one **permit** statement; it need not be the first entry. • Extended IP access lists are numbered 100 to 199 or 2000 to 2699.

	Command or Action	Purpose
	Example: `Device(config)# access-list 107 permit tcp any 172.69.0.0 0.0.255.255 eq telnet`	• If the *source-wildcard* or *destination-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255. • TCP and other protocols have additional syntax available. See the **access-list** command in the command reference for complete syntax.
Step 5	**access-list** *access-list-number* **remark** *remark* **Example:** `Device(config)# access-list 107 remark deny all other TCP packets`	(Optional) Adds a user-friendly comment about an access list entry. • A remark of up to 100 characters can precede or follow an access list entry.
Step 6	**access-list** *access-list-number* **deny** *protocol* { *source* [*source-wildcard*] | **any** } { *destination* [*destination-wildcard*] | **any** } [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config)# access-list 107 deny tcp any any`	Denies any packet that matches all of the conditions specified in the statement. • If the *source-wildcard* or *destination-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255.
Step 7	Repeat some combination of Steps 3 through 6 until you have specified the fields and values on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.
Step 8	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.
Step 9	**show ip access-list** **Example:**	(Optional) Displays the contents of all current IP access lists.

	Command or Action	Purpose
	`Device# show ip access-list`	

Applying an Access List to an Interface

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface TenGigabitEthernet4/1/0`	Specifies an interface and enters interface configuration mode.
Step 4	**ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**} **Example:** `Device(config-if)# ip access-group acl1 in`	Applies the specified access list to the inbound interface. • To filter source addresses, apply the access list to the inbound interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Creating an IP Access List and Applying It to an Interface

Example: Filtering on Host Source Address

In the following example, the workstation belonging to user1 is allowed access to Ten Gigabit Ethernet interface 4/1/0, and the workstation belonging to user2 is not allowed access:

```
interface TenGigabitEthernet4/1/0
 ip access-group workstations in
!
```

```
ip access-list standard workstations
 remark Permit only user1 workstation through
 permit 172.16.2.88
 remark Do not allow user2 workstation through
 deny 172.16.3.13
```

Example: Filtering on Subnet Source Address

In the following example, the user1 subnet is not allowed access to Ten Gigabit Ethernet interface 4/1/0, but the Main subnet is allowed access:

```
interface TenGigabitEthernet4/1/0
 ip access-group prevention in
!
ip access-list standard prevention
 remark Do not allow user1 subnet through
 deny 172.22.0.0 0.0.255.255
 remark Allow Main subnet
 permit 172.25.0.0 0.0.255.255
```

Example: Filtering on Source and Destination Addresses and IP Protocols

The following configuration example shows an interface with two access lists, one applied to outgoing packets and one applied to incoming packets. The standard access list named Internet-filter filters outgoing packets on source address. The only packets allowed out the interface must be from source 172.16.3.4.

The extended access list named marketing-group filters incoming packets. The access list permits Telnet packets from any source to network 172.26.0.0 and denies all other TCP packets. It permits any ICMP packets. It denies UDP packets from any source to network 172.26.0 0 on port numbers less than 1024. Finally, the access list denies all other IP packets and performs logging of packets passed or denied by that entry.

```
interface TenGigabitEthernet4/1/0
 ip address 172.20.5.1 255.255.255.0
 ip access-group Internet-filter out
 ip access-group marketing-group in
!
ip access-list standard Internet-filter
 permit 172.16.3.4
ip access-list extended marketing-group
 permit tcp any 172.26.0.0 0.0.255.255 eq telnet
 deny tcp any any
 permit icmp any any
 deny udp any 172.26.0.0 0.0.255.255 lt 1024
 deny ip any any
```

Example: Filtering on Source Addresses Using a Numbered Access List

In the following example, network 10.0.0.0 is a Class A network whose second octet specifies a subnet; that is, its subnet mask is 255.255.0.0. The third and fourth octets of a network 10.0.0.0 address specify a particular host. Using access list 2, the Cisco IOS-XE software would accept one address on subnet 48 and reject all others on that subnet. The last line of the list shows that the software would accept addresses on all other network 10.0.0.0 subnets.

```
interface TenGigabitEthernet4/1/0
 ip access-group 2 in
```

```
!
access-list 2 permit 10.48.0.3
access-list 2 deny 10.48.0.0  0.0.255.255
access-list 2 permit 10.0.0.0  0.255.255.255
```

Example: Preventing Telnet Access to a Subnet

In the following example, the user1 subnet is not allowed to telnet out of Ten Gigabit Ethernet interface 4/1/0:

```
interface TenGigabitEthernet4/1/0
 ip access-group telnetting out
!
ip access-list extended telnetting
 remark Do not allow user1 subnet to telnet out
 deny tcp 172.20.0.0 0.0.255.255 any eq telnet
 remark Allow Top subnet to telnet out
 permit tcp 172.33.0.0 0.0.255.255 any eq telnet
```

Example: Filtering on TCP and ICMP Using Port Numbers

In the following example, the first line of the extended access list named acl1 permits any incoming TCP connections with destination ports greater than 1023. The second line permits incoming TCP connections to the Simple Mail Transfer Protocol (SMTP) port of host 172.28.1.2. The last line permits incoming ICMP messages for error feedback.

```
interface TenGigabitEthernet4/1/0
 ip access-group acl1 in
!
ip access-list extended acl1
 permit tcp any 172.28.0.0 0.0.255.255 gt 1023
 permit tcp any host 172.28.1.2 eq 25
 permit icmp any 172.28.0.0 255.255.255.255
```

Example: Allowing SMTP E-mail and Established TCP Connections

Suppose you have a network connected to the Internet, and you want any host on an Ethernet to be able to form TCP connections to any host on the Internet. However, you do not want IP hosts to be able to form TCP connections to hosts on the Ten Gigabit Ethernet except to the mail (SMTP) port of a dedicated mail host.

SMTP uses TCP port 25 on one end of the connection and a random port number on the other end. The same two port numbers are used throughout the life of the connection. Mail packets coming in from the Internet will have a destination port of 25. Outbound packets will have the port numbers reversed. The fact that the secure system behind the router always will accept mail connections on port 25 is what makes possible separate control of incoming and outgoing services. The access list can be configured on either the outbound or inbound interface.

In the following example, the Ten Gigabit Ethernet network is a Class B network with the address 172.18.0.0, and the address of the mail host is 172.18.1.2. The **established** keyword is used only for the TCP protocol to indicate an established connection. A match occurs if the TCP datagram has the ACK or RST bits set, which indicate that the packet belongs to an existing connection.

```
interface TenGigabitEthernet4/1/0
 ip access-group 102 in
!
```

```
access-list 102 permit tcp any 172.18.0.0 0.0.255.255 established
access-list 102 permit tcp any host 172.18.1.2 eq 25
```

Example: Preventing Access to the Web by Filtering on Port Name

In the following example, the w1and w2 workstations are not allowed web access; other hosts on network 172.20.0.0 are allowed web access:

```
interface TenGigabitEthernet4/1/0
 ip access-group no-web out
!
ip access-list extended no-web
 remark Do not allow w1 to browse the web
 deny host 172.20.3.85 any eq http
 remark Do not allow w2 to browse the web
 deny host 172.20.3.13 any eq http
 remark Allow others on our network to browse the web
 permit 172.20.0.0 0.0.255.255 any eq http
```

Example: Filtering on Source Address and Logging the Packets

The following example defines access lists 1 and 2, both of which have logging enabled:

```
interface TenGigabitEthernet4/1/0
 ip address 172.16.1.1 255.0.0.0
 ip access-group 1 in
 ip access-group 2 out
!
access-list 1 permit 172.25.0.0 0.0.255.255 log
access-list 1 deny 172.30.0.0 0.0.255.255 log
!
access-list 2 permit 172.27.3.4 log
access-list 2 deny 172.17.0.0 0.0.255.255 log
```

If the interface receives 10 packets from 172.25.7.7 and 14 packets from 172.17.23.21, the first log will look like the following:

```
list 1 permit 172.25.7.7 1 packet
list 2 deny 172.17.23.21 1 packet
```

Five minutes later, the console will receive the following log:

```
list 1 permit 172.25.7.7 9 packets
list 2 deny 172.17.23.21 13 packets
```

Example: Limiting Debug Output

The following sample configuration uses an access list to limit the **debug** command output. Limiting the **debug** output restricts the volume of data to what you are interested in, saving you time and resources.

```
Device(config)# ip access-list acl1
Device(config-std-nacl)# remark Displays only advertisements for LDP peer in acl1
Device(config-std-nacl)# permit host 10.0.0.44

Device# debug mpls ldp advertisements peer-acl acl1
```

```
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.17.0.33
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.16.0.31
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.22.0.33
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.0.1
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.0.3
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.1.33
```

Additional References Creating an IP Access List and Applying It to an Interface

Related Documents

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
• Order of access list entries • Access list entries based on time of day or week • Packets with noninitial fragments	Refining an IP Access List
Filtering on IP options, TCP flags, or noncontiguous ports	Creating an IP Access List for Filtering
Controlling logging-related parameters	Understanding Access Control List Loggingl

Standards and RFCs

Standard/RFC	Title
No new or modified standards or RFCs are supported by this feature, and support for existing standards or RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

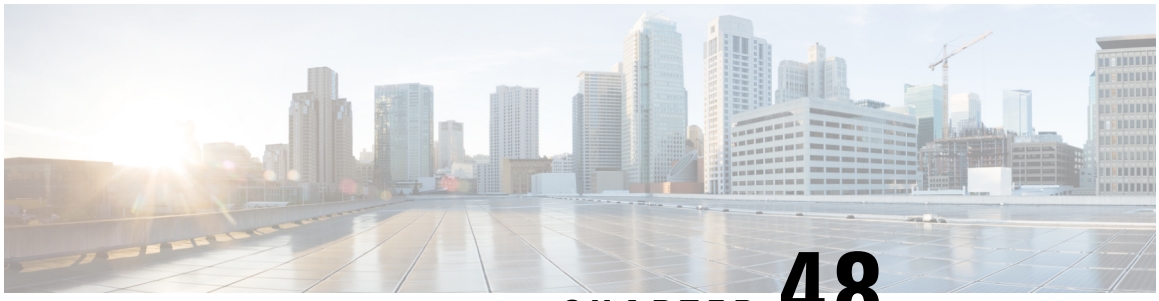
Feature Information Creating an IP Access List and Applying It to an Interface

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 123: Feature Information for Creating an IP Access List and Applying It to an Interface

Feature Name	Releases	Feature Information
IP Access Lists	Cisco IOS-XE Release 3.15.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 48

Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports

This module describes how to use an IP access list to filter IP packets that contain certain IP Options, TCP flags, noncontiguous ports.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 771
- Prerequisites for Creating an IP Access List to Filter IP Options TCP Flags Noncontiguous Ports , on page 772
- Information About Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports , on page 772
- How to Create an IP Access List to Filter IP Options TCP Flags Noncontiguous Ports , on page 776
- Configuration Examples for Filtering IP Options, TCP Flags, Noncontiguous Ports , on page 786
- Additional References, on page 789
- Feature Information for Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values , on page 790

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 124: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Creating an IP Access List to Filter IP Options TCP Flags Noncontiguous Ports

Before you perform any of the tasks in this module, you should be familiar with the information in the following modules:

- "IP Access List Overview"
- "Creating an IP Access List and Applying It to an Interface"

Information About Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports

IP Options

IP uses four key mechanisms in providing its service: Type of Service, Time to Live, Options, and Header Checksum.

The Options, commonly referred to as IP Options, provide for control functions that are required in some situations but unnecessary for the most common communications. IP Options include provisions for time stamps, security, and special routing.

IP Options may or may not appear in datagrams. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular datagram, not their implementation. In some environments the security option may be required in all datagrams.

The option field is variable in length. There may be zero or more options. IP Options can have one of two formats:

- Format 1: A single octet of option-type.
- Format 2: An option-type octet, an option-length octet, and the actual option-data octets.

The option-length octet counts the option-type octet, the option-length octet, and the option-data octets.

The option-type octet is viewed as having three fields: a 1-bit copied flag, a 2-bit option class, and a 5-bit option number. These fields form an 8-bit value for the option type field. IP Options are commonly referred to by their 8-bit value.

For a complete list and description of IP Options, refer to RFC 791, *Internet Protocol* at the following URL: http://www.faqs.org/rfcs/rfc791.html

Benefits of Filtering IP Options

- Filtering of packets that contain IP Options from the network relieves downstream devices and hosts of the load from options packets.
- This feature also minimizes load to the Route Processor (RP) for packets with IP Options that require RP processing on distributed systems. Previously, the packets were always routed to or processed by the RP CPU. Filtering the packets prevents them from impacting the RP.

Benefits of Filtering on TCP Flags

The ACL TCP Flags Filtering feature provides a flexible mechanism for filtering on TCP flags. Previously, an incoming packet was matched as long as any TCP flag in the packet matched a flag specified in the access control entry (ACE). This behavior allows for a security loophole, because packets with all flags set could get past the access control list (ACL). The ACL TCP Flags Filtering feature allows you to select any combination of flags on which to filter. The ability to match on a flag set and on a flag not set gives you a greater degree of control for filtering on TCP flags, thus enhancing security.

Because TCP packets can be sent as false synchronization packets that can be accepted by a listening port, it is recommended that administrators of firewall devices set up some filtering rules to drop false TCP packets.

The ACEs that make up an access list can be configured to detect and drop unauthorized TCP packets by allowing only the packets that have a very specific group of TCP flags set or not set. The ACL TCP Flags Filtering feature provides a greater degree of packet-filtering control in the following ways:

- You can select any desired combination of TCP flags on which to filter TCP packets.
- You can configure ACEs to allow matching on a flag that is set, as well as on a flag that is not set.

TCP Flags

The table below lists the TCP flags, which are further described in RFC 793, *Transmission Control Protocol*.

Table 125: TCP Flags

TCP Flag	Purpose
ACK	Acknowledge flag—Indicates that the acknowledgment field of a segment specifies the next sequence number the sender of this segment is expecting to receive.
FIN	Finish flag—Used to clear connections.
PSH	Push flag—Indicates the data in the call should be immediately pushed through to the receiving user.
RST	Reset flag—Indicates that the receiver should delete the connection without further interaction.
SYN	Synchronize flag—Used to establish connections.
URG	Urgent flag—Indicates that the urgent field is meaningful and must be added to the segment sequence number.

Benefits of Using the Named ACL Support for Noncontiguous Ports on an Access Control Entry Feature

This feature greatly reduces the number of access control entries (ACEs) required in an access control list to handle multiple entries for the same source address, destination address, and protocol. If you maintain large numbers of ACEs, use this feature to consolidate existing groups of access list entries wherever it is possible and when you create new access list entries. When you configure access list entries with noncontiguous ports, you will have fewer access list entries to maintain.

How Filtering on TTL Value Works

IP extended named and numbered access lists may filter on the TTL value of packets arriving at or leaving an interface. Packets with any possible TTL values 0 through 255 may be permitted or denied (filtered). Like filtering on other fields, such as source or destination address, the **ip access-group** command specifies **in** or **out** , which makes the access list ingress or egress and applies it to incoming or outgoing packets, respectively. The TTL value is checked in conjunction with the specified protocol, application, and any other settings in the access list entry, and all conditions must be met.

Special Handling for Packets with TTL Value of 0 or 1 Arriving at an Ingress Interface

The software switching paths—distributed Cisco Express Forwarding (dCEF), CEF, fast switching, and process switching—will usually permit or discard the packets based on the access list statements. However, when the TTL value of packets arriving at an ingress interface have a TTL of 0 or 1, special handling is required. The packets with a TTL value of 0 or 1 get sent to the process level before the ingress access list is

checked in CEF, dCEF, or the fast switching paths. The ingress access list is applied to packets with TTL values 2 through 255 and a permit or deny decision is made.

Packets with a TTL value of 0 or 1 are sent to the process level because they will never be forwarded out of the device; the process level must check whether each packet is destined for the device and whether an Internet Control Message Protocol (ICMP) TTL Expire message needs to be sent back. This means that even if an ACL with TTL value 0 or 1 filtering is configured on the ingress interface with the intention to drop packets with a TTL of 0 or 1, the dropping of the packets will not happen in the faster paths. It will instead happen in the process level when the process applies the ACL. This is also true for hardware switching platforms. Packets with TTL value of 0 or 1 are sent to the process level of the route processor (RP) or Multilayer Switch Feature Card (MSFC).

On egress interfaces, access list filtering on TTL value works just like other access list features. The check will happen in the fastest switching path enabled in the device. This is because the faster switching paths handle all the TTL values (0 through 255) equally on the egress interface.

Control Plane Policing for Filtering TTL Values 0 and 1

The special behavior for packets with a TTL value of 0 or 1 results in higher CPU usage for the device. If you are filtering on TTL value of 0 or 1, you should use control plane policing (CPP) to protect the CPU from being overwhelmed. In order to leverage CPP, you must configure an access list especially for filtering TTL values 0 and 1 and apply the access list through CPP. This access list will be a separate access list from any other interface access lists. Because CPP works for the entire system, not just on individual interfaces, you would need to configure only one such special access list for the entire device. This task is described in the section "Enabling Control Plane Policing to Filter on TTL Values 0 and 1".

Benefits of Filtering on TTL Value

- Filtering on time-to-live (TTL) value provides a way to control which packets are allowed to reach the device or are prevented from reaching the device. By looking at your network layout, you can choose whether to accept or deny packets from a certain device based on how many hops away it is. For example, in a small network, you can deny packets from a location more than three hops away. Filtering on TTL value allows you to validate if the traffic originated from a neighboring device. You can accept only packets that reach you in one hop, for example, by accepting only packets with a TTL value of one less than the initial TTL value of a particular protocol.
- Many control plane protocols communicate only with their neighbors, but receive packets from everyone. By applying an access list that filters on TTL to receiving routers, you can block unwanted packets.
- The Cisco software sends all packets with a TTL value of 0 or 1 to the process level. The device must then send an Internet Control Message Protocol (ICMP) TTL value expire message to the source. By filtering packets that have a TTL value of 0 through 2, you can reduce the load on the process level.

How to Create an IP Access List to Filter IP Options TCP Flags Noncontiguous Ports

Filtering Packets That Contain IP Options

Complete these steps to configure an access list to filter packets that contain IP options and to verify that the access list has been configured correctly.

Note

- The ACL Support for Filtering IP Options feature can be used only with named, extended ACLs.
- Resource Reservation Protocol (RSVP) Multiprotocol Label Switching Traffic Engineering (MPLS TE), Internet Group Management Protocol Version 2 (IGMPV2), and other protocols that use IP options packets may not function in drop or ignore mode if this feature is configured.
- On most Cisco devices, a packet with IP options is not switched in hardware, but requires control plane software processing (primarily because there is a need to process the options and rewrite the IP header), so all IP packets with IP options will be filtered and switched in software.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *access-list-name* **Example:** `Device(config)# ip access-list extended mylist1`	Specifies the IP access list by name and enters named access list configuration mode.
Step 4	[*sequence-number*] **deny** *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# deny ip any any option traceroute`	(Optional) Specifies a **deny** statement in named IP access list mode. • This access list happens to use a **deny** statement first, but a **permit** statement could appear first, depending on the order of statements you need. • Use the **option** keyword and *option-value* argument to filter packets that contain a particular IP Option.

	Command or Action	Purpose
		• In this example, any packet that contains the traceroute IP option will be filtered out. • Use the **no** *sequence-number* form of this command to delete an entry.
Step 5	[*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# permit ip any any option security`	Specifies a **permit** statement in named IP access list mode. • In this example, any packet (not already filtered) that contains the security IP option will be permitted. • Use the **no** *sequence-number* form of this command to delete an entry.
Step 6	Repeat Step 4 or Step 5 as necessary.	Allows you to revise the access list.
Step 7	**end** **Example:** `Device(config-ext-nacl)# end`	(Optional) Exits named access list configuration mode and returns to privileged EXEC mode.
Step 8	**show ip access-lists** *access-list-name* **Example:** `Device# show ip access-lists mylist1`	(Optional) Displays the contents of the IP access list.

What to Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.

Note To effectively eliminate all packets that contain IP Options, we recommend that you configure the global **ip options drop** command.

Filtering Packets That Contain TCP Flags

This task configures an access list to filter packets that contain TCP flags and verifies that the access list has been configured correctly.

Note

- TCP flag filtering can be used only with named, extended ACLs.
- The ACL TCP Flags Filtering feature is supported only for Cisco ACLs.
- Previously, the following command-line interface (CLI) format could be used to configure a TCP flag-checking mechanism:

permit tcp any any rst The following format that represents the same ACE can now be used: **permit tcp any any match-any +rst** Both the CLI formats are accepted; however, if the new keywords **match-all** or **match-any** are chosen, they must be followed by the new flags that are prefixed with " **+** " or " **-** ". It is advisable to use only the old format or the new format in a single ACL. You cannot mix and match the old and new CLI formats.

Caution If a device having ACEs with the new syntax format is reloaded with a previous version of the Cisco software that does not support the ACL TCP Flags Filtering feature, the ACEs will not be applied, leading to possible security loopholes.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *access-list-name* **Example:** `Device(config)# ip access-list extended kmd1`	Specifies the IP access list by name and enters named access list configuration mode.
Step 4	[*sequence-number*] **permit tcp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established** | { **match-any** | **match-all** } { **+** | **-** } *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# permit tcp any any match-any +rst`	Specifies a **permit** statement in named IP access list mode. • This access list happens to use a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • Use the TCP command syntax of the **permit** command.

	Command or Action	Purpose
		• Any packet with the RST TCP header flag set will be matched and allowed to pass the named access list kmd1 in Step 3.
Step 5	[*sequence-number*] **deny tcp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established** | { **match-any** | **match-all** } { + | - } *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# deny tcp any any match-all -ack -fin`	(Optional) Specifies a **deny** statement in named IP access list mode. • This access list happens to use a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • Use the TCP command syntax of the **deny** command. • Any packet that does not have the ACK flag set, and also does not have the FIN flag set, will not be allowed to pass the named access list kmd1 in Step 3. • See the **deny** (IP) command for additional command syntax to permit upper-layer protocols (ICMP, IGMP, TCP, and UDP).
Step 6	Repeat Step 4 or Step 5 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.	Allows you to revise the access list.
Step 7	**end** **Example:** `Device(config-ext-nacl)# end`	(Optional) Exits the configuration mode and returns to privileged EXEC mode.
Step 8	**show ip access-lists** *access-list-name* **Example:** `Device# show ip access-lists kmd1`	(Optional) Displays the contents of the IP access list. • Review the output to confirm that the access list includes the new entry.

Configuring an Access Control Entry with Noncontiguous Ports

Perform this task to create access list entries that use noncontiguous TCP or UDP port numbers. Although this task uses TCP ports, you could use the UDP syntax of the **permit** and **deny** commands to filter noncontiguous UDP ports.

Although this task uses a **permit** command first, use the **permit** and **deny** commands in the order that achieves your filtering goals.

Note The ACL—Named ACL Support for Noncontiguous Ports on an Access Control Entry feature can be used only with named, extended ACLs.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *access-list-name* **Example:** `Device(config)# ip access-list extended acl-extd-1`	Specifies the IP access list by name and enters named access list configuration mode.
Step 4	[*sequence-number*] **permit tcp** *source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established** { **match-any** | **match-all** } { **+** | **-** } *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# permit tcp any eq telnet ftp any eq 450 679`	Specifies a **permit** statement in named IP access list configuration mode. • Operators include **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range). • If the operator is positioned after the source and source-wildcard arguments, it must match the source port. If the operator is positioned after the destination and destination-wildcard arguments, it must match the destination port. • The **range** operator requires two port numbers. You can configure up to 10 ports after the **eq** and **neq** operators. All other operators require one port number. • To filter UDP ports, use the UDP syntax of this command.
Step 5	[*sequence-number*] **deny tcp** *source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established** { **match-any** | **match-all** } { **+** | **-** } *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]	(Optional) Specifies a **deny** statement in named access list configuration mode. • Operators include **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range).

	Command or Action	Purpose
	Example: `Device(config-ext-nacl)# deny tcp any neq 45 565 632`	• If the *operator* is positioned after the *source* and *source-wildcard* arguments, it must match the source port. If the *operator* is positioned after the *destination* and *destination-wildcard* arguments, it must match the destination port. • The **range** operator requires two port numbers. You can configure up to 10 ports after the **eq** and **neq** operators. All other operators require one port number. • To filter UDP ports, use the UDP syntax of this command.
Step 6	Repeat Step 4 or Step 5 as necessary, adding statements by sequence number where you planned. Use the **no** *sequence-number* command to delete an entry.	Allows you to revise the access list.
Step 7	**end** **Example:** `Device(config-ext-nacl)# end`	(Optional) Exits named access list configuration mode and returns to privileged EXEC mode.
Step 8	**show ip access-lists** *access-list-name* **Example:** `Device# show ip access-lists kmd1`	(Optional) Displays the contents of the access list.

Consolidating Access List Entries with Noncontiguous Ports into One Access List Entry

Perform this task to consolidate a group of access list entries with noncontiguous ports into one access list entry.

Although this task uses TCP ports, you could use the UDP syntax of the **permit** and **deny** commands to filter noncontiguous UDP ports.

Although this task uses a **permit** command first, use the **permit** and **deny** commands in the order that achieves your filtering goals.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show ip access-lists** *access-list-name* **Example:** `Device# show ip access-lists mylist1`	(Optional) Displays the contents of the IP access list. • Review the output to see if you can consolidate any access list entries.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**ip access-list extended** *access-list-name* **Example:** `Device(config)# ip access-list extended mylist1`	Specifies the IP access list by name and enters named access list configuration mode.
Step 5	**no** [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# no 10`	Removes the redundant access list entry that can be consolidated. • Repeat this step to remove entries to be consolidated because only the port numbers differ. • After this step is repeated to remove the access list entries 20, 30, and 40, for example, those entries are removed because they will be consolidated into one **permit** statement. • If a *sequence-number* is specified, the rest of the command syntax is optional.
Step 6	[*sequence-number*] **permit** *protocol source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator port* [*port*]] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# permit tcp any neq 45 565 632 any eq 23 45 34 43`	Specifies a **permit** statement in named access list configuration mode. • In this instance, a group of access list entries with noncontiguous ports was consolidated into one **permit** statement. • You can configure up to 10 ports after the **eq** and **neq** operators.
Step 7	Repeat Steps 5 and 6 as necessary, adding **permit** or **deny** statements to consolidate access list entries where possible. Use the **no** *sequence-number* command to delete an entry.	Allows you to revise the access list.
Step 8	**end** **Example:** `Device(config-std-nacl)# end`	(Optional) Exits named access list configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 9	**show ip access-lists** *access-list-name* **Example:** `Device# show ip access-lists mylist1`	(Optional) Displays the contents of the access list.

What To Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.

Filtering Packets Based on TTL Value

Because access lists are very flexible, it is not possible to define only one combination of **permit** and **deny** commands to filter packets based on the TTL value. This task illustrates just one example that achieves TTL filtering. Configure the appropriate **permit** and **deny** statements that will accomplish your filtering plan.

Note When the access list specifies the operation EQ or NEQ, depending on the Cisco software release in use on the device, the access lists can specify up to ten TTL values. The number of TTL values can vary by the Cisco software release.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *access-list-name* **Example:** `Device(config)# ip access-list extended ttlfilter`	Defines an IP access list by name. • An access list that filters on TTL value must be an extended access list.
Step 4	[*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard*[**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] [**log**] [**time-range** *time-range-name*] [**fragments**]	Sets conditions to allow a packet to pass a named IP access list. • Every access list must have at least one **permit** statement.

	Command or Action	Purpose
	Example: `Device(config-ext-nacl)# permit ip host 172.16.1.1 any ttl lt 2`	• This example permits packets from source 172.16.1.1 to any destination with a TTL value less than 2.
Step 5	Continue to add **permit** or **deny** statements to achieve the filtering you want.	--
Step 6	**exit** **Example:** `Device(config-ext-nacl)# exit`	Exits any configuration mode to the next highest mode in the command-line interface (CLI) mode hierarchy.
Step 7	**interface** *type number* **Example:** `Device(config)# interface TenGigabitEthernet4/1/0`	Configures an interface type and enters interface configuration mode.
Step 8	**ip access-group** *access-list-name* {**in** | **out**} **Example:** `Device(config-if)# ip access-group ttlfilter in`	Applies the access list to an interface.

Enabling Control Plane Policing to Filter on TTL Values 0 and 1

Perform this task to filter IP packets based on a TTL value of 0 or 1 and to protect the CPU from being overwhelmed. This task configures an access list for classification on TTL value 0 and 1, configures the Modular QoS Command-Line Interface (CLI) (MQC), and applies a policy map to the control plane. Any packets that pass the access list are dropped. This special access list is separate from any other interface access lists.

Because access lists are very flexible, it is not possible to define only one combination of **permit** and **deny** commands to filter packets based on the TTL value. This task illustrates just one example that achieves TTL filtering. Configure the appropriate **permit** and **deny** statements that will accomplish your filtering plan.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ip access-list extended** *access-list-name* **Example:** `Device(config)# ip access-list extended ttlfilter`	Defines an IP access list by name. • An access list that filters on a TTL value must be an extended access list.
Step 4	[*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* **ttl** *operator* value **Example:** `Device(config-ext-nacl)# permit ip host 172.16.1.1 any ttl lt 2`	Sets conditions to allow a packet to pass a named IP access list. • Every access list must have at least one **permit** statement. • This example permits packets from source 172.16.1.1 to any destination with a TTL value less than 2.
Step 5	Continue to add **permit** or **deny** statements to achieve the filtering you want.	The packets that pass the access list will be dropped.
Step 6	**exit** **Example:** `Device(config-ext-nacl)# exit`	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.
Step 7	**class-map** *class-map-name* [**match-all** | **match-any**] **Example:** `Device(config)# class-map acl-filtering`	Creates a class map to be used for matching packets to a specified class.
Step 8	**match access-group** { *access-group* | **name** *access-group-name* } **Example:** `Device(config-cmap)# match access-group name ttlfilter`	Configures the match criteria for a class map on the basis of the specified access control list.
Step 9	**exit** **Example:** `Device(config-cmap)# exit`	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.
Step 10	**policy-map** *policy-map-name* **Example:** `Device(config)# policy-map acl-filter`	Creates or modifies a policy map that can be attached to one or more interface to specify a service policy.

	Command or Action	Purpose
Step 11	**class** { *class-name* | **class-default** } **Example:** `Device(config-pmap)# class acl-filter-class`	Specifies the name of the class whose policy you want to create or change or to specify the default class (commonly known as the class-default class) before you configure its policy.
Step 12	**drop** **Example:** `Device(config-pmap-c)# drop`	Configures a traffic class to discard packets belonging to a specific class.
Step 13	**exit** **Example:** `Device(config-pmap-c)# exit`	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.
Step 14	**exit** **Example:** `Device(config-pmap)# exit`	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.
Step 15	**control-plane** **Example:** `Device(config)# control-plane`	Associates or modifies attributes or parameters that are associated with the control plane of the device.
Step 16	**service-policy** { **input** | **output** } *policy-map-name* **Example:** `Device(config-cp)# service-policy input acl-filter`	Attaches a policy map to a control plane for aggregate control plane services.

Configuration Examples for Filtering IP Options, TCP Flags, Noncontiguous Ports

Example: Filtering Packets That Contain IP Options

The following example shows an extended access list named mylist2 that contains access list entries (ACEs) that are configured to permit TCP packets only if they contain the IP Options that are specified in the ACEs:

```
ip access-list extended mylist2
 10 permit ip any any option eool
 20 permit ip any any option record-route
```

```
 30 permit ip any any option zsu
 40 permit ip any any option mtup
```

The **show access-list** command has been entered to show how many packets were matched and therefore permitted:

```
Device# show ip access-list mylist2
Extended IP access list test
10 permit ip any any option eool (1 match)
20 permit ip any any option record-route (1 match)
30 permit ip any any option zsu (1 match)
40 permit ip any any option mtup (1 match)
```

Example: Filtering Packets That Contain TCP Flags

The following access list allows TCP packets only if the TCP flags ACK and SYN are set and the FIN flag is not set:

```
ip access-list extended aaa
 permit tcp any any match-all +ack +syn -fin
 end
```

The **show access-list** command has been entered to display the ACL:

```
Device# show access-list aaa

Extended IP access list aaa
 10 permit tcp any any match-all +ack +syn -fin
```

Example: Creating an Access List Entry with Noncontiguous Ports

The following access list entry can be created because up to ten ports can be entered after the **eq** and **neq** operators:

```
ip access-list extended aaa
 permit tcp any eq telnet ftp any eq 23 45 34
 end
```

Enter the **show access-lists** command to display the newly created access list entry.

```
Device# show access-lists aaa

Extended IP access list aaa
 10 permit tcp any eq telnet ftp any eq 23 45 34
```

Example: Consolidating Some Existing Access List Entries into One Access List Entry with Noncontiguous Ports

The **show access-lists** command is used to display a group of access list entries for the access list named abc:

```
Device# show access-lists abc
Extended IP access list abc
```

```
 10 permit tcp any eq telnet any eq 450
 20 permit tcp any eq telnet any eq 679
 30 permit tcp any eq ftp any eq 450
 40 permit tcp any eq ftp any eq 679
```

Because the entries are all for the same **permit** statement and simply show different ports, they can be consolidated into one new access list entry. The following example shows the removal of the redundant access list entries and the creation of a new access list entry that consolidates the previously displayed group of access list entries:

```
ip access-list extended abc
 no 10
 no 20
 no 30
 no 40
 permit tcp any eq telnet ftp any eq 450 679
 end
```

When the **show access-lists** command is reentered, the consolidated access list entry is displayed:

```
Device# show access-lists abc
Extended IP access list abc
 10 permit tcp any eq telnet ftp any eq 450 679
```

Example: Filtering on TTL Value

The following access list filters IP packets containing type of service (ToS) level 3 with time-to-live (TTL) values 10 and 20. It also filters IP packets with a TTL greater than 154 and applies that rule to noninitial fragments. It permits IP packets with a precedence level of flash and a TTL value not equal to 1, and it sends log messages about such packets to the console. All other packets are denied.

```
ip access-list extended incomingfilter
 deny ip any any tos 3 ttl eq 10 20
 deny ip any any ttl gt 154 fragments
 permit ip any any precedence flash ttl neq 1 log
!
interface TenGigabitEthernet4/1/0


ip access-group incomingfilter in
```

Example: Control Plane Policing to Filter on TTL Values 0 and 1

The following example configures a traffic class called acl-filter-class for use in a policy map called acl-filter. An access list permits IP packets from any source having a time-to-live (TTL) value of 0 or 1. Any packets matching the access list are dropped. The policy map is attached to the control plane.

```
ip access-list extended ttlfilter


 permit ip any any ttl eq 0 1


class-map acl-filter-class


 match access-group name ttlfilter
```

```
policy-map acl-filter
 class acl-filter-class
 drop
control-plane
 service-policy input acl-filter
```

Additional References

Related Documents

Related Topic	Document Title
Security commands	*Cisco IOS Security Command Reference*
Configuring the device to drop or ignore packets containing IP Options by using the **no ip options** command.	*ACL IP Options Selective Drop*
Overview information about access lists.	*IP Access List Overview*
Information about creating an IP access list and applying it to an interface	*Creating an IP Access List and Applying It to an Interface*
QoS commands	*Cisco IOS Quality of Service Solutions Command Reference*

RFCs

RFC	Title
RFC 791	*Internet Protocol* http://www.faqs.org/rfcs/rfc791.html
RFC 793	*Transmission Control Protocol*
RFC 1393	*Traceroute Using an IP Option*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

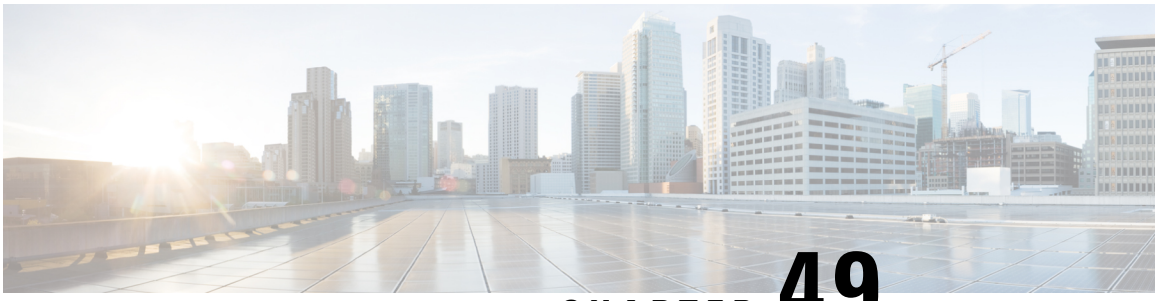
Feature Information for Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 126: Feature Information for Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports, or TTL Values

Feature Name	Releases	Feature Information
IP Access Lists	Cisco IOS-XE Release 3.15.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 49

Refining an IP Access List

There are several ways to refine an access list while or after you create it. You can change the order of the entries in an access list or add entries to an access list. You can restrict access list entries to a certain time of day or week, or achieve finer granularity when filtering packets by filtering noninitial fragments of packets.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 791
- Information About Refining an IP Access List, on page 792
- How to Refine an IP Access List, on page 795
- Configuration Examples for Refining an IP Access List, on page 800
- Additional References, on page 802
- Feature Information for Refining an IP Access List , on page 803

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 127: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About Refining an IP Access List

Access List Sequence Numbers

The ability to apply sequence numbers to IP access list entries simplifies access list changes. Prior to the IP Access List Entry Sequence Numbering feature, there was no way to specify the position of an entry within an access list. If you wanted to insert an entry in the middle of an existing list, all of the entries after the desired position had to be removed, then the new entry was added, and then all the removed entries had to be reentered. This method was cumbersome and error prone.

Sequence numbers allow users to add access list entries and resequence them. When you add a new entry, you specify the sequence number so that it is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced to create room to insert the new entry.

Benefits of Access List Sequence Numbers

An access list sequence number is a number at the beginning of a **permit** or **deny** command in an access list. The sequence number determines the order that the entry appears in the access list. The ability to apply sequence numbers to IP access list entries simplifies access list changes.

Prior to having sequence numbers, users could only add access list entries to the end of an access list; therefore, needing to add statements anywhere except the end of the list required reconfiguring the entire access list. There was no way to specify the position of an entry within an access list. If a user wanted to insert an entry (statement) in the middle of an existing list, all of the entries after the desired position had to be removed, then the new entry was added, and then all the removed entries had to be reentered. This method was cumbersome and error prone.

This feature allows users to add sequence numbers to access list entries and resequence them. When a user adds a new entry, the user chooses the sequence number so that it is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced to create room to insert the new entry. Sequence numbers make revising an access list much easier.

Sequence Numbering Behavior

- For backward compatibility with previous releases, if entries with no sequence numbers are applied, the first entry is assigned a sequence number of 10, and successive entries are incremented by 10. The maximum sequence number is 2147483647. If the generated sequence number exceeds this maximum number, the following message is displayed:

```
Exceeded maximum sequence number.
```

- If the user enters an entry without a sequence number, it is assigned a sequence number that is 10 greater than the last sequence number in that access list and is placed at the end of the list.
- If the user enters an entry that matches an already existing entry (except for the sequence number), then no changes are made.
- If the user enters a sequence number that is already present, the following error message is generated:

```
Duplicate sequence number.
```

- If a new access list is entered from global configuration mode, then sequence numbers for that access list are generated automatically.
- Sequence numbers are not nvgened. That is, the sequence numbers themselves are not saved. In the event that the system is reloaded, the configured sequence numbers revert to the default sequence starting number and increment. The function is provided for backward compatibility with software releases that do not support sequence numbering.
- This feature works with named and numbered, standard and extended IP access lists.

Benefits of Time Ranges

Benefits and possible uses of time ranges include the following:

- The network administrator has more control over permitting or denying a user access to resources. These resources could be an application (identified by an IP address/mask pair and a port number), policy routing, or an on-demand link (identified as interesting traffic to the dialer).
- Network administrators can set time-based security policy, including the following:
  - Perimeter security using access lists
  - Data confidentiality with IP Security Protocol (IPsec)

- When provider access rates vary by time of day, it is possible to automatically reroute traffic cost effectively.
- Network administrators can control logging messages. Access list entries can log traffic at certain times of the day, but not constantly. Therefore, administrators can simply deny access without needing to analyze many logs generated during peak hours.

Benefits Filtering Noninitial Fragments of Packets

Filter noninitial fragments of packets with an extended access list if you want to block more of the traffic you intended to block, not just the initial fragment of such packets. You should first understand the following concepts.

If the **fragments** keyword is used in additional IP access list entries that deny fragments, the fragment control feature provides the following benefits:

Additional Security

You are able to block more of the traffic you intended to block, not just the initial fragment of such packets. The unwanted fragments no longer linger at the receiver until the reassembly timeout is reached because they are blocked before being sent to the receiver. Blocking a greater portion of unwanted traffic improves security and reduces the risk from potential hackers.

Reduced Cost

By blocking unwanted noninitial fragments of packets, you are not paying for traffic you intended to block.

Reduced Storage

By blocking unwanted noninitial fragments of packets from ever reaching the receiver, that destination does not have to store the fragments until the reassembly timeout period is reached.

Expected Behavior Is Achieved

The noninitial fragments will be handled in the same way as the initial fragment, which is what you would expect. There are fewer unexpected policy routing results and fewer fragments of packets being routed when they should not be.

Access List Processing of Fragments

The behavior of access list entries regarding the use or lack of use of the **fragments** keyword can be summarized as follows:

If the Access-List Entry Has...	Then...

...no **fragments** keyword (the default), and assuming all of the access-list entry information matches,	For an access list entry that contains only Layer 3 information:
	• The entry is applied to nonfragmented packets, initial fragments, and noninitial fragments.
	For an access list entry that contains Layer 3 and Layer 4 information:
	• The entry is applied to nonfragmented packets and initial fragments.
	  • If the entry is a **permit** statement, then the packet or fragment is permitted.
	  • If the entry is a **deny** statement, then the packet or fragment is denied.
	• The entry is also applied to noninitial fragments in the following manner. Because noninitial fragments contain only Layer 3 information, only the Layer 3 portion of an access list entry can be applied. If the Layer 3 portion of the access list entry matches, and
	  • If the entry is a **permit** statement, then the noninitial fragment is permitted.
	  • If the entry is a **deny** statement, then the next access list entry is processed.
	Note The **deny** statements are handled differently for noninitial fragments versus nonfragmented or initial fragments.

...the **fragments** keyword, and assuming all of the access-list entry information matches,	The access list entry is applied only to noninitial fragments. The **fragments** keyword cannot be configured for an access list entry that contains any Layer 4 information.

Be aware that you should not add the **fragments** keyword to every access list entry because the first fragment of the IP packet is considered a nonfragment and is treated independently of the subsequent fragments. An initial fragment will not match an access list **permit** or **deny** entry that contains the **fragments** keyword. The packet is compared to the next access list entry, and so on, until it is either permitted or denied by an access list entry that does not contain the **fragments** keyword. Therefore, you may need two access list entries for every **deny** entry. The first **deny** entry of the pair will not include the **fragments** keyword and applies to the initial fragment. The second **deny** entry of the pair will include the **fragments** keyword and applies to the subsequent fragments. In the cases in which there are multiple **deny** entries for the same host but with different Layer 4 ports, a single **deny** access list entry with the **fragments** keyword for that host is all that needs to be added. Thus all the fragments of a packet are handled in the same manner by the access list.

Packet fragments of IP datagrams are considered individual packets, and each counts individually as a packet in access list accounting and access list violation counts.

How to Refine an IP Access List

The tasks in this module provide you with various ways to refine an access list if you did not already do so while you were creating it. You can change the order of the entries in an access list, add entries to an access

list, restrict access list entries to a certain time of day or week, or achieve finer granularity when filtering packets by filtering on noninitial fragments of packets.

Revising an Access List Using Sequence Numbers

Perform this task if you want to add entries to an existing access list, change the order of entries, or simply number the entries in an access list to accommodate future changes.

Note Remember that if you want to delete an entry from an access list, you can simply use the **no deny** or **no permit** form of the command, or the **no** *sequence-number* command if the statement already has a sequence number.

Note

- Access list sequence numbers do not support dynamic, reflexive, or firewall access lists.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list resequence** *access-list-name starting-sequence-number increment* **Example:** `Router(config)# ip access-list resequence kmd1 100 15`	Resequences the specified IP access list using the starting sequence number and the increment of sequence numbers. • This example resequences an access list named kmd1. The starting sequence number is 100 and the increment is 15.
Step 4	**ip access-list** { **standard** | **extended** } *access-list-name* **Example:** `Router(config)# ip access-list standard xyz123`	Specifies the IP access list by name and enters named access list configuration mode. • If you specify **standard** , make sure you specify subsequent **permit** and **deny** statements using the standard access list syntax. • If you specify **extended** , make sure you specify subsequent **permit** and **deny**

	Command or Action	Purpose
		statements using the extended access list syntax.
Step 5	Do one of the following: • *sequence-number* **permit** *source source-wildcard* • *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-std-nacl)# 105 permit 10.5.5.5 0.0.0.255`	Specifies a permit statement in named IP access list mode. • This access list happens to use a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • See the **permit** (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP). • Use the **no** *sequence-number* command to delete an entry. • As the prompt indicates, this access list was a standard access list. If you had specified **extended** in Step 4, the prompt for this step would be Router(config-ext-nacl)# and you would use the extended **permit** command syntax.
Step 6	Do one of the following: • *sequence-number* **deny** *source source-wildcard* • *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-std-nacl)# 110 deny 10.6.6.7 0.0.0.255`	(Optional) Specifies a deny statement in named IP access list mode. • This access list happens to use a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • See the **deny** (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP). • Use the **no** *sequence-number* command to delete an entry. • As the prompt indicates, this access list was a standard access list. If you had specified **extended** in Step 4, the prompt for this step would be Router(config-ext-nacl)# and you would use the extended **deny** command syntax.
Step 7	Repeat Step 5 and Step 6 as necessary, adding statements by sequence number where you	Allows you to revise the access list.

	Command or Action	Purpose
	planned. Use the **no** *sequence-number* command to delete an entry.	
Step 8	**end** **Example:** `Router(config-std-nacl)# end`	(Optional) Exits the configuration mode and returns to privileged EXEC mode.
Step 9	**show ip access-lists** *access-list-name* **Example:** `Router# show ip access-lists xyz123`	(Optional) Displays the contents of the IP access list. • Review the output to see that the access list includes the new entry.

Examples

The following is sample output from the **show ip access-lists** command when the **xyz123** access list is specified.

```
Router# show ip access-lists xyz123
Standard IP access list xyz123
100 permit 10.4.4.0, wildcard bits 0.0.0.255
105 permit 10.5.5.5, wildcard bits 0.0.0.255
115 permit 10.0.0.0, wildcard bits 0.0.0.255
130 permit 10.5.5.0, wildcard bits 0.0.0.255
145 permit 10.0.0.0, wildcard bits 0.0.0.255
```

Restricting an Access List Entry to a Time of Day or Week

By default, access list statements are always in effect once they are applied. However, you can define the times of the day or week that **permit** or **deny** statements are in effect by defining a time range, and then referencing the time range by name in an individual access list statement. IP and Internetwork Packet Exchange (IPX) named or numbered extended access lists can use time ranges.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip access-list extended** *name* **Example:** `Router(config)# ip access-list extended rstrct4`	Defines an extended IP access list using a name and enters extended named access list configuration mode.
Step 4	[*sequence-number*] **deny** *protocol source* [*source-wildcard*] [*operator port* [*port*]] *destination* [*destination-wildcard*] [*operator port* [*port*]] **Example:** `Router(config-ext-nacl)# deny ip any 172.20.1.1`	(Optional) Denies any packet that matches all of the conditions specified in the statement. • This statement will apply to nonfragmented packets and initial fragments.
Step 5	[*sequence-number*] **deny** *protocol source* [*source-wildcard*][*operator port* [*port*]] *destination* [*destination-wildcard*] [*operator port* [*port*]] **fragments** **Example:** `Router(config-ext-nacl)# deny ip any 172.20.1.1 fragments`	(Optional) Denies any packet that matches all of the conditions specified in the statement • This statement will apply to noninitial fragments.
Step 6	[*sequence-number*] **permit** *protocol source* [*source-wildcard*] [*operator port* [*port*]] *destination* [*destination-wildcard*] [*operator port* [*port*]] **Example:** `Router(config-ext-nacl)# permit tcp any any`	Permits any packet that matches all of the conditions specified in the statement. • Every access list needs at least one **permit** statement. • If the *source-wildcard* or *destination-wildcard* is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively. • Optionally use the keyword **any** as a substitute for the *source source-wildcard* or *destination destination-wildcard* to specify the address and wildcard of 0.0.0.0 255.255.255.255.
Step 7	Repeat some combination of Steps 4 through 6 until you have specified the values on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.
Step 8	**end** **Example:** `Router(config-ext-nacl)# end`	Ends configuration mode and returns the system to privileged EXEC mode.

	Command or Action	Purpose
Step 9	**show ip access-list** **Example:** `Router# show ip access-list`	(Optional) Displays the contents of all current IP access lists.

What to Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.

Note To effectively eliminate all packets that contain IP Options, we recommend that you configure the global **ip options drop** command.

Configuration Examples for Refining an IP Access List

Example Resequencing Entries in an Access List

The following example shows an access list before and after resequencing. The starting value is 1, and increment value is 2. The subsequent entries are ordered based on the increment values that users provide, and the range is from 1 to 2147483647.

When an entry with no sequence number is entered, by default it has a sequence number of 10 more than the last entry in the access list.

```
Router# show access-list carls
Extended IP access list carls
    10 permit ip host 10.3.3.3 host 172.16.5.34
    20 permit icmp any any
    30 permit tcp any host 10.3.3.3
    40 permit ip host 10.4.4.4 any
    50 Dynamic test permit ip any any
    60 permit ip host 172.16.2.2 host 10.3.3.12
    70 permit ip host 10.3.3.3 any log
    80 permit tcp host 10.3.3.3 host 10.1.2.2
    90 permit ip host 10.3.3.3 any
    100 permit ip any any
Router(config)# ip access-list extended carls
Router(config)# ip access-list resequence carls 1 2
Router(config)# end
Router# show access-list carls
Extended IP access list carls
    1 permit ip host 10.3.3.3 host 172.16.5.34
    3 permit icmp any any
    5 permit tcp any host 10.3.3.3
    7 permit ip host 10.4.4.4 any
    9 Dynamic test permit ip any any
    11 permit ip host 172.16.2.2 host 10.3.3.12
    13 permit ip host 10.3.3.3 any log
    15 permit tcp host 10.3.3.3 host 10.1.2.2
```

```
17 permit ip host 10.3.3.3 any
19 permit ip any any
```

Example Adding an Entry with a Sequence Number

In the following example, an new entry (sequence number 15) is added to an access list:

```
Router# show ip access-list
Standard IP access list tryon
2 permit 10.4.4.2, wildcard bits 0.0.255.255
5 permit 10.0.0.44, wildcard bits 0.0.0.255
10 permit 10.0.0.1, wildcard bits 0.0.0.255
20 permit 10.0.0.2, wildcard bits 0.0.0.255
Router(config)# ip access-list standard tryon
Router(config-std-nacl)# 15 permit 10.5.5.5 0.0.0.255
Router# show ip access-list
Standard IP access list tryon
2 permit 10.4.0.0, wildcard bits 0.0.255.255
5 permit 10.0.0.0, wildcard bits 0.0.0.255
10 permit 10.0.0.0, wildcard bits 0.0.0.255
15 permit 10.5.5.0, wildcard bits 0.0.0.255
20 permit 10.0.0.0, wildcard bits 0.0.0.255
```

Example Adding an Entry with No Sequence Number

The following example shows how an entry with no specified sequence number is added to the end of an access list. When an entry is added without a sequence number, it is automatically given a sequence number that puts it at the end of the access list. Because the default increment is 10, the entry will have a sequence number 10 higher than the last entry in the existing access list.

```
Router(config)# ip access-list standard resources
Router(config-std-nacl)# permit 10.1.1.1 0.0.0.255
Router(config-std-nacl)# permit 10.2.2.2 0.0.0.255
Router(config-std-nacl)# permit 10.3.3.3 0.0.0.255
Router# show access-list
Standard IP access list resources
10 permit 10.1.1.1, wildcard bits 0.0.0.255
20 permit 10.2.2.2, wildcard bits 0.0.0.255
30 permit 10.3.3.3, wildcard bits 0.0.0.255
Router(config)# ip access-list standard resources
Router(config-std-nacl)# permit 10.4.4.4 0.0.0.255
Router(config-std-nacl)# end
Router# show access-list
Standard IP access list resources
10 permit 10.1.1.1, wildcard bits 0.0.0.255
20 permit 10.2.2.2, wildcard bits 0.0.0.255
30 permit 10.3.3.3, wildcard bits 0.0.0.255
40 permit 10.4.4.4, wildcard bits 0.0.0.255
```

Example Time Ranges Applied to IP Access List Entries

The following example creates a time range called no-http, which extends from Monday to Friday from 8:00 a.m. to 6:00 p.m. That time range is applied to the **deny** statement, thereby denying HTTP traffic on Monday through Friday from 8:00 a.m. to 6:00 p.m.

The time range called udp-yes defines weekends from noon to 8:00 p.m. That time range is applied to the **permit** statement, thereby allowing UDP traffic on Saturday and Sunday from noon to 8:00 p.m. only. The access list containing both statements is applied to inbound packets on Ten Gigabit Ethernet interface 4/1/0.

```
time-range no-http
 periodic weekdays 8:00 to 18:00
!
time-range udp-yes
 periodic weekend 12:00 to 20:00
!
ip access-list extended strict
 deny tcp any any eq http time-range no-http
 permit udp any any time-range udp-yes
!
interface TenGigabitEthernet4/1/0
 ip access-group strict in
```

Example Filtering IP Packet Fragments

In the following access list, the first statement will deny only noninitial fragments destined for host 172.16.1.1. The second statement will permit only the remaining nonfragmented and initial fragments that are destined for host 172.16.1.1 TCP port 80. The third statement will deny all other traffic. In order to block noninitial fragments for any TCP port, we must block noninitial fragments for all TCP ports, including port 80 for host 172.16.1.1. That is, non-initial fragments will not contain Layer 4 port information, so, in order to block such traffic for a given port, we have to block fragments for all ports.

```
access-list 101 deny ip any host 172.16.1.1 fragments
access-list 101 permit tcp any host 172.16.1.1 eq 80
access-list 101 deny ip any any
```

Additional References

Related Documents

Related Topic	Document Title
Using the **time-range** command to establish time ranges	The chapter *Performing Basic System Management* in the *Cisco IOS XE Network Management Configuration Guide*
Network management command descriptions	*Cisco IOS Network Management Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

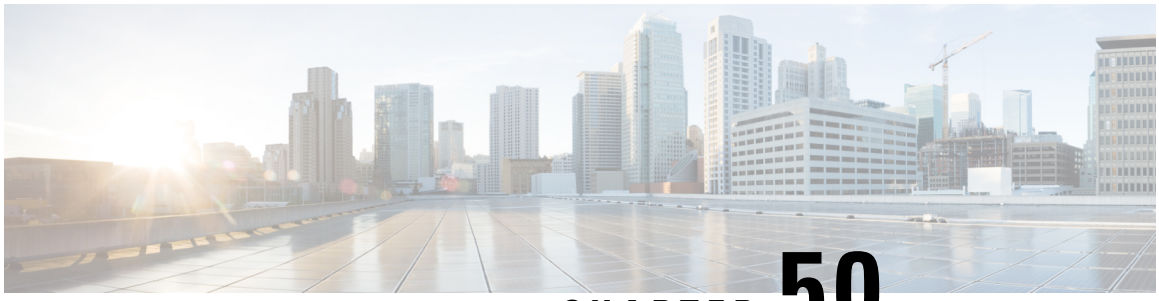
Feature Information for Refining an IP Access List

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 128: Feature Information for Refining an IP Access List

Feature Name	Releases	Feature Information
IP Access Lists	Cisco IOS-XE Release 3.15.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 50

IP Named Access Control Lists

Access control lists (ACLs) perform packet filtering to control the movement of packets through a network. Packet filtering provides security by limiting the access of traffic into a network, restricting user and device access to a network, and preventing traffic from leaving a network. IP access lists reduce the chance of spoofing and denial-of-service attacks, and allow dynamic, temporary user-access through a firewall.

The IP Named Access Control Lists feature gives network administrators the option of using names to identify their access lists.

This module describes IP named access lists and how to configure them.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 805
- Information About IP Named Access Control Lists, on page 806
- How to Configure IP Named Access Control Lists, on page 810
- Additional References for IP Named Access Control Lists, on page 813
- Feature Information for IP Named Access Control Lists, on page 813

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 129: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About IP Named Access Control Lists

Definition of an Access List

Access control lists (ACLs) perform packet filtering to control the movement of packets through a network. Packet filtering provides security by limiting the access of traffic into a network, restricting user and device access to a network, and preventing traffic from leaving a network. IP access lists reduce the chance of spoofing and denial-of-service attacks, and allow dynamic, temporary user-access through a firewall.

IP access lists can also be used for purposes other than security, such as to control bandwidth, restrict the content of routing updates, redistribute routes, trigger dial-on-demand (DDR) calls, limit debug output, and identify or classify traffic for quality of service (QoS) features.

An access list is a sequential list that consists of at least one **permit** statement and possibly one or more **deny** statements. In the case of IP access lists, these statements can apply to IP addresses, upper-layer IP protocols, or other fields in IP packets.

Access lists are identified and referenced by a name or a number. Access lists act as packet filters, filtering packets based on the criteria defined in each access list.

After you configure an access list, for the access list to take effect, you must either apply the access list to an interface (by using the **ip access-group** command), a vty (by using the **access-class** command), or reference

the access list by any command that accepts an access list. Multiple commands can reference the same access list.

In the following configuration, an IP access list named branchoffices is configured on Ten Gigabit Ethernet interface 4/1/0 and applied to incoming packets. Networks other than the ones specified by the source address and mask pair cannot access Ten Gigabit Ethernet interface 4/1/0. The destinations for packets coming from sources on network 172.16.7.0 are unrestricted. The destination for packets coming from sources on network 172.16.2.0 must be 172.31.5.4.

```
ip access-list extended branchoffices
 10 permit 172.16.7.0 0.0.0.3 any
 20 permit 172.16.2.0 0.0.0.255 host 172.31.5.4
!
interface TenGigabitEthernet4/1/0
 ip access-group branchoffices in
```

Named or Numbered Access Lists

All access lists must be identified by a name or a number. Named access lists are more convenient than numbered access lists because you can specify a meaningful name that is easier to remember and associate with a task. You can reorder statements in or add statements to a named access list.

Named access lists support the following features that are not supported by numbered access lists:

- IP options filtering
- Noncontiguous ports
- TCP flag filtering
- Deleting of entries with the **no permit** or **no deny** command

Note Not all commands that accept a numbered access list will accept a named access list. For example, vty uses only numbered access lists.

Benefits of IP Access Lists

Access control lists (ACLs) perform packet filtering to control the flow of packets through a network. Packet filtering can restrict the access of users and devices to a network, providing a measure of security. Access lists can save network resources by reducing traffic. The benefits of using access lists are as follows:

- Authenticate incoming rsh and rcp requests—Access lists can simplify the identification of local users, remote hosts, and remote users in an authentication database that is configured to control access to a device. The authentication database enables Cisco software to receive incoming remote shell (rsh) and remote copy (rcp) protocol requests.
- Block unwanted traffic or users—Access lists can filter incoming or outgoing packets on an interface, thereby controlling access to a network based on source addresses, destination addresses, or user authentication. You can also use access lists to determine the types of traffic that are forwarded or blocked at device interfaces. For example, you can use access lists to permit e-mail traffic to be routed through a network and to block all Telnet traffic from entering the network.

- Control access to vty—Access lists on an inbound vty (Telnet) can control who can access the lines to a device. Access lists on an outbound vty can control the destinations that the lines from a device can reach.
- Identify or classify traffic for QoS features—Access lists provide congestion avoidance by setting the IP precedence for Weighted Random Early Detection (WRED) and committed access rate (CAR). Access lists also provide congestion management for class-based weighted fair queueing (CBWFQ), priority queueing, and custom queueing.
- Limit debug command output—Access lists can limit debug output based on an IP address or a protocol.
- Provide bandwidth control—Access lists on a slow link can prevent excess traffic on a network.
- Provide NAT control—Access lists can control which addresses are translated by Network Address Translation (NAT).
- Reduce the chance of DoS attacks—Access lists reduce the chance of denial-of-service (DoS) attacks. Specify IP source addresses to control traffic from hosts, networks, or users from accessing your network. Configure the TCP Intercept feature to can prevent servers from being flooded with requests for connection.
- Restrict the content of routing updates—Access lists can control routing updates that are sent, received, or redistributed in networks.
- Trigger dial-on-demand calls—Access lists can enforce dial and disconnect criteria.

Access List Rules

The following rules apply to access lists:

- Only one access list per interface, per protocol, and per direction is allowed.
- An access list must contain at least one **permit** statement or all packets are denied entry into the network.
- The order in which access list conditions or match criteria are configured is important. While deciding whether to forward or block a packet, Cisco software tests the packet against each criteria statement in the order in which these statements are created. After a match is found, no more criteria statements are checked. The same **permit** or **deny** statements specified in a different order can result in a packet being passed under one circumstance and denied in another circumstance.
- If an access list is referenced by a name, but the access list does not exist, all packets pass. An interface or command with an empty access list applied to it permits all traffic into the network.
- Standard access lists and extended access lists cannot have the same name.
- Inbound access lists process packets before the packets are routed to an outbound interface. Inbound access lists that have filtering criteria that deny packet access to a network saves the overhead of routing lookup. Packets that are permitted access to a network based on the configured filtering criteria are processed for routing. For inbound access lists, when you configure a **permit** statement, packets are processed after they are received, and when you configure a **deny** statement, packets are discarded.
- Outbound access lists process packets before they leave the device. Incoming packets are routed to the outbound interface and then processed by the outbound access list. For outbound access lists, when you configure a **permit** statement, packets are sent to the output buffer, and when you configure a **deny** statement, packets are discarded.

- An access list can control traffic arriving at a device or leaving a device, but not traffic originating at a device.

Helpful Hints for Creating IP Access Lists

The following tips will help you avoid unintended consequences and help you create more efficient, useful access lists.

- Create the access list before applying it to an interface (or elsewhere), because if you apply a nonexistent access list to an interface and then proceed to configure the access list, the first statement is put into effect, and the implicit **deny** statement that follows could cause you immediate access problems.
- Another reason to configure an access list before applying it is because an interface with an empty access list applied to it permits all traffic.
- All access lists need at least one **permit** statement; otherwise, all packets are denied and no traffic passes.
- Because the software stops testing conditions after it encounters the first match (to either a **permit** or **deny** statement), you will reduce processing time and resources if you put the statements that packets are most likely to match at the beginning of the access list. Place more frequently occurring conditions before less frequent conditions.
- Organize your access list so that more specific references in a network or subnet appear before more general ones.
- Use the statement **permit any any** if you want to allow all other packets not already denied. Using the statement **permit any any** in effect avoids denying all other packets with the implicit deny statement at the end of an access list. Do not make your first access list entry **permit any any** because all traffic will get through; no packets will reach the subsequent testing. In fact, once you specify **permit any any** , all traffic not already denied will get through.
- Although all access lists end with an implicit **deny** statement, we recommend use of an explicit **deny** statement (for example, **deny ip any any**). On most platforms, you can display the count of packets denied by issuing the **show access-list** command, thus finding out more information about who your access list is disallowing. Only packets denied by explicit **deny** statements are counted, which is why the explicit **deny** statement will yield more complete data for you.
- While you are creating an access list or after it is created, you might want to delete an entry.
  - You cannot delete an entry from a numbered access list; trying to do so will delete the entire access list. If you need to delete an entry, you need to delete the entire access list and start over.
  - You can delete an entry from a named access list. Use the **no permit** or **no deny** command to delete the appropriate entry.
- In order to make the purpose of individual statements more scannable and easily understood at a glance, you can write a helpful remark before or after any statement by using the **remark** command.
- If you want to deny access to a particular host or network and find out if someone from that network or host is attempting to gain access, include the **log** keyword with the corresponding **deny** statement so that the packets denied from that source are logged for you.
- This hint applies to the placement of your access list. When trying to save resources, remember that an inbound access list applies the filter conditions before the routing table lookup. An outbound access list applies the filter conditions after the routing table lookup.

Where to Apply an Access List

You can apply access lists to the inbound or outbound interfaces of a device. Applying an access list to an inbound interface controls the traffic that enters the interface and applying an access list to an outbound interface controls the traffic that exits the interface.

When software receives a packet at the inbound interface, the software checks the packet against the statements that are configured for the access list. If the access list permits packets, the software processes the packet. Applying access lists to filter incoming packets can save device resources because filtered packets are discarded before entering the device.

Access lists on outbound interfaces filter packets that are transmitted (sent) out of the interface. You can use the TCP Access Control List (ACL) Splitting feature of the Rate-Based Satellite Control Protocol (RBSCP) on the outbound interface to control the type of packets that are subject to TCP acknowledgment (ACK) splitting on an outbound interface.

You can reference an access list by using a **debug** command to limit the amount of debug logs. For example, based on the filtering or matching criteria of the access list, debug logs can be limited to source or destination addresses or protocols.

You can use access lists to control routing updates, dial-on-demand (DDR), and quality of service (QoS) features.

How to Configure IP Named Access Control Lists

Creating an IP Named Access List

You can create an IP named access list to filter source addresses and destination addresses or a combination of addresses and other IP fields. Named access lists allow you to identify your access lists with an intuitive name.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list extended** *name* **Example:** `Device(config)# ip access-list extended acl1`	Defines an extended IP access list using a name and enters extended named access list configuration mode.

	Command or Action	Purpose
Step 4	**remark** *remark* **Example:** `Device(config-ext-nacl)# remark protect server by denying sales access to the acl1 network`	(Optional) Adds a description for an access list statement. • A remark can precede or follow an IP access list entry. • In this example, the **remark** command reminds the network administrator that the **deny** command configured in Step 5 denies the Sales network access to the interface.
Step 5	**deny** *protocol* [*source source-wildcard*] {**any** | **host** {*address* | *name*} {*destination* [*destination-wildcard*] {**any** | **host** {*address* | *name*} [**log**] **Example:** `Device(config-ext-nacl)# deny ip 192.0.2.0 0.0.255.255 host 192.0.2.10 log`	(Optional) Denies all packets that match all conditions specified by the remark.
Step 6	**remark** *remark* **Example:** `Device(config-ext-nacl)# remark allow TCP from any source to any destination`	(Optional) Adds a description for an access list statement. • A remark can precede or follow an IP access list entry.
Step 7	**permit** *protocol* [*source source-wildcard*] {**any** | **host** {*address* | *name*} {*destination* [*destination-wildcard*] {**any** | **host** {*address* | *name*} [**log**] **Example:** `Device(config-ext-nacl)# permit tcp any any`	Permits all packets that match all conditions specified by the statement.
Step 8	Repeat Steps 4 through 7 to specify more statements for your access list.	**Note** All source addresses that are not specifically permitted by a statement are denied by an implicit deny statement at the end of the access list.
Step 9	**end** **Example:** `Device(config-ext-nacl)# end`	Exits extended named access list configuration mode and returns to privileged EXEC mode.
Step 10	**show ip access-lists** **Example:** `Device# show ip access-lists`	Displays the contents of all current IP access lists.

Example:

The following is sample output from the **show ip access-lists** command:

```
Device# show ip access-lists acl1

Extended IP access list acl1
   permit tcp any 192.0.2.0 255.255.255.255 eq telnet
   deny tcp any any
   deny udp any 192.0.2.0 255.255.255.255 lt 1024
   deny ip any any log
```

Applying an Access List to an Interface

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface TenGigabitEthernet4/1/0`	Specifies an interface and enters interface configuration mode.
Step 4	**ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**} **Example:** `Device(config-if)# ip access-group acl1 in`	Applies the specified access list to the inbound interface. • To filter source addresses, apply the access list to the inbound interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Additional References for IP Named Access Control Lists

Related Documents

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

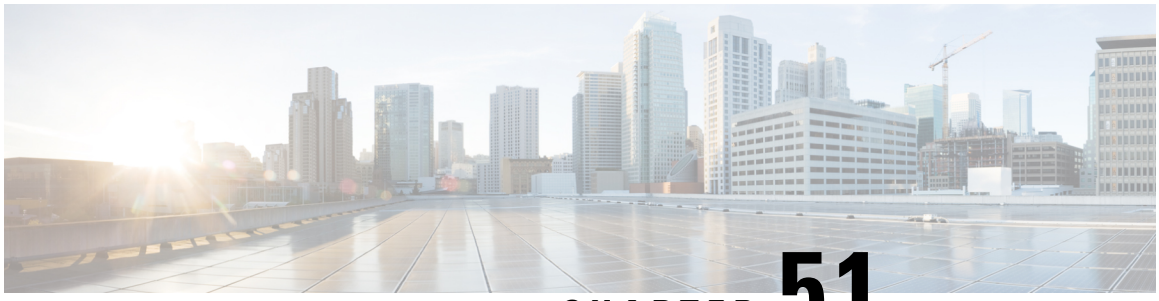
Feature Information for IP Named Access Control Lists

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 130: Feature Information for IP Named Access Control Lists

Feature Name	Releases	Feature Information
IP Access Lists	Cisco IOS-XE Release 3.15.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 51

IPv4 ACL Chaining Support

ACL Chaining, also known as Multi-Access Control List, allows you to split access control lists (ACLs). This module describes how with the IPv4 ACL Chaining Support feature, you can explicitly split ACLs into common and user-specific ACLs and bind both ACLs to a target for traffic filtering on a device. In this way, the common ACLs in Ternary Content Addressable Memory (TCAM) are shared by multiple targets, thereby reducing the resource usage.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 815
- Restrictions for IPv4 ACL Chaining Support, on page 816
- Information About IPv4 ACL Chaining Support, on page 816
- How to Configure IPv4 ACL Chaining Support, on page 817
- Configuration Examples for IPv4 ACL Chaining Support, on page 818
- Additional References for IPv4 ACL Chaining Support, on page 819
- Feature Information for IPv4 ACL Chaining Support , on page 820

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 131: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for IPv4 ACL Chaining Support

- A single access control List (ACL) cannot be used for both common and regular ACLs for the same target in the same direction.
- ACL chaining applies to only security ACLs. It is not supported for feature policies, such as Quality of Service (QoS), Firewall Services Module (FW) and Policy Based Routing (PBR).
- Per-target statistics are not supported for common ACLs.

Information About IPv4 ACL Chaining Support

ACL Chaining Overview

The packet filter process supports only a single Access control list (ACL) to be applied per direction and per protocol on an interface. This leads to manageability and scalability issues if there are common ACL entries needed on many interfaces. Duplicate Access control entries (ACEs) are configured for all those interfaces, and any modification to the common ACEs needs to be performed for all ACLs.

A typical ACL on the edge box for an Internet Service Provider (ISP) has two sets of ACEs:

- Common ISP specific ACEs
- Customer/interface specific ACEs

The purpose of these address blocks is to deny access to ISP's protected infrastructure networks and anti-spoofing protection by allowing only customer source address blocks. This results in configuring unique ACL per interface and most of the ACEs being common across all ACLs on a device. ACL provisioning and modification is very cumbersome, hence, any changes to the ACE impacts every target.

IPv4 ACL Chaining Support

IPv4 ACL Chaining Support allows you to split the Access control list (ACL) into common and customer-specific ACLs and attach both ACLs to a common session. In this way, only one copy of the common ACL is attached to Ternary Content Addressable Memory (TCAM) and shared by all users, thereby making it easier to maintain the common ACEs.

The IPv4 ACL Chaining feature allows two IPV4 ACLs to be active on an interface per direction:

- Common
- Regular
- Common and Regular

Note If you configure both common and regular ACLs on an interface, the common ACL is considered over a regular ACL.

How to Configure IPv4 ACL Chaining Support

ACL chaining is supported by extending the **ip traffic filter** command.

The **ip traffic filter** command is not additive. When you use this command, it replaces earlier instances of the command.

For more information, refer to the *IPv6 ACL Chaining with a Common ACL* section in the Security Configuration Guide: Access Control Lists Configuration Guide.

Configuring an Interface to Accept Common ACL

Perform this task to configure the interface to accept a common Access control list (ACL) along with an interface-specific ACL:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface TenGigabitEthernet4/1/0`	Configures an interface and enters the interface configuration mode.
Step 4	**ip access-group** {**common** {*common-access-list-name* {*regular-access-list* | **acl**}} {**in** | **out**}} **Example:** `Device(config-if)# ipv4 access-group common acl-p acl1 in`	Configures the interface to accept a common ACL along with the interface-specific ACL.
Step 5	**end** **Example:** `Device(config-if)# end`	(Optional) Exits the configuration mode and returns to privileged EXEC mode.

Configuration Examples for IPv4 ACL Chaining Support

This section provides configuration examples of Common Access Control List (ACL).

Example: Configuring an Interface to Accept a Common ACL

This example shows how to replace an Access Control List (ACL) configured on the interface without explicitly deleting the ACL:

```
interface TenGigabitEthernet4/1/0
ipv4 access-group common  C_acl ACL1 in
end
replace interface acl ACL1 by ACL2
interface TenGigabitEthernet4/1/0
ipv4 access-group common  C_acl ACL2 in
end
```

This example shows how common ACL cannot be replaced on interfaces without deleting it explicitly from the interface:

```
interface TenGigabitEthernet4/1/0
ipv4 access-group common  C_acl1 ACL1 in
end
change the common acl to C_acl2
interface TenGigabitEthernet4/1/0
no ipv4 access-group common  C_acl1  ACL1 in
end
```

```
interface TenGigabitEthernet4/1/0
ipv4 access-group common  C_acl2  ACL1 in
end
```

Note When reconfiguring a common ACL, you must ensure that no other interface on the line card is attached to the common ACL.

Note If both common ACL and interface ACL are attached to an interface and only one of the above is reconfigured on the interface, then the other is removed automatically.

This example shows how the interface ACL is removed:

```
interface TenGigabitEthernet4/1/0
ipv4 access-group common C_acl1 ACL1 in
end
```

Additional References for IPv4 ACL Chaining Support

Related Documents

Related Topic	Document Title
IPv6 ACL Chaining Support	Security Configuration Guide: Access Control Lists
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

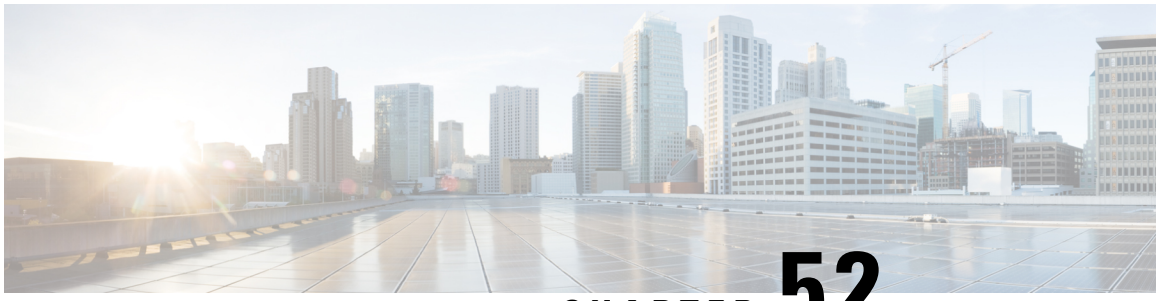
Feature Information for IPv4 ACL Chaining Support

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 132: Feature Information for IPv4 ACL Chaining Support

Feature Name	Releases	Feature Information
IP Access Lists	Cisco IOS-XE Release 3.15.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 52

IPv6 ACL Chaining with a Common ACL

ACL Chaining, also known as Multi-Access Control List (ACL), allows you to split ACLs. This document describes how with the IPv6 ACL Chaining Support feature, you can explicitly split ACLs into common and user-specific ACLs and bind both ACLs to a target for traffic filtering on a device. In this way, the common ACLs in Ternary Content Addressable Memory (TCAM) are shared by multiple targets, thereby reducing the resource usage.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 821
- Information About IPv6 ACL Chaining with a Common ACL, on page 822
- How to Configure IPv6 ACL Chaining with a Common ACL, on page 823
- Configuration Examples for IPv6 ACL Chaining with a Common ACL, on page 824
- Additional References for IPv6 ACL Chaining with a Common ACL, on page 825
- Feature Information for IPv6 ACL Chaining with a Common ACL , on page 826

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 133: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About IPv6 ACL Chaining with a Common ACL

ACL Chaining Overview

The packet filter process supports only a single Access control list (ACL) to be applied per direction and per protocol on an interface. This leads to manageability and scalability issues if there are common ACL entries needed on many interfaces. Duplicate Access control entries (ACEs) are configured for all those interfaces, and any modification to the common ACEs needs to be performed for all ACLs.

A typical ACL on the edge box for an Internet Service Provider (ISP) has two sets of ACEs:

- Common ISP specific ACEs
- Customer/interface specific ACEs

The purpose of these address blocks is to deny access to ISP's protected infrastructure networks and anti-spoofing protection by allowing only customer source address blocks. This results in configuring unique ACL per interface and most of the ACEs being common across all ACLs on a device. ACL provisioning and modification is very cumbersome, hence, any changes to the ACE impacts every target.

IPv6 ACL Chaining with a Common ACL

With IPv6 ACL Chaining, you can configure a traffic filter with the following:

- Common ACL
- Specific ACL
- Common and Specific ACL

Each Access control list (ACL) is matched in a sequence. For example, if you have specified both the ACLs - a common and a specific ACL, the packet is first matched against the common ACL; if a match is not found, it is then matched against the specific ACL.

Note Any IPv6 ACL may be configured on a traffic filter as a common or specific ACL. However, the same ACL cannot be specified on the same traffic filter as both common and specific.

How to Configure IPv6 ACL Chaining with a Common ACL

Before you begin

IPv6 ACL chaining is configured on an interface using an extension of the existing IPv6 traffic-filter command: **ipv6 traffic-filter** [**common** *common-acl*] [*specific-acl*] [**in** | **out**]

Note You may choose to configure either of the following:

- Only a common ACL. For example: **ipv6 traffic-filter common** *common-acl*
- Only a specific ACL. For example: **ipv6 traffic-filter** *common-acl*
- Both ACLs. For example: **ipv6 traffic-filter common** *common-acl specific-acl*

The ipv6 traffic-filter command is not additive. When you use the command, it replaces earlier instances of the command. For example, the command sequence: **ipv6 traffic-filter** [**common** *common-acl*] [*specific-acl*] **in ipv6 traffic-filter** [*specific-acl*] **in** binds a common ACL to the traffic filter, removes the common ACL and then binds a specific ACL.

Configuring IPv6 ACL to an Interface

Perform this task to configure the interface to accept a common access control list (ACL) along with an interface-specific ACL:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface TenGigabitEthernet4/1/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 4	**ipv6 traffic filter** {*common-access-list-name* {**in** | **out**}} **Example:** `Device(config)# ipv6 traffic-filter outbound out`	Applies the specified IPv6 access list to the interface specified in the previous step.
Step 5	**end** **Example:** `Device(config-if)# end`	(Optional) Exits the configuration mode and returns to privileged EXEC mode.

Configuration Examples for IPv6 ACL Chaining with a Common ACL

You may configure the following combinations in no particular order:

- A common ACL, for example: **ipv6 traffic-filter common** *common-acl* **in**
- A specific ACL, for example: **ipv6 traffic-filter** *specific-acl* **in**
- Both ACLs, for example: **ipv6 traffic-filter common** *common-acl* *specific-acl* **in**

Example: Configuring an Interface to Accept a Common ACL

This example shows how to replace an access control list (ACL) configured on the interface without explicitly deleting the ACL:

```
interface TenGigabitEthernet4/1/0
ipv6 access-group common  C_acl ACL1 in
end
replace interface acl ACL1 by ACL2
interface TenGigabitEthernet4/1/0
ipv6 access-group common  C_acl ACL2 in
end
```

This example shows how to delete a common ACL from an interface. A common ACL cannot be replaced on interfaces without deleting it explicitly from the interface.

```
interface TenGigabitEthernet4/1/0
ipv6 access-group common  C_acl1 ACL1 in
end
change the common acl to C_acl2
interface TenGigabitEthernet4/1/0
no ipv6 access-group common  C_acl1  ACL1 in
end
interface TenGigabitEthernet4/1/0
ipv6 access-group common  C_acl2  ACL1 in
end
```

Note When reconfiguring a common ACL, you must ensure that no other interface on the line card is attached to the common ACL.

Note If both common ACL and interface ACL are attached to an interface and only one of the above is reconfigured on the interface, then the other is removed automatically.

This example shows how to remove the interface ACL:

```
interface TenGigabitEthernet4/1/0
ipv6 access-group common C_acl1 ACL1 in
end
```

Additional References for IPv6 ACL Chaining with a Common ACL

Related Documents

Related Topic	Document Title
IPv4 ACL Chaining Support	Security Configuration Guide: Access Control Lists, Cisco IOS XE Release 3S
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

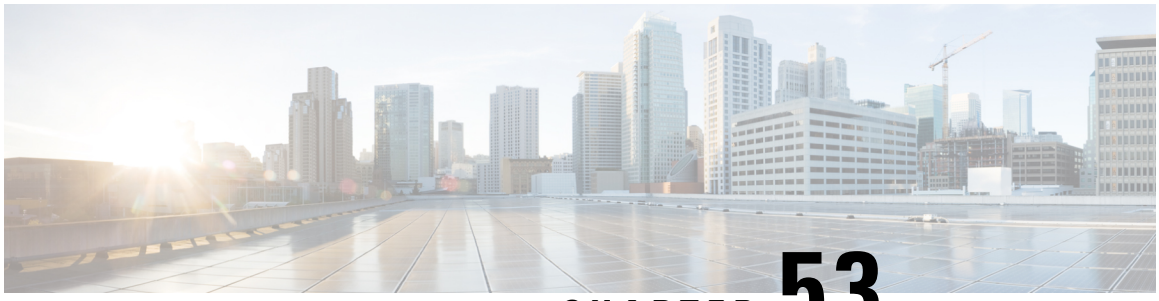
Feature Information for IPv6 ACL Chaining with a Common ACL

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 134: Feature Information for IPv6 ACL Chaining with a Common ACL

Feature Name	Releases	Feature Information
IPv6 Access Lists	Cisco IOS-XE Release 3.15.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 53

Commented IP Access List Entries

The Commented IP Access List Entries feature allows you to include comments or remarks about **deny** or **permit** conditions in any IP access list. These remarks make access lists easier for network administrators to understand. Each remark is limited to 100 characters in length.

This module provides information about the Commented IP Access List Entries feature.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 827
- Information About Commented IP Access List Entries, on page 828
- How to Configure Commented IP Access List Entries, on page 829
- Additional References for Commented IP Access List Entries, on page 830
- Feature Information for Commented IP Access List Entries , on page 831

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 135: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About Commented IP Access List Entries

Benefits of IP Access Lists

Access control lists (ACLs) perform packet filtering to control the flow of packets through a network. Packet filtering can restrict the access of users and devices to a network, providing a measure of security. Access lists can save network resources by reducing traffic. The benefits of using access lists are as follows:

- Authenticate incoming rsh and rcp requests—Access lists can simplify the identification of local users, remote hosts, and remote users in an authentication database that is configured to control access to a device. The authentication database enables Cisco software to receive incoming remote shell (rsh) and remote copy (rcp) protocol requests.
- Block unwanted traffic or users—Access lists can filter incoming or outgoing packets on an interface, thereby controlling access to a network based on source addresses, destination addresses, or user authentication. You can also use access lists to determine the types of traffic that are forwarded or blocked at device interfaces. For example, you can use access lists to permit e-mail traffic to be routed through a network and to block all Telnet traffic from entering the network.
- Control access to vty—Access lists on an inbound vty (Telnet) can control who can access the lines to a device. Access lists on an outbound vty can control the destinations that the lines from a device can reach.

- Identify or classify traffic for QoS features—Access lists provide congestion avoidance by setting the IP precedence for Weighted Random Early Detection (WRED) and committed access rate (CAR). Access lists also provide congestion management for class-based weighted fair queueing (CBWFQ), priority queueing, and custom queueing.
- Limit debug command output—Access lists can limit debug output based on an IP address or a protocol.
- Provide bandwidth control—Access lists on a slow link can prevent excess traffic on a network.
- Provide NAT control—Access lists can control which addresses are translated by Network Address Translation (NAT).
- Reduce the chance of DoS attacks—Access lists reduce the chance of denial-of-service (DoS) attacks. Specify IP source addresses to control traffic from hosts, networks, or users from accessing your network. Configure the TCP Intercept feature to can prevent servers from being flooded with requests for connection.
- Restrict the content of routing updates—Access lists can control routing updates that are sent, received, or redistributed in networks.
- Trigger dial-on-demand calls—Access lists can enforce dial and disconnect criteria.

Access List Remarks

You can include comments or remarks about entries in any IP access list. An access list remark is an optional remark before or after an access list entry that describes the entry so that you do not have to interpret the purpose of the entry. Each remark is limited to 100 characters in length.

The remark can go before or after a **permit** or **deny** statement. Be consistent about where you add remarks. Users may be confused if some remarks precede the associated **permit** or **deny** statements and some remarks follow the associated statements.

The following is an example of a remark that describes function of the subsequent **deny** statement:

```
ip access-list extended telnetting
 remark Do not allow host1 subnet to telnet out
 deny tcp host 172.16.2.88 any eq telnet
```

How to Configure Commented IP Access List Entries

Writing Remarks in a Named or Numbered Access List

You can use a named or numbered access list configuration. You must apply the access list to an interface or terminal line after the access list is created for the configuration to work.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list** {**standard** | **extended**} {*name* | *number*} **Example:** `Device(config)# ip access-list extended telnetting`	Identifies the access list by a name or number and enters extended named access list configuration mode.
Step 4	**remark** *remark* **Example:** `Device(config-ext-nacl)# remark Do not allow host1 subnet to telnet out`	Adds a remark for an entry in a named IP access list. • The remark indicates the purpose of the **permit** or **deny** statement.
Step 5	**deny** *protocol* **host** *host-address* **any eq** *port* **Example:** `Device(config-ext-nacl)# deny tcp host 172.16.2.88 any eq telnet`	Sets conditions in a named IP access list that denies packets.
Step 6	**end** **Example:** `Device(config-ext-nacl)# end`	Exits extended named access list configuration mode and enters privileged EXEC mode.

Additional References for Commented IP Access List Entries

Related Documents

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

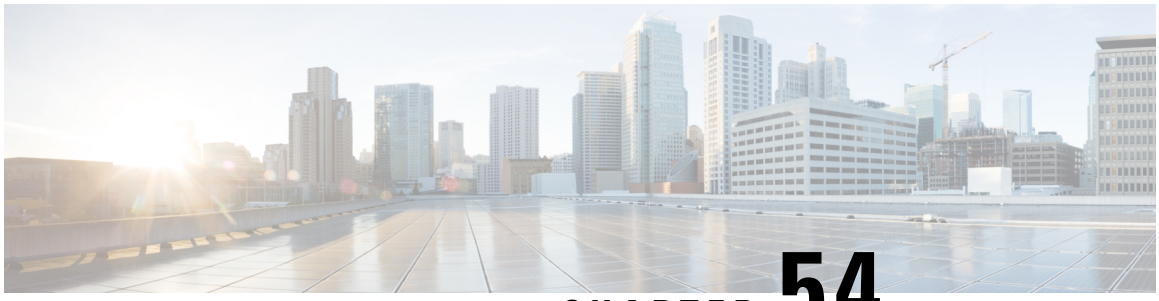
Feature Information for Commented IP Access List Entries

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 136: Feature Information for Commented IP Access List Entries

Feature Name	Releases	Feature Information
IP Access Lists	Cisco IOS-XE Release 3.15.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 54

Standard IP Access List Logging

The Standard IP Access List Logging feature provides the ability to log messages about packets that are permitted or denied by a standard IP access list. Any packet that matches the access list logs an information message about the packet at the device console.

This module provides information about standard IP access list logging.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 833
- Restrictions for Standard IP Access List Logging, on page 834
- Information About Standard IP Access List Logging, on page 834
- How to Configure Standard IP Access List Logging, on page 835
- Configuration Examples for Standard IP Access List Logging, on page 837
- Additional References for Standard IP Access List Logging, on page 837
- Feature Information for Standard IP Access List Logging , on page 838

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 137: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for Standard IP Access List Logging

IP access list logging is supported only for routed interfaces or router access control lists (ACLs).

Information About Standard IP Access List Logging

Standard IP Access List Logging

The Standard IP Access List Logging feature provides the ability to log messages about packets that are permitted or denied by a standard IP access list. Any packet that matches the access list causes an information log message about the packet to be sent to the device console. The log level of messages that are printed to the device console is controlled by the **logging console** command.

The first packet that the access list inspects triggers the access list to log a message at the device console. Subsequent packets are collected over 5-minute intervals before they are displayed or logged. Log messages include information about the access list number, the source IP address of packets, the number of packets from the same source that were permitted or denied in the previous 5-minute interval, and whether a packet was permitted or denied. You can also monitor the number of packets that are permitted or denied by a particular access list, including the source address of each packet.

How to Configure Standard IP Access List Logging

Creating a Standard IP Access List Using Numbers

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} **host** *address* [**log**] **Example:** `Device(config)# access-list 1 permit host 10.1.1.1 log`	Defines a standard numbered IP access list using a source address and wildcard, and configures the logging of informational messages about packets that match the access list entry at the device console.
Step 4	**access-list** *access-list-number* {**deny** | **permit**} **any** [**log**] **Example:** `Device(config)# access-list 1 permit any log`	Defines a standard numbered IP access list by using an abbreviation for the source and source mask 0.0.0.0 255.255.255.255.
Step 5	**interface** *type number* **Example:** `Device(config)# interface TenGigabitEthernet4/1/0`	Configures an interface and enters interface configuration mode.
Step 6	**ip access-group** *access-list-number* {**in** | **out**} **Example:** `Device(config-if)# ip access-group 1 in`	Applies the specified numbered access list to the incoming or outgoing interface. • When you filter based on source addresses, you typically apply the access list to an incoming interface.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Creating a Standard IP Access List Using Names

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list standard** *name* **Example:** `Device(config)# ip access-list standard acl1`	Defines a standard IP access list and enters standard named access list configuration mode.
Step 4	{**deny** | **permit**} {**host** *address* | **any**} **log** **Example:** `Device(config-std-nacl)# permit host 10.1.1.1 log`	Sets conditions in a named IP access list that will deny packets from entering a network or permit packets to enter a network, and configures the logging of informational messages about packets that match the access list entry at the device console.
Step 5	**exit** **Example:** `Device(config-std-nacl)# exit`	Exits standard named access list configuration mode and enters global configuration mode.
Step 6	**interface** *type number* **Example:** `Device(config)# interface TenGigabitEthernet4/1/0`	Configures an interface and enters interface configuration mode.
Step 7	**ip access-group** *access-list-name* {**in** | **out**} **Example:** `Device(config-if)# ip access-group acl1 in`	Applies the specified access list to the incoming or outgoing interface. • When you filter based on source addresses, you typically apply the access list to an incoming interface.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuration Examples for Standard IP Access List Logging

Example: Limiting Debug Output

The following sample configuration uses an access list to limit the **debug** command output. Limiting the **debug** output restricts the volume of data to what you are interested in, saving you time and resources.

```
Device(config)# ip access-list acl1
Device(config-std-nacl)# remark Displays only advertisements for LDP peer in acl1
Device(config-std-nacl)# permit host 10.0.0.44

Device# debug mpls ldp advertisements peer-acl acl1

tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.17.0.33
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.16.0.31
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.22.0.33
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.0.1
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.0.3
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.1.33
```

Additional References for Standard IP Access List Logging

Related Documents

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

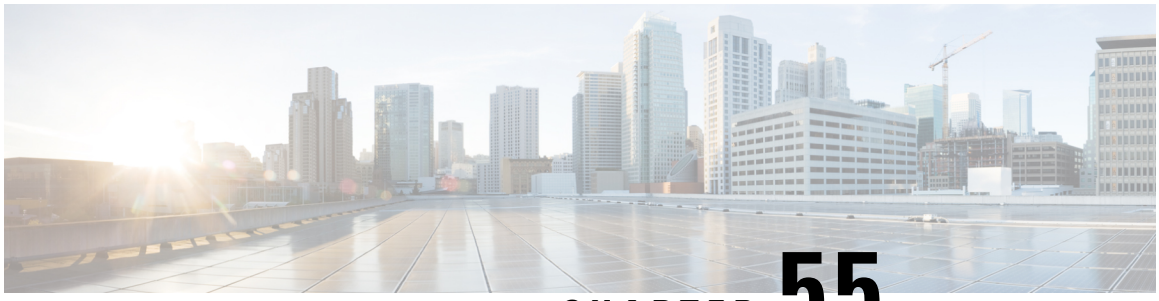
Feature Information for Standard IP Access List Logging

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 138: Feature Information for Standard IP Access List Logging

Feature Name	Releases	Feature Information
IP Access Lists	Cisco IOS-XE Release 3.15.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 55

IP Access List Entry Sequence Numbering

The IP Access List Entry Sequence Numbering feature allows you to apply sequence numbers to **permit** or **deny** statements as well as reorder, add, or remove such statements from a named IP access list. The IP Access List Entry Sequence Numbering feature makes revising IP access lists much easier. Prior to this feature, you could add access list entries to the end of an access list only; therefore, needing to add statements anywhere except at the end of a named IP access list required reconfiguring the entire access list.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 839
- Restrictions for IP Access List Entry Sequence Numbering, on page 840
- Information About IP Access List Entry Sequence Numbering, on page 840
- How to Use Sequence Numbers in an IP Access List, on page 844
- Configuration Examples for IP Access List Entry Sequence Numbering, on page 847
- Additional References, on page 849
- Feature Information for IP Access List Entry Sequence Numbering , on page 849

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 139: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for IP Access List Entry Sequence Numbering

- This feature does not support dynamic, reflexive, or firewall access lists.
- This feature does not support old-style numbered access lists, which existed before named access lists. Keep in mind that you can name an access list with a number, so numbers are allowed when they are entered in the standard or extended named access list (NACL) configuration mode.

Information About IP Access List Entry Sequence Numbering

Purpose of IP Access Lists

Access lists perform packet filtering to control which packets move through the network and where. Such control can help limit network traffic and restrict the access of users and devices to the network. Access lists have many uses, and therefore many commands accept a reference to an access list in their command syntax. Access lists can be used to do the following:

- Filter incoming packets on an interface.
- Filter outgoing packets on an interface.

- Restrict the contents of routing updates.
- Limit debug output based on an address or protocol.
- Control virtual terminal line access.
- Identify or classify traffic for advanced features, such as congestion avoidance, congestion management, and priority and custom queuing.
- Trigger dial-on-demand routing (DDR) calls.

How an IP Access List Works

An access list is a sequential list consisting of a permit statement and a deny statement that apply to IP addresses and possibly upper-layer IP protocols. The access list has a name by which it is referenced. Many software commands accept an access list as part of their syntax.

An access list can be configured and named, but it is not in effect until the access list is referenced by a command that accepts an access list. Multiple commands can reference the same access list. An access list can control traffic arriving at the device or leaving the device, but not traffic originating at the device.

IP Access List Process and Rules

- The software tests the source or destination address or the protocol of each packet being filtered against the conditions in the access list, one condition (**permit** or **deny** statement) at a time.
- If a packet does not match an access list statement, the packet is then tested against the next statement in the list.
- If a packet and an access list statement match, the rest of the statements in the list are skipped and the packet is permitted or denied as specified in the matched statement. The first entry that the packet matches determines whether the software permits or denies the packet. That is, after the first match, no subsequent entries are considered.
- If the access list denies the address or protocol, the software discards the packet and returns an Internet Control Message Protocol (ICMP) Host Unreachable message.
- If no conditions match, the packet is dropped. This is because each access list ends with an unwritten or implicit **deny** statement. That is, if the packet has not been permitted by the time it was tested against each statement, it is denied.
- Because the software stops testing conditions after the first match, the order of the conditions is critical. The same **permit** or **deny** statements specified in a different order could result in a packet being passed under one circumstance and denied in another circumstance.
- If an access list is referenced by name in a command, but the access list does not exist, all packets pass.
- Only one access list per interface, per protocol, per direction is allowed.
- Inbound access lists process packets arriving at the device. Incoming packets are processed before being routed to an outbound interface. An inbound access list is efficient because it saves the overhead of routing lookups if the packet is to be discarded because it is denied by the filtering tests. If the packet is permitted by the tests, it is then processed for routing. For inbound lists, **permit** means continue to process the packet after receiving it on an inbound interface; **deny** means discard the packet.

- Outbound access lists process packets before they leave the device. Incoming packets are routed to the outbound interface and then processed through the outbound access list. For outbound lists, **permit** means send it to the output buffer; **deny** means discard the packet.

Helpful Hints for Creating IP Access Lists

The following tips will help you avoid unintended consequences and help you create more efficient, useful access lists.

- Create the access list before applying it to an interface (or elsewhere), because if you apply a nonexistent access list to an interface and then proceed to configure the access list, the first statement is put into effect, and the implicit **deny** statement that follows could cause you immediate access problems.
- Another reason to configure an access list before applying it is because an interface with an empty access list applied to it permits all traffic.
- All access lists need at least one **permit** statement; otherwise, all packets are denied and no traffic passes.
- Because the software stops testing conditions after it encounters the first match (to either a **permit** or **deny** statement), you will reduce processing time and resources if you put the statements that packets are most likely to match at the beginning of the access list. Place more frequently occurring conditions before less frequent conditions.
- Organize your access list so that more specific references in a network or subnet appear before more general ones.
- Use the statement **permit any any** if you want to allow all other packets not already denied. Using the statement **permit any any** in effect avoids denying all other packets with the implicit deny statement at the end of an access list. Do not make your first access list entry **permit any any** because all traffic will get through; no packets will reach the subsequent testing. In fact, once you specify **permit any any** , all traffic not already denied will get through.
- Although all access lists end with an implicit **deny** statement, we recommend use of an explicit **deny** statement (for example, **deny ip any any**). On most platforms, you can display the count of packets denied by issuing the **show access-list** command, thus finding out more information about who your access list is disallowing. Only packets denied by explicit **deny** statements are counted, which is why the explicit **deny** statement will yield more complete data for you.
- While you are creating an access list or after it is created, you might want to delete an entry.
  - You cannot delete an entry from a numbered access list; trying to do so will delete the entire access list. If you need to delete an entry, you need to delete the entire access list and start over.
  - You can delete an entry from a named access list. Use the **no permit** or **no deny** command to delete the appropriate entry.
- In order to make the purpose of individual statements more scannable and easily understood at a glance, you can write a helpful remark before or after any statement by using the **remark** command.
- If you want to deny access to a particular host or network and find out if someone from that network or host is attempting to gain access, include the **log** keyword with the corresponding **deny** statement so that the packets denied from that source are logged for you.

- This hint applies to the placement of your access list. When trying to save resources, remember that an inbound access list applies the filter conditions before the routing table lookup. An outbound access list applies the filter conditions after the routing table lookup.

Source and Destination Addresses

Source and destination address fields in an IP packet are two typical fields on which to base an access list. Specify source addresses to control the packets being sent from certain networking devices or hosts. Specify destination addresses to control the packets being sent to certain networking devices or hosts.

Wildcard Mask and Implicit Wildcard Mask

When comparing the address bits in an access list entry to a packet being submitted to the access list, address filtering uses wildcard masking to determine whether to check or ignore the corresponding IP address bits. By carefully setting wildcard masks, an administrator can select one or more IP addresses for permit or deny tests.

Wildcard masking for IP address bits uses the number 1 and the number 0 to specify how the software treats the corresponding IP address bits. A wildcard mask is sometimes referred to as an inverted mask because a 1 and 0 mean the opposite of what they mean in a subnet (network) mask.

- A wildcard mask bit 0 means check the corresponding bit value.
- A wildcard mask bit 1 means ignore that corresponding bit value.

If you do not supply a wildcard mask with a source or destination address in an access list statement, the software assumes a default wildcard mask of 0.0.0.0.

Unlike subnet masks, which require contiguous bits indicating network and subnet to be ones, wildcard masks allow noncontiguous bits in the mask.

Transport Layer Information

You can filter packets based on transport layer information, such as whether the packet is a TCP, UDP, Internet Control Message Protocol (ICMP) or Internet Group Management Protocol (IGMP) packet.

Benefits IP Access List Entry Sequence Numbering

The ability to apply sequence numbers to IP access list entries simplifies access list changes. Prior to the IP Access List Entry Sequence Numbering feature, there was no way to specify the position of an entry within an access list. If you wanted to insert an entry (statement) in the middle of an existing list, all of the entries *after* the desired position had to be removed. Then, once you added the new entry, you needed to reenter all of the entries you removed earlier. This method was cumbersome and error prone.

The IP Access List Entry Sequence Numbering feature allows you to add sequence numbers to access list entries and resequence them. When you add a new entry, you can choose the sequence number so that the entry is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced (reordered) to create room to insert the new entry.

Sequence Numbering Behavior

- For backward compatibility with previous releases, if entries with no sequence numbers are applied, the first entry is assigned a sequence number of 10, and successive entries are incremented by 10. The maximum sequence number is 2147483647. If the generated sequence number exceeds this maximum number, the following message is displayed:

```
Exceeded maximum sequence number.
```

- If you enter an entry without a sequence number, it is assigned a sequence number that is 10 greater than the last sequence number in that access list and is placed at the end of the list.
- If you enter an entry that matches an already existing entry (except for the sequence number), then no changes are made.
- If you enter a sequence number that is already present, the following error message is generated:

```
Duplicate sequence number.
```

- If a new access list is entered from global configuration mode, then sequence numbers for that access list are generated automatically.
- Distributed support is provided so that the sequence numbers of entries in the Route Processor (RP) and line card (LC) are always synchronized.
- Sequence numbers are not nvgened. That is, the sequence numbers themselves are not saved. In the event that the system is reloaded, the configured sequence numbers revert to the default sequence starting number and increment from that number. The function is provided for backward compatibility with software releases that do not support sequence numbering.
- The IP Access List Entry Sequence Numbering feature works with named standard and extended IP access lists. Because the name of an access list can be designated as a number, numbers are acceptable.

How to Use Sequence Numbers in an IP Access List

Sequencing Access-List Entries and Revising the Access List

This task shows how to assign sequence numbers to entries in a named IP access list and how to add or delete an entry to or from an access list. When completing this task, keep the following points in mind:

- Resequencing the access list entries is optional. The resequencing step in this task is shown as required because that is one purpose of this feature and this task demonstrates that functionality.
- In the following procedure, the **permit** command is shown in Step 5 and the **deny** command is shown in Step 6. However, that order can be reversed. Use the order that suits the need of your configuration.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list resequence** *access-list-name starting-sequence-number increment* **Example:** `Device(config)# ip access-list resequence kmd1 100 15`	Resequences the specified IP access list using the starting sequence number and the increment of sequence numbers.
Step 4	**ip access-list** { **standard** | **extended** } *access-list-name* **Example:** `Device(config)# ip access-list standard kmd1`	Specifies the IP access list by name and enters named access list configuration mode. • If you specify **standard** , make sure you subsequently specify **permit** and/or **deny** statements using the standard access list syntax. • If you specify **extended** , make sure you subsequently specify **permit** and/or **deny** statements using the extended access list syntax.
Step 5	Do one of the following: • *sequence-number* **permit** *source source-wildcard* • *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-std-nacl)# 105 permit 10.5.5.5 0.0.0 255`	Specifies a permit statement in named IP access list mode. • This access list happens to use a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • As the prompt indicates, this access list was a standard access list. If you had specified **extended** in Step 4, the prompt for this step would be Device(config-ext-nacl) and you would use the extended **permit** command syntax.

	Command or Action	Purpose
Step 6	Do one of the following: • *sequence-number* **deny** *source source-wildcard* • *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-std-nacl)# 105 deny 10.6.6.7 0.0.0 255`	(Optional) Specifies a deny statement in named IP access list mode. • This access list uses a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • As the prompt indicates, this access list was a standard access list. If you had specified **extended** in Step 4, the prompt for this step would be Device(config-ext-nacl) and you would use the extended **deny** command syntax.
Step 7	Do one of the following: • *sequence-number* **permit** *source source-wildcard* • *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# 150 permit tcp any any log`	Specifies a permit statement in named IP access list mode. • This access list happens to use a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • See the permit (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP). • Use the **no** *sequence-number* command to delete an entry.
Step 8	Do one of the following: • *sequence-number* **deny** *source source-wildcard* • *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# 150 deny tcp any any log`	(Optional) Specifies a deny statement in named IP access list mode. • This access list happens to use a **permit** statement first, but a **deny** statement could appear first, depending on the order of statements you need. • See the deny (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP). • Use the **no** *sequence-number* command to delete an entry.
Step 9	Repeat Step 5 and/or Step 6 to add sequence number statements, as applicable.	Allows you to revise the access list.
Step 10	**end** **Example:**	(Optional) Exits the configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-std-nacl)# end`	
Step 11	**show ip access-lists** *access-list-name* **Example:** `Device# show ip access-lists kmd1`	(Optional) Displays the contents of the IP access list.

Examples

Review the output of the **show ip access-lists** command to see that the access list includes the new entries:

```
Device# show ip access-lists kmd1

Standard IP access list kmd1
100 permit 10.4.4.0, wildcard bits 0.0.0.255
105 permit 10.5.5.0, wildcard bits 0.0.0.255
115 permit 10.0.0.0, wildcard bits 0.0.0.255
130 permit 10.5.5.0, wildcard bits 0.0.0.255
145 permit 10.0.0.0, wildcard bits 0.0.0.255
```

Configuration Examples for IP Access List Entry Sequence Numbering

Example: Resequencing Entries in an Access List

The following example shows access list resequencing. The starting value is 1, and increment value is 2. The subsequent entries are ordered based on the increment values specified, and the range is from 1 to 2147483647.

When an entry with no sequence number is entered, by default the entry has a sequence number of 10 more than the last entry in the access list.

```
Device# show access-list 150

Extended IP access list 150
    10 permit ip host 10.3.3.3 host 172.16.5.34
    20 permit icmp any any
    30 permit tcp any host 10.3.3.3
    40 permit ip host 10.4.4.4 any
    50 Dynamic test permit ip any any
    60 permit ip host 172.16.2.2 host 10.3.3.12
    70 permit ip host 10.3.3.3 any log
    80 permit tcp host 10.3.3.3 host 10.1.2.2
    90 permit ip host 10.3.3.3 any
    100 permit ip any any

Device(config)# ip access-list extended 150
Device(config)# ip access-list resequence 150 1 2
```

```
Device(config)# exit

Device# show access-list 150

Extended IP access list 150
    1 permit ip host 10.3.3.3 host 172.16.5.34
    3 permit icmp any any
    10 permit tcp any any eq 22 log
    5 permit tcp any host 10.3.3.3
    7 permit ip host 10.4.4.4 any
    9 Dynamic test permit ip any any
    11 permit ip host 172.16.2.2 host 10.3.3.12
    13 permit ip host 10.3.3.3 any log
    15 permit tcp host 10.3.3.3 host 10.1.2.2
    17 permit ip host 10.3.3.3 any
    19 permit ip any any
```

Example: Adding Entries with Sequence Numbers

In the following example, an new entry is added to a specified access list:

```
Device# show ip access-list

Standard IP access list tryon
2 permit 10.4.4.2, wildcard bits 0.0.255.255
5 permit 10.0.0.44, wildcard bits 0.0.0.255
10 permit 10.0.0.1, wildcard bits 0.0.0.255
20 permit 10.0.0.2, wildcard bits 0.0.0.255

Device(config)# ip access-list standard tryon
Device(config-std-nacl)# 15 permit 10.5.5.5 0.0.0.255
Device(config-std-nacl)# exit
Device(config)# exit
Device# show ip access-list

Standard IP access list tryon
2 permit 10.4.0.0, wildcard bits 0.0.255.255
5 permit 10.0.0.0, wildcard bits 0.0.0.255
10 permit 10.0.0.0, wildcard bits 0.0.0.255
15 permit 10.5.5.0, wildcard bits 0.0.0.255
20 permit 10.0.0.0, wildcard bits 0.0.0.255
```

Example: Entry Without Sequence Number

The following example shows how an entry with no specified sequence number is added to the end of an access list. When an entry is added without a sequence number, it is automatically given a sequence number that puts it at the end of the access list. Because the default increment is 10, the entry will have a sequence number 10 higher than the last entry in the existing access list.

```
Device(config)# ip access-list standard 1
Device(config-std-nacl)# permit 10.1.1.1 0.0.0.255
Device(config-std-nacl)# permit 10.2.2.2 0.0.0.255
Device(config-std-nacl)# permit 10.3.3.3 0.0.0.255
Device(config-std-nacl)## exit
Device# show access-list

Standard IP access list 1
10 permit 0.0.0.0, wildcard bits 0.0.0.255
```

```
20 permit 0.0.0.0, wildcard bits 0.0.0.255
30 permit 0.0.0.0, wildcard bits 0.0.0.255

Device(config)# ip access-list standard 1
Device(config-std-nacl)# permit 10.4.4.4 0.0.0.255
Device(config-std-nacl)# end
Device(config-std-nacl)## exit
Device# show access-list

Standard IP access list 1
10 permit 0.0.0.0, wildcard bits 0.0.0.255
20 permit 0.0.0.0, wildcard bits 0.0.0.255
30 permit 0.0.0.0, wildcard bits 0.0.0.255
40 permit 0.0.0.0, wildcard bits 0.0.0.255
```

Additional References

Related Documents

Related Topic	Document Title
IP access list commands	*Cisco IOS Security Command Reference*
Configuring IP access lists	*Creating an IP Access List and Applying It to an Interface*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

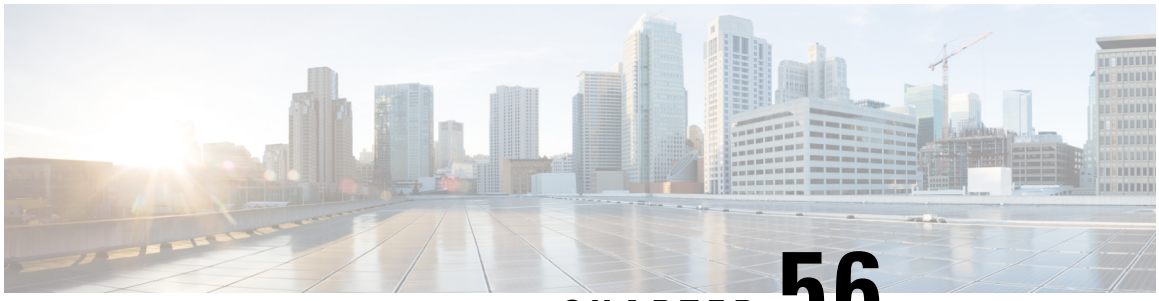
Feature Information for IP Access List Entry Sequence Numbering

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 140: Feature Information for IP Access List Entry Sequence Numbering

Feature Name	Releases	Feature Information
IP Access Lists	Cisco IOS-XE Release 3.15.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 56

ACL IP Options Selective Drop

The ACL IP Options Selective Drop feature allows Cisco routers to filter packets containing IP options or to mitigate the effects of IP options on a router or downstream routers by dropping these packets or ignoring the processing of the IP options.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 851
- Restrictions for ACL IP Options Selective Drop, on page 852
- Information About ACL IP Options Selective Drop, on page 852
- How to Configure ACL IP Options Selective Drop, on page 853
- Configuration Examples for ACL IP Options Selective Drop, on page 854
- Additional References for IP Access List Entry Sequence Numbering, on page 854
- Feature Information for ACL IP Options Selective Drop , on page 855

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 141: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for ACL IP Options Selective Drop

Resource Reservation Protocol (RSVP) (Multiprotocol Label Switching traffic engineering [MPLS TE]), Internet Group Management Protocol Version 2 (IGMPv2), and other protocols that use IP options packets may not function in drop or ignore modes.

Information About ACL IP Options Selective Drop

Using ACL IP Options Selective Drop

The ACL IP Options Selective Drop feature allows a router to filter IP options packets, thereby mitigating the effects of these packets on a router and downstream routers, and perform the following actions:

- Drop all IP options packets that it receives and prevent options from going deeper into the network.
- Ignore IP options packets destined for the router and treat them as if they had no IP options.

For many users, dropping the packets is the best solution. However, in environments in which some IP options may be legitimate, reducing the load that the packets present on the routers is sufficient. Therefore, users may prefer to skip options processing on the router and forward the packet as though it were pure IP.

Benefits of Using ACL IP Options Selective Drop

- Drop mode filters packets from the network and relieves downstream routers and hosts of the load from options packets.
- Drop mode minimizes loads to the Route Processor (RP) for options that require RP processing on distributed systems. Previously, the packets were always routed to or processed by the RP CPU. Now, the ignore and drop forms prevent the packets from impacting the RP performance.

How to Configure ACL IP Options Selective Drop

Configuring ACL IP Options Selective Drop

This section describes how to configure the ACL IP Options Selective Drop feature.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip options** { **drop** | **ignore** } **Example:** `Router(config)# ip options drop`	Drops or ignores IP options packets that are sent to the router.
Step 4	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show ip traffic** **Example:** `Router# show ip traffic`	(Optional) Displays statistics about IP traffic.

Configuration Examples for ACL IP Options Selective Drop

Example Configuring ACL IP Options Selective Drop

The following example shows how to configure the router (and downstream routers) to drop all options packets that enter the network:

```
Router(config)# ip options drop
% Warning:RSVP and other protocols that use IP Options packets may not function in drop or
 ignore modes.
end
```

Example Verifying ACL IP Options Selective Drop

The following sample output is displayed after using the **ip options drop** command:

```
Router# show ip traffic
IP statistics:
  Rcvd:  428 total, 323 local destination
         0 format errors, 0 checksum errors, 0 bad hop count
         0 unknown protocol, 0 not a gateway
         0 security failures, 0 bad options, 0 with options
  Opts:  0 end, 0 nop, 0 basic security, 0 loose source route
         0 timestamp, 0 extended security, 0 record route
         0 stream ID, 0 strict source route, 0 alert, 0 cipso, 0 ump
         0 other, 30 ignored
  Frags: 0 reassembled, 0 timeouts, 0 couldn't reassemble
         0 fragmented, 0 fragments, 0 couldn't fragment
  Bcast: 0 received, 0 sent
  Mcast: 323 received, 809 sent
  Sent:  809 generated, 591 forwarded
  Drop:  0 encapsulation failed, 0 unresolved, 0 no adjacency
         0 no route, 0 unicast RPF, 0 forced drop, 0 unsupported-addr
         0 options denied, 0 source IP address zero
```

Additional References for IP Access List Entry Sequence Numbering

The following sections provide references related to IP access lists.

Related Documents

Related Topic	Document Title
Configuring IP access lists	"Creating an IP Access List and Applying It to an Interface"

Related Topic	Document Title
IP access list commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

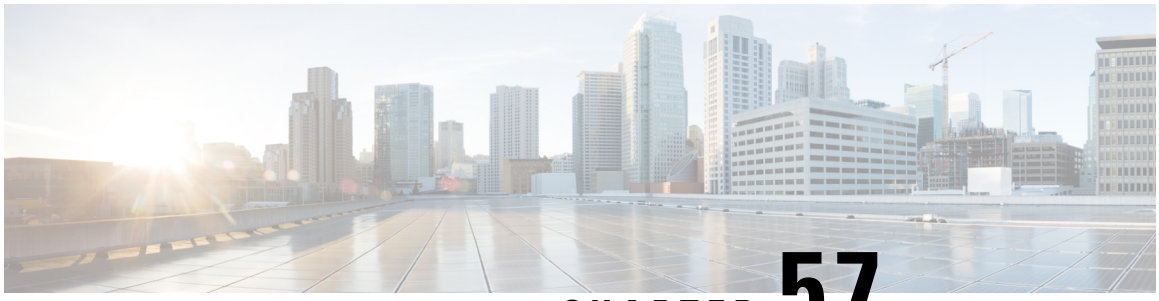
Feature Information for ACL IP Options Selective Drop

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 142: Feature Information for ACL IP Options Selective Drop

Feature Name	Releases	Feature Information
IP Access Lists	Cisco IOS-XE Release 3.15.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 57

ACL Syslog Correlation

The Access Control List (ACL) Syslog Correlation feature appends a tag (either a user-defined cookie or a device-generated MD5 hash value) to access control entry (ACE) syslog entries. This tag uniquely identifies the ACE , within the ACL, that generated the syslog entry.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 857
- Prerequisites for ACL Syslog Correlation, on page 858
- Information About ACL Syslog Correlation, on page 858
- How to Configure ACL Syslog Correlation, on page 859
- Configuration Examples for ACL Syslog Correlation, on page 865
- Additional References for IPv6 IOS Firewall, on page 866
- Feature Information for ACL Syslog Correlation , on page 867

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 143: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for ACL Syslog Correlation

Before you configure the ACL Syslog Correlation feature, you must understand the concepts in the "IP Access List Overview" module.

The ACL Syslog Correlation feature appends a user-defined cookie or a device-generated hash value to ACE messages in the syslog. These values are only appended to ACE messages when the log option is enabled for the ACE.

Information About ACL Syslog Correlation

ACL Syslog Correlation Tags

The ACL Syslog Correlation feature appends a tag (either a user-defined cookie or a device-generated MD5 hash value) to access control entry (ACE) syslog entries. This tag uniquely identifies an ACE that generated the syslog entry.

Network management software can use the tag to identify which ACE generated a specific syslog event. For example, network administrators can select an ACE rule in the network management application and can then view the corresponding syslog events for that ACE rule.

To append a tag to the syslog message, the ACE that generates the syslog event must have the log option enabled. The system appends only one type of tag (either a user-defined cookie or a device-generated MD5 hash value) to each message.

To specify a user-defined cookie tag, the user must enter the cookie value when configuring the ACE log option. The cookie must be in alpha-numeric form, it cannot be greater than 64 characters, and it cannot start with hex-decimal notation (such as 0x).

To specify a device-generated MD5 hash value tag, the hash-generation mechanism must be enabled on the device and the user must not enter a cookie value while configuring the ACE log option.

ACE Syslog Messages

When a packet is matched against an access control entry (ACE) in an ACL, the system checks whether the log option is enabled for that event. If the log option is enabled and the ACL Syslog Correlation feature is configured on the device, the system attaches the tag to the syslog message. The tag is displayed at the end of the syslog message, in addition to the standard information.

The following is a sample syslog message showing a user-defined cookie tag:

```
Jun 5 12:55:44.359: %SEC-6-IPACCESSLOGP: list logacl permitted tcp 192.168.16.1(38402) ->
192.168.16.2(23), 1 packet [User_permiited_ACE]
```

The following is a sample syslog message showing a hash value tag:

```
Jun 5 12:55:44.359: %SEC-6-IPACCESSLOGP: list logacl permitted tcp 192.168.16.1(38402) ->
192.168.16.2(23), 1 packet [0x723E6E12]
```

How to Configure ACL Syslog Correlation

Enabling Hash Value Generation on a Device

Perform this task to configure the device to generate an MD5 hash value for each log-enabled access control entry (ACE) in the system that is not configured with a user-defined cookie.

When the hash value generation setting is enabled, the system checks all existing ACEs and generates a hash value for each ACE that requires one. When the hash value generation setting is disabled, all previously generated hash values are removed from the system.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ip access-list logging hash-generation** **Example:** `Device(config)# ip access-list logging hash-generation`	Enables hash value generation on the device. • If an ACE exists that is log enabled, and requires a hash value, the device automatically generates the value and displays the value on the console.
Step 4	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Step 5	Do one of the following:
	• **show ip access-list** *access-list-number*
	• **show ip access-list** *access-list-name*
	Example:
	`Device# show ip access-list 101`
	Example:
	`Device# show ip access-list acl`
		(Optional) Displays the contents of the numbered or named IP access list.
		• Review the output to confirm that the access list for a log-enabled ACE includes the generated hash value.

Disabling Hash Value Generation on a Device

Perform this task to disable hash value generation on the device. When the hash value generation setting is disabled, all previously generated hash values are removed from the system.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no ip access-list logging hash-generation** **Example:**	Disables hash value generation on the device. • The system removes any previously created hash values from the system.

	Command or Action	Purpose
	`Device(config)# no ip access-list logging hash-generation`	
Step 4	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Step 5	Do one of the following:
	• **show ip access-list** *access-list-number*
	• **show ip access-list** *access-list-name*
	Example:
	`Device# show ip access-list 101`
	Example:
	`Device# show ip access-list acl`
		(Optional) Displays the contents of the IP access list.
		• Review the output to confirm that the access list for a log-enabled ACE does not have a generated hash value.

Configuring ACL Syslog Correlation Using a User-Defined Cookie

Perform this task to configure the ACL Syslog Correlation feature on a device for a specific access list, using a user-defined cookie as the syslog message tag.

The example in this section shows how to configure the ACL Syslog Correlation feature using a user-defined cookie for a numbered access list. However, you can configure the ACL Syslog Correlation feature using a user-defined cookie for both numbered and named access lists, and for both standard and extended access lists.

Note The following restrictions apply when choosing the user-defined cookie value:

- The maximum number of characters is 64.
- The cookie cannot start with hexadecimal notation (such as 0x).
- The cookie cannot be the same as, or a subset of, the following keywords: **reflect** , **fragment** , **time-range** . For example, reflect and ref are not valid values. However, the cookie can start with the keywords. For example, reflectedACE and fragment_33 are valid values
- The cookie must contains only alphanumeric characters.

>

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* **permit** *protocol source destination* **log** *word* **Example:** `Device(config)# access-list 101 permit tcp host 10.1.1.1 host 10.1.1.2 log UserDefinedValue`	Defines an extended IP access list and a user-defined cookie value. • Enter the cookie value as the *word* argument.
Step 4	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip access-list** *access-list-number* **Example:** `Device# show ip access-list 101`	(Optional) Displays the contents of the IP access list. • Review the output to confirm that the access list includes the user-defined cookie value.

Examples

The following is sample output from the **show ip access-list** command for an access list with a user-defined cookie value.

```
Device# show ip access-list
101
Extended IP access list 101
30 permit tcp host 10.1.1.1 host 10.1.1.2 log (tag = UserDefinedValue)
```

Configuring ACL Syslog Correlation Using a Hash Value

Perform this task to configure the ACL Syslog Correlation feature on a device for a specific access list, using a device-generated hash value as the syslog message tag.

The steps in this section shows how to configure the ACL Syslog Correlation feature using a device-generated hash value for a numbered access list. However, you can configure the ACL Syslog Correlation feature using a device-generated hash value for both numbered and named access lists, and for both standard and extended access lists.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list logging hash-generation** **Example:** `Device(config)# ip access-list logging hash-generation`	Enables hash value generation on the device. • If an ACE exists that is log enabled, and requires a hash value, the device automatically generates the value and displays the value on the console.
Step 4	access-list *access-list-number* **permit** *protocol source destination* **log** **Example:** `Device(config)# access-list 102 permit tcp host 10.1.1.1 host 10.1.1.2 log`	Defines an extended IP access list. • Enable the log option for the access list, but do not specify a cookie value. • The device automatically generates a hash value for the newly defined access list.
Step 5	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip access-list** *access-list-number* **Example:** `Device# show ip access-list 102`	(Optional) Displays the contents of the IP access list. • Review the output to confirm that the access list includes the router-generated hash value.

Examples

The following is sample output from the **show ip access-list** command for an access list with a device-generated hash value.

```
Device# show ip access-list
102
Extended IP access list 102
10 permit tcp host 10.1.1.1 host 10.1.1.2 log (hash = 0x7F9CF6B9)
```

Changing the ACL Syslog Correlation Tag Value

Perform this task to change the value of the user-defined cookie or replace a device-generated hash value with a user-defined cookie.

The steps in this section shows how to change the ACL Syslog Correlation tag value on a numbered access list. However, you can change the ACL Syslog Correlation tag value for both numbered and named access lists, and for both standard and extended access lists.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	show access-list **Example:** `Device(config)# show access-list`	(Optional) Displays the contents of the access list.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 4	access-list *access-list-number* **permit** *protocol source destination* **log** *word*
	Example:
	`Device(config)# access-list 101 permit tcp host 10.1.1.1 host 10.1.1.2 log NewUDV`
	Example:
	`OR`
	Example:
	Example:
	`Device(config)# access-list 101 permit tcp any any log replacehash`
		Modifies the cookie or changes the hash value to a cookie.
		• You must enter the entire access list configuration command, replacing the previous tag value with the new tag value.

Step 5	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show ip access-list** *access-list-number* **Example:** `Device# show ip access-list 101`	(Optional) Displays the contents of the IP access list. • Review the output to confirm the changes.

Troubleshooting Tips

Use the **debug ip access-list hash-generation** command to display access list debug information. The following is an example of the **debug** command output:

```
Device# debug ip access-list hash-generation
 Syslog hash code generation debugging is on
Device# show debug
IP ACL:
 Syslog hash code generation debugging is on
Device# no debug ip access-list hash-generation

 Syslog hash code generation debugging is off
Device# show debug
Device#
```

Configuration Examples for ACL Syslog Correlation

Example: Configuring ACL Syslog Correlation Using a User-Defined Cookie

The following example shows how to configure the ACL Syslog Correlation feature on a device using a user-defined cookie.

```
Device#
Device# debug ip access-list hash-generation
Syslog MD5 hash code generation debugging is on
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# access-list 33 permit 10.10.10.6 log cook_33_std
Device(config)# do show ip access 33
Standard IP access list 33
10 permit 10.10.10.6 log (tag = cook_33_std)
Device(config)# end
```

Example: Configuring ACL Syslog Correlation using a Hash Value

The following examples shows how to configure the ACL Syslog Correlation feature on a device using a device-generated hash value.

```
Device# debug ip access-list hash-generation
Syslog MD5 hash code generation debugging is on
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# access-list 33 permit 10.10.10.7 log
```

```
Device(config)#
*Nov 7 13:51:23.615: %IPACL-HASHGEN: Hash Input: 33 standard permit 10.10.10.7
Hash Output: 0xCE87F535
Device(config)#
do show ip access 33

Standard IP access list 33
    10 permit 10.10.10.6 log (tag = cook_33_std)
    20 permit 10.10.10.7 log (hash = 0xCE87F535)
```

Example: Changing the ACL Syslog Correlation Tag Value

The following example shows how to replace an existing access list user-defined cookie with a new cookie value, and how to replace a device-generated hash value with a user-defined cookie value.

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# do show ip access-list 101
Extended IP access list 101
    10 permit tcp host 10.1.1.1 host 10.1.1.2 log (tag = MyCookie)
    20 permit tcp any any log (hash = 0x75F078B9)
Device(config)# access-list 101 permit tcp host 10.1.1.1 host 10.1.1.2 log NewUDV
Device(config)# do show access-list
Extended IP access list 101
    10 permit tcp host 10.1.1.1 host 10.1.1.2 log (tag = NewUDV)
    20 permit tcp any any log (hash = 0x75F078B9)
Device(config)# access-list 101 permit tcp any any log replacehash
Device(config)# do show access-list
Extended IP access list 101
    10 permit tcp host 10.1.1.1 host 10.1.1.2 log (tag = NewUDV)
    20 permit tcp any any log (tag = replacehash)
```

Additional References for IPv6 IOS Firewall

Related Documents

Related Topic	Document Title
Security commands	• *Cisco IOS Security Command Reference: Commands A to C* • *Cisco IOS Security Command Reference: Commands D to L* • *Cisco IOS Security Command Reference: Commands M to R* • *Cisco IOS Security Command Reference: Commands S to Z*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

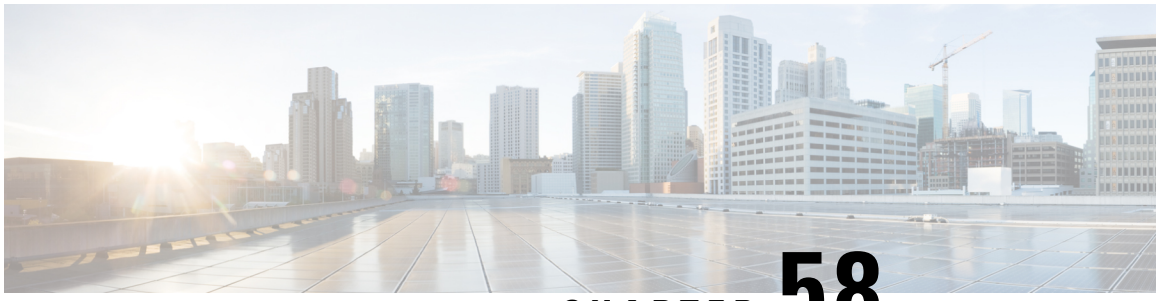
Feature Information for ACL Syslog Correlation

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 144: Feature Information for ACL Syslog Correlation

Feature Name	Releases	Feature Information
IP Access Lists	Cisco IOS-XE Release 3.15.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 58

IPv6 Access Control Lists

Access lists determine what traffic is blocked and what traffic is forwarded at device interfaces and allow filtering of traffic based on source and destination addresses, and inbound and outbound traffic to a specific interface. Standard IPv6 ACL functionality was extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control. Standard IPv6 ACL functionality was extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 869
- Information About IPv6 Access Control Lists, on page 870
- How to Configure IPv6 Access Control Lists, on page 871
- Configuration Examples for IPv6 Access Control Lists, on page 875
- Additional References, on page 876
- Feature Information for IPv6 Access Control Lists, on page 876

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 145: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About IPv6 Access Control Lists

Access Control Lists for IPv6 Traffic Filtering

The standard ACL functionality in IPv6 is similar to standard ACLs in IPv4. Access lists determine what traffic is blocked and what traffic is forwarded at device interfaces and allow filtering based on source and destination addresses, inbound and outbound to a specific interface. Each access list has an implicit deny statement at the end. IPv6 ACLs are defined and their deny and permit conditions are set using the **ipv6 access-list** command with the **deny** and **permit** keywords in global configuration mode.

IPv6 extended ACLs augments standard IPv6 ACL functionality to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control (functionality similar to extended ACLs in IPv4).

IPv6 Packet Inspection

The following header fields are used for IPv6 inspection: traffic class, flow label, payload length, next header, hop limit, and source or destination IP address. For further information on and descriptions of the IPv6 header fields, see RFC 2474.

Access Class Filtering in IPv6

Filtering incoming and outgoing connections to and from the device based on an IPv6 ACL is performed using the **ipv6 access-class** command in line configuration mode. The **ipv6 access-class** command is similar to the **access-class** command, except the IPv6 ACLs are defined by a name. If the IPv6 ACL is applied to inbound traffic, the source address in the ACL is matched against the incoming connection source address and the destination address in the ACL is matched against the local device address on the interface. If the IPv6 ACL is applied to outbound traffic, the source address in the ACL is matched against the local device address on the interface and the destination address in the ACL is matched against the outgoing connection source address. We recommend that identical restrictions are set on all the virtual terminal lines because a user can attempt to connect to any of them.

How to Configure IPv6 Access Control Lists

Configuring IPv6 Traffic Filtering

Creating and Configuring an IPv6 ACL for Traffic Filtering

Note IPv6 ACLs on the Cisco cBR router do not contain implicit permit rules. The IPv6 neighbor discovery process uses the IPv6 network-layer service; therefore, to enable IPv6 neighbor discovery, you must add IPv6 ACLs to allow IPv6 neighbor discovery packets to be sent and received on an interface. In IPv4, the Address Resolution Protocol (ARP), which is equivalent to the IPv6 neighbor discovery process, uses a separate data-link-layer protocol; therefore, by default IPv4 ACLs implicitly allow ARP packets to be sent and received on an interface.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list inbound`	Defines an IPv6 ACL, and enters IPv6 access list configuration mode. • The *access-list name* argument specifies the name of the IPv6 ACL. IPv6 ACL names cannot contain a space or quotation mark, or begin with a numeral.

	Command or Action	Purpose

Step 4	Do one of the following:
	• **permit protocol** { *source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address* } [*operator* [*port-number*]] { *destination-ipv6-prefix* / *prefix-length* | **any** | **host** *destination-ipv6-address* } [**operator** [*port-number*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]
	• **deny** *protocol* { *source-ipv6-prefix* / *prefix-length* | **any** | **host** *source-ipv6-address* } [*operator port-number*]] { *destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* } [*operator* [*port-number*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] [**undetermined-transport**
	Example:
	`Device(config-ipv6-acl)# permit tcp 2001:DB8:0300:0201::/32 eq telnet any`
	Example:
	`Device(config-ipv6-acl)# deny tcp host 2001:DB8:1::1 any log-input`
		Specifies permit or deny conditions for an IPv6 ACL.

Applying the IPv6 ACL to an Interface

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface TenGigabitEthernet4/1/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 4	**ipv6 traffic-filter** *access-list-name* {**in**| **out**} **Example:** `Device(config-if)# ipv6 traffic-filter outbound out`	Applies the specified IPv6 access list to the interface specified in the previous step.

Controlling Access to a vty

Creating an IPv6 ACL to Provide Access Class Filtering

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list cisco`	Defines an IPv6 ACL, and enters IPv6 access list configuration mode.
Step 4	Do one of the following: • **permit protocol** { *source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address* } [*operator* [	Specifies permit or deny conditions for an IPv6 ACL.

	Command or Action	Purpose

	port-number]] { *destination-ipv6-prefix* / *prefix-length* | **any** | **host** *destination-ipv6-address* } [*operator* [*port-number*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*
	• **deny** *protocol* { *source-ipv6-prefix* / *prefix-length* | **any** | **host** *source-ipv6-address* } [*operator* *port-number*]] { *destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* } [*operator* [*port-number*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] [**undetermined-transport**
	Example:
	`Device(config-ipv6-acl)# permit ipv6 host 2001:DB8:0:4::32 any`
	Example:
	`Device(config-ipv6-acl)# deny ipv6 host 2001:DB8:0:6::6 any`	

Applying an IPv6 ACL to the Virtual Terminal Line

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*] **Example:** `Device(config)# line vty 0 4`	Identifies a specific line for configuration and enters line configuration mode. • In this example, the **vty** keyword is used to specify the virtual terminal lines for remote console access.
Step 4	**ipv6 access-class** *ipv6-access-list-name* { **in** | **out** } **Example:** `Device(config-line)# ipv6 access-class cisco in`	Filters incoming and outgoing connections to and from the device based on an IPv6 ACL.

Configuration Examples for IPv6 Access Control Lists

Example: Verifying IPv6 ACL Configuration

In this example, the **show ipv6 access-list** command is used to verify that IPv6 ACLs are configured correctly:

```
Device> show ipv6 access-list

IPv6 access list inbound
    permit tcp any any eq bgp (8 matches) sequence 10
    permit tcp any any eq telnet  (15 matches) sequence 20
    permit udp any any  sequence 30

IPv6 access list Virtual-Access2.1#427819008151 (per-user)
    permit tcp host 2001:DB8:1::32 eq bgp host 2001:DB8:2::32 eq 11000 sequence 1
    permit tcp host 2001:DB8:1::32 eq telnet host 2001:DB8:2::32 eq 11001 sequence 2
```

Example: Creating and Applying an IPv6 ACL

The following example shows how to restrict HTTP access to certain hours during the day and log any activity outside of the permitted hours:

```
Device# configure terminal
Device(config)# time-range lunchtime
Device(config-time-range)# periodic weekdays 12:00 to 13:00
Device(config-time-range)# exit
Device(config)# ipv6 access-list INBOUND
Device(config-ipv6-acl)# permit tcp any any eq www time-range lunchtime
Device(config-ipv6-acl)# deny tcp any any eq www log-input
Device(config-ipv6-acl)# permit tcp 2001:DB8::/32 any
```

```
Device(config-ipv6-acl)# permit udp 2001:DB8::/32 any
Device(config-ipv6-acl)# end
```

Example: Controlling Access to a vty

In the following example, incoming connections to the virtual terminal lines 0 to 4 are filtered based on the IPv6 access list named acl1:

```
ipv6 access-list acl1
 permit ipv6 host 2001:DB8:0:4::2/32 any
!
line vty 0 4
 ipv6 access-class acl1 in
```

Additional References

Related Documents

Related Topic	Document Title
IP access list commands	*Cisco IOS Security Command Reference*
Configuring IP access lists	*Creating an IP Access List and Applying It to an Interface*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

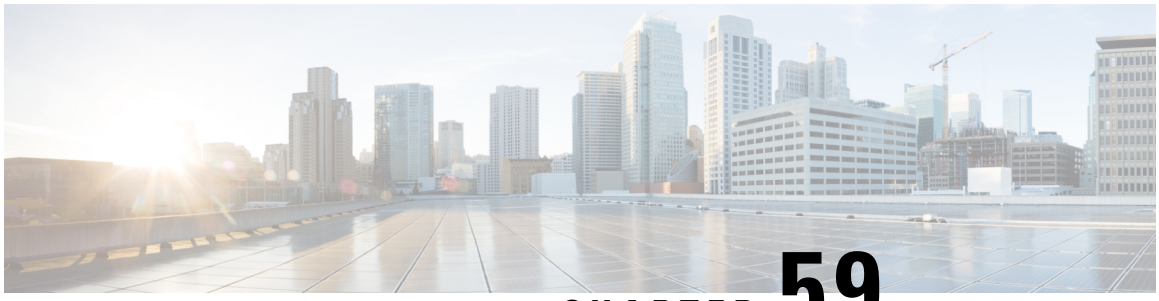
Feature Information for IPv6 Access Control Lists

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 146: Feature Information for IPv6 Access Control Lists

Feature Name	Releases	Feature Information
IPv6 Access Lists	Cisco IOS-XE Release 3.15	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 59

IPv6 Template ACL

When user profiles are configured using vendor-specific attribute (VSA) Cisco AV-pairs, similar per-user IPv6 ACLs may be replaced by a single template ACL. That is, one ACL represents many similar ACLs. By using IPv6 template ACLs, you can increase the total number of per-user ACLs while minimizing the memory and Ternary Content Addressable Memory (TCAM) resources needed to support the ACLs.

The IPv6 Template ACL feature can create templates using the following ACL fields:

- IPv6 source and destination addresses
- TCP and UDP, including all associated ports (0 through 65535)
- ICMP neighbor discovery advertisements and solicitations
- IPv6 DSCP with specified DSCP values

ACL names are dynamically generated by this feature; for example:

- 6Temp_#152875854573--Example of a dynamically generated template name for a template ACL parent
- Virtual-Access2.32135#152875854573--Example of a child ACL or an ACL that has not yet been made part of a template.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 880
- Information About IPv6 ACL—Template ACL, on page 880
- How to Enable IPv6 ACL—Template ACL, on page 881
- Configuration Examples for IPv6 ACL—Template ACL, on page 882
- Additional References, on page 882

• Feature Information for IPv6 Template ACL , on page 883

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 147: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About IPv6 ACL—Template ACL

IPv6 Template ACL

When user profiles are configured using vendor-specific attribute (VSA) Cisco AV-pairs, similar per-user IPv6 ACLs may be replaced by a single template ACL. That is, one ACL represents many similar ACLs. By using IPv6 template ACLs, you can increase the total number of per-user ACLs while minimizing the memory and Ternary Content Addressable Memory (TCAM) resources needed to support the ACLs.

The IPv6 Template ACL feature can create templates using the following ACL fields:

- IPv6 source and destination addresses
- TCP and UDP, including all associated ports (0 through 65535)
- ICMP neighbor discovery advertisements and solicitations
- IPv6 DSCP with specified DSCP values

ACL names are dynamically generated by this feature; for example:

- 6Temp_#152875854573--Example of a dynamically generated template name for a template ACL parent
- Virtual-Access2.32135#152875854573--Example of a child ACL or an ACL that has not yet been made part of a template.

How to Enable IPv6 ACL—Template ACL

Enabling IPv6 Template Processing

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list template** [*number-of-rules*] **Example:** `Router(config)# access-list template 50`	Enables template ACL processing. • The example in this task specifies that ACLs with 50 or fewer rules will be considered for template ACL status. • The *number-of-rules* argument default is 100.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and places the router in privileged EXEC mode.
Step 5	**show access-list template** { **summary** | *aclname* | **exceed** *number* | **tree** }	Displays information about ACL templates.

	Command or Action	Purpose
	Example: `Router# show access-list template summary`	

Configuration Examples for IPv6 ACL—Template ACL

Example: IPv6 Template ACL Processing

In this example, the contents of ACL1 and ACL2 are the same, but the names are different:

```
ipv6 access-list extended ACL1 (PeerIP: 2001:1::1/64)
permit igmp any                  2003:1::1/64
permit icmp 2002:5::B/64         any
permit udp  any                  host 2004:1::5
permit udp  any                  host 2002:2BC::a
permit icmp host 2001:BC::7      host 2003:3::7
ipv6 access-list extended ACL2 (PeerIP: 2007:2::7/64)
permit igmp any                  2003:1::1/64
permit icmp 2002:5::B/64         any
permit udp  any                  host 2004:1::5
permit udp  any                  host 2002:2BC::a
permit icmp host 2001:BC::7      host 2003:3::7
```

The template for these ACLs is as follows:

```
ipv6 access-list extended Template_1
permit igmp any                  2003:1::1/64
permit icmp 2002:5::B/64         any
permit udp  any                  host 2004:1::5
permit udp  any                  host 2002:2BC::a
permit icmp host 2001:BC::7      host 2003:3::7
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

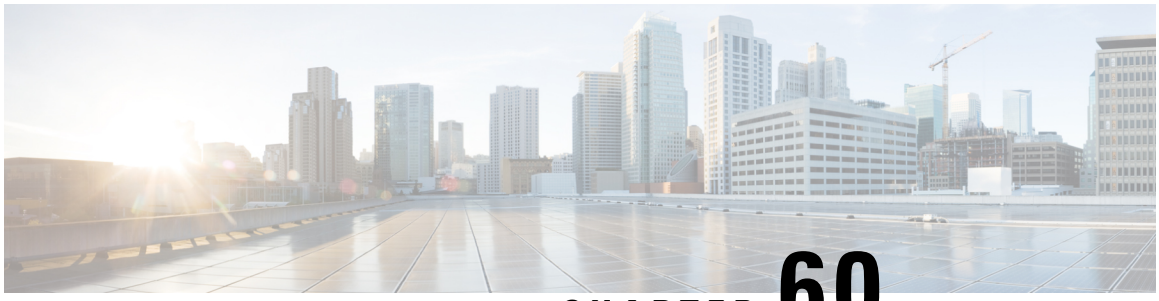
Feature Information for IPv6 Template ACL

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 148: Feature Information for IPv6 Template ACL

Feature Name	Releases	Feature Information
IPv6 Access Lists	Cisco IOS-XE Release 3.15.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 60

IPv6 ACL Extensions for Hop by Hop Filtering

The IPv6 ACL Extensions for Hop by Hop Filtering feature allows you to control IPv6 traffic that might contain hop-by-hop extension headers. You can configure an access control list (ACL) to deny all hop-by-hop traffic or to selectively permit traffic based on protocol.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 885
- Information About IPv6 ACL Extensions for Hop by Hop Filtering, on page 886
- How to Configure IPv6 ACL Extensions for Hop by Hop Filtering, on page 887
- Configuration Example for IPv6 ACL Extensions for Hop by Hop Filtering, on page 888
- Additional References, on page 889
- Feature Information for IPv6 ACL Extensions for Hop by Hop Filtering , on page 889

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 149: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About IPv6 ACL Extensions for Hop by Hop Filtering

ACLs and Traffic Forwarding

IPv6 access control lists (ACLs) determine what traffic is blocked and what traffic is forwarded at device interfaces. ACLs allow filtering based on source and destination addresses, inbound and outbound to a specific interface. Use the **ipv6 access-list** command to define an IPv6 ACL, and the **deny** and **permit** commands to configure its conditions.

The IPv6 ACL Extensions for Hop by Hop Filtering feature implements RFC 2460 to support traffic filtering in any upper-layer protocol type.

How to Configure IPv6 ACL Extensions for Hop by Hop Filtering

Configuring IPv6 ACL Extensions for Hop by Hop Filtering

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list hbh-acl`	Defines an IPv6 ACL and enters IPv6 access list configuration mode.
Step 4	**permit** *protocol* { *source-ipv6-prefix* / *prefix-length* | **any** | **host** *source-ipv6-address* | **auth** } [*operator* [*port-number*]] { *destination-ipv6-prefix* / *prefix-length* | **any** | **host** *destination-ipv6-address* | **auth** } [*operator* [*port-number*]] [**dest-option-type** [*header-number* | *header-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**hbh**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**reflect** *name* [**timeout** *value*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] **Example:** `Device(config-ipv6-acl)# permit icmp any any dest-option-type`	Sets permit conditions for the IPv6 ACL.
Step 5	**deny** *protocol* { *source-ipv6-prefix* / *prefix-length* | **any** | **host** *source-ipv6-address* | **auth** } [*operator* [*port-number*]] { *destination-ipv6-prefix* / *prefix-length* | **any** | **host** *destination-ipv6-address* | **auth** } [*operator* [*port-number*]] [**dest-option-type** [*header-number* | *header-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**hbh**] [	Sets deny conditions for the IPv6 ACL.

	Command or Action	Purpose
	log] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] [**undetermined-transport**] **Example:** `Device(config-ipv6-acl)# deny icmp any any dest-option-type`	
Step 6	**end** **Example:** `Device (config-ipv6-acl)# end`	Returns to privileged EXEC configuration mode.

Configuration Example for IPv6 ACL Extensions for Hop by Hop Filtering

Example: IPv6 ACL Extensions for Hop by Hop Filtering

```
Device(config)# ipv6 access-list hbh_acl
Device(config-ipv6-acl)# permit tcp any any hbh
Device(config-ipv6-acl)# permit tcp any any
Device(config-ipv6-acl)# permit udp any any
Device(config-ipv6-acl)# permit udp any any hbh
Device(config-ipv6-acl)# permit hbh any any
Device(config-ipv6-acl)# permit any any
Device(config-ipv6-acl)# hardware statistics
Device(config-ipv6-acl)# exit

! Assign an IP address and add the ACL on the interface.

Device(config)# interface TenGigabitEthernet4/1/0
Device(config-if)# ipv6 address 1001::1/64
Device(config-if)# ipv6 traffic-filter hbh_acl in
Device(config-if)# exit
Device(config)# exit
Device# clear counters
Clear "show interface" counters on all interfaces [confirm]
Device#

! Verify the configurations.

Device# show running-config interface TenGigabitEthernet4/1/0

Building configuration...

Current configuration : 114 bytes
!
interface TenGigabitEthernet4/1/0
no switchport
ipv6 address 1001::1/64
```

```
ipv6 traffic-filter hbh_acl
end
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 ACL Extensions for Hop by Hop Filtering

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release,

feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

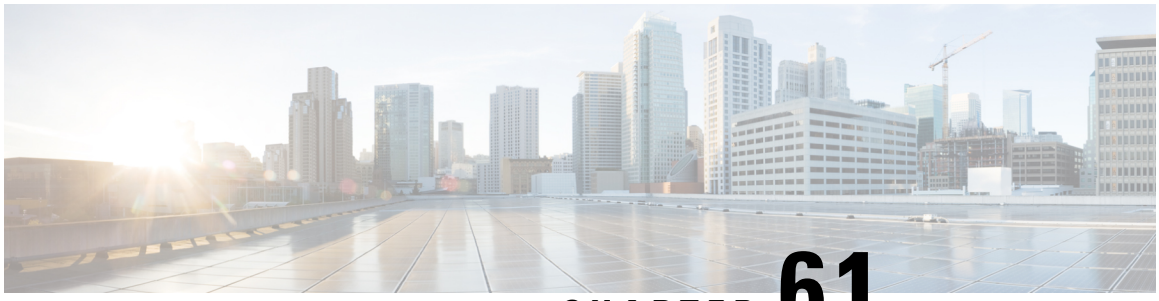
Table 150: Feature Information for IPv6 ACL Extensions for Hop by Hop Filtering

Feature Name	Releases	Feature Information
IPv6 Access Lists	Cisco IOS-XE Release 3.15.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.

PART VIII

Application—Voice and Video Configuration

- Unique Device Identifier Retrieval , on page 893
- Advanced-Mode DOCSIS Set-Top Gateway 1.2 for the Cisco CMTS Routers, on page 901
- Cisco Network Registrar for the Cisco CMTS Routers, on page 927



CHAPTER 61

Unique Device Identifier Retrieval

The Unique Device Identifier (UDI) Retrieval feature provides the ability to retrieve and display the UDI information from any Cisco product that has electronically stored such identity information.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 893
- Unique Device Identifier Overview, on page 894
- Benefits of the Unique Device Identifier Retrieval Feature, on page 895
- Retrieving the Unique Device Identifier, on page 895
- Troubleshooting Tips, on page 898
- Additional References, on page 898
- Feature Information for Unique Device Identifier Retrieval , on page 899

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 151: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Unique Device Identifier Overview

Each identifiable product is an entity, as defined by the Entity MIB (RFC-2737) and its supporting documents. Some entities, such as a chassis, will have sub-entities like slots. An Ethernet switch might be a member of a super-entity like a stack. Most Cisco entities that can be ordered leave the factory with an assigned UDI. The UDI information is printed on a label that is affixed to the physical hardware device, and it is also stored electronically on the device in order to facilitate remote retrieval.

A UDI consists of the following elements:

- Product identifier (PID)
- Version identifier (VID)
- Serial number (SN)

The PID is the name by which the product can be ordered; it has been historically called the “Product Name” or “Part Number.” This is the identifier that one would use to order an exact replacement part.

The VID is the version of the product. Whenever a product has been revised, the VID will be incremented. The VID is incremented according to a rigorous process derived from Telcordia GR-209-CORE, an industry guideline that governs product change notices.

The SN is the vendor-unique serialization of the product. Each manufactured product will carry a unique serial number assigned at the factory, which cannot be changed in the field. This is the means by which to identify an individual, specific instance of a product.

Benefits of the Unique Device Identifier Retrieval Feature

- Identifies individual Cisco products in your networks.
- Reduces operating expenses for asset management through simple, cross-platform, consistent identification of Cisco products.
- Identifies PIDs for replaceable products.
- Facilitates discovery of products subject to recall or revision.
- Automates Cisco product inventory (capital and asset management).
- Provides a mechanism to determine the entitlement level of a Cisco product for repair and replacement service.

Product Item Descriptor for Cable Products

For information on the Product Item Descriptor (PID), see the product hardware installation guide available on Cisco.com.

Retrieving the Unique Device Identifier

To use UDI retrieval, the Cisco product in use must be UDI-enabled. A UDI-enabled Cisco product supports five required Entity MIB objects. The five Entity MIB v2 (RFC-2737) objects are:

- entPhysicalName
- entPhysicalDescr
- entPhysicalModelName
- entPhysicalHardwareRev
- entPhysicalSerialNum

Although the **show inventory** command may be available, using that command on devices that are not UDI-enabled will likely produce no output.

Enter the **show inventory** command to retrieve and display information about all of the Cisco products installed in the networking device that are assigned a PID, VID, and SN. If a Cisco entity is not assigned a PID, that entity is not retrieved or displayed.

```
Router# show inventory

NAME: "Chassis", DESCR: "Cisco cBR-8 CCAP Chassis"
PID: CBR-8-CCAP-CHASS  , VID: V01, SN: FXS1739Q0PR

NAME: "clc 3", DESCR: "Cisco cBR CCAP Line Card"
PID: CBR-CCAP-LC-40G   , VID: V01, SN: TEST1234567

NAME: "Cable PHY Module", DESCR: "CLC Downstream PHY Module 3/0"
PID: CBR-D30-DS-MOD    , VID: V01, SN: CAT1725E1BZ

NAME: "Cable PHY Module", DESCR: "CLC Downstream PHY Module 3/1"
PID: CBR-D30-DS-MOD    , VID: V01, SN: CAT1725E1AT

NAME: "Cable PHY Module", DESCR: "CLC Upstream PHY Module 3/2"
PID: CBR-D30-US-MOD    , VID: V01, SN: CAT1717E0FF

NAME: "sup 1", DESCR: "Cisco cBR CCAP Supervisor Card"
PID: CBR-CCAP-SUP-60G  , VID: V01, SN: CAT1824E0MT
```

```
NAME: "harddisk 5/1", DESCR: "Hard Disk"
PID: UGB88RTB100HE3-BCU-DID, VID:    , SN: 11000066829

NAME: "sup-pic 5/1", DESCR: "Cisco cBR CCAP Supervisor Card PIC"
PID: CBR-SUP-8X10G-PIC , VID: V01, SN: CAT1720E0F4

NAME: "SFP+ module 5/1/0", DESCR: "iNSI xcvr"
PID: SFP+ 10GBASE-SR   , VID: A   , SN: FNS172720X6

NAME: "SFP+ module 5/1/1", DESCR: "iNSI xcvr"
PID: SFP+ 10GBASE-LR   , VID: A   , SN: UGT085P

NAME: "SFP+ module 5/1/2", DESCR: "iNSI xcvr"
PID: SFP+ 10GBASE-LR   , VID: A   , SN: UGT087Z

NAME: "SFP+ module 5/1/3", DESCR: "iNSI xcvr"
PID: SFP+ 10GBASE-SR   , VID: G4.1, SN: AVD1729A38T

NAME: "SFP+ module 5/1/7", DESCR: "iNSI xcvr"
PID: 10GE ZR           , VID: A   , SN: FNS11300AUH

NAME: "Power Supply Module 0", DESCR: "Cisco cBR CCAP AC Power Supply"
PID: PWR-3KW-AC-V2     , VID: V02, SN: DTM17370345

NAME: "Power Supply Module 2", DESCR: "Cisco cBR CCAP AC Power Supply"
PID: PWR-3KW-AC-V2     , VID: V02, SN: DTM173702KF
```

For diagnostic purposes, the **show inventory** command can be used with the **raw** keyword to display every RFC 2737 entity including those without a PID, UDI, or other physical identification.

Note The **raw** keyword option is primarily intended for troubleshooting problems with the **show inventory** command itself.

```
Router# show inventory raw

NAME: "Chassis", DESCR: "Cisco cBR-8 CCAP Chassis"
PID: CBR-8-CCAP-CHASS  , VID: V01, SN: FXS1739Q0PR

NAME: "slot 0/0", DESCR: "Chassis Slot"
PID:                   , VID:    , SN:

NAME: "slot 0/1", DESCR: "Chassis Slot"
PID:                   , VID:    , SN:

NAME: "slot 1/0", DESCR: "Chassis Slot"
PID:                   , VID:    , SN:

NAME: "slot 1/1", DESCR: "Chassis Slot"
PID:                   , VID:    , SN:

NAME: "slot 2/0", DESCR: "Chassis Slot"
PID:                   , VID:    , SN:

NAME: "slot 2/1", DESCR: "Chassis Slot"
PID:                   , VID:    , SN:

NAME: "slot 3/0", DESCR: "Chassis Slot"
```

```
PID:                    , VID:    , SN:

NAME: "clc 3", DESCR: "Cisco cBR CCAP Line Card"
PID: CBR-CCAP-LC-40G    , VID: V01, SN: TEST1234567

NAME: "12_CUR: Sens 3/0", DESCR: "12_CUR: Sens"
PID:                    , VID:    , SN:

NAME: "12_CUR: Vin 3/1", DESCR: "12_CUR: Vin"
PID:                    , VID:    , SN:

NAME: "12_CUR: ADin 3/2", DESCR: "12_CUR: ADin"
PID:                    , VID:    , SN:

NAME: "G0_CUR: Sens 3/3", DESCR: "G0_CUR: Sens"
PID:                    , VID:    , SN:

NAME: "G0_CUR: Vin 3/4", DESCR: "G0_CUR: Vin"
PID:                    , VID:    , SN:

NAME: "G0_CUR: ADin 3/5", DESCR: "G0_CUR: ADin"
PID:                    , VID:    , SN:

NAME: "G1_CUR: Sens 3/6", DESCR: "G1_CUR: Sens"
PID:                    , VID:    , SN:

NAME: "G1_CUR: Vin 3/7", DESCR: "G1_CUR: Vin"
PID:                    , VID:    , SN:

NAME: "G1_CUR: ADin 3/8", DESCR: "G1_CUR: ADin"
PID:                    , VID:    , SN:

NAME: "LB_CUR: Sens 3/9", DESCR: "LB_CUR: Sens"
PID:                    , VID:    , SN:

NAME: "LB_CUR: Vin 3/10", DESCR: "LB_CUR: Vin"
PID:                    , VID:    , SN:

NAME: "LB_CUR: ADin 3/11", DESCR: "LB_CUR: ADin"
PID:                    , VID:    , SN:

NAME: "Temp: CAPRICA 3/12", DESCR: "Temp: CAPRICA"
PID:                    , VID:    , SN:

NAME: "Temp: BASESTAR 3/13", DESCR: "Temp: BASESTAR"
PID:                    , VID:    , SN:

NAME: "Temp: RAIDER 3/14", DESCR: "Temp: RAIDER"
PID:                    , VID:    , SN:

NAME: "Temp: CPU 3/15", DESCR: "Temp: CPU"
PID:                    , VID:    , SN:

NAME: "Temp: INLET 3/16", DESCR: "Temp: INLET"
PID:                    , VID:    , SN:

NAME: "Temp: OUTLET 3/17", DESCR: "Temp: OUTLET"
PID:                    , VID:    , SN:

NAME: "Temp: DIGITAL 3/18", DESCR: "Temp: DIGITAL"
PID:                    , VID:    , SN:

NAME: "Temp: UPX 3/19", DESCR: "Temp: UPX"
```

```
PID:                    , VID:    , SN:
```

Troubleshooting Tips

If any of the Cisco products do not have an assigned PID, the output may display incorrect PIDs and the VID and SN elements may be missing, as in the following example.

```
NAME: "POS3/0/0", DESCR: "Skystone 4302 Sonet Framer"

PID: FastEthernet, VID: , SN:

NAME: "Serial1/0", DESCR: "M4T"

PID: M4T                , VID:    , SN:
```

In the sample output, the PID is exactly the same as the product description. The UDI is designed for use with new Cisco products that have a PID assigned. UDI information on older Cisco products is not always reliable.

Additional References

Related Documents

Related Topic	Document Title
Information about managing configuration files	Cisco IOS Configuration Fundamentals Configuration Guide
Commands for showing interface statistics	Cisco IOS Interface Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2737	*Entity MIB (Version 2)*

MIBs

MIB	MIBs Link
CISCO-ENTITY-ASSET-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

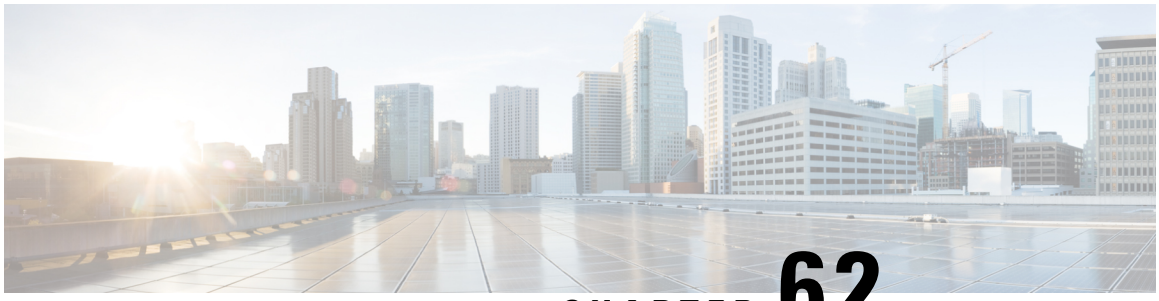
Feature Information for Unique Device Identifier Retrieval

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 152: Feature Information for Unique Device Identifier Retrieval

Feature Name	Releases	Feature Information
Unique Device Identifier Retrieval	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Routers.



CHAPTER 62

Advanced-Mode DOCSIS Set-Top Gateway 1.2 for the Cisco CMTS Routers

The Advanced-Mode DOCSIS Set-Top Gateway (A-DSG) Issue 1.2 introduces support for the latest DOCSIS Set-Top specification from CableLabs™, to include the following enhancements:

- *DOCSIS Set-top Gateway (DSG) Interface Specification*
- A-DSG 1.2 introduces support for the DOCS-DSG-IF MIB.

Cisco A-DSG 1.2 is certified by CableLabs™, and is a powerful tool in support of latest industry innovations. A-DSG 1.2 offers substantial support for enhanced DOCSIS implementation in the broadband cable environment. The set-top box (STB) dynamically learns the overall environment from the Cisco CMTS router, to include MAC address, traffic management rules, and classifiers.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 902
- Prerequisites for Advanced-Mode DSG Issue 1.2, on page 902
- Restrictions for Advanced-Mode DSG Issue 1.2, on page 902
- Information About Advanced-Mode DSG Issue 1.2, on page 903
- How to Configure Advanced-Mode DSG Issue 1.2, on page 906
- How to Monitor and Debug the Advanced-mode DOCSIS Set-Top Gateway Feature, on page 920
- Configuration Examples for Advanced-Mode DSG, on page 923
- Additional References, on page 926
- Feature Information for Advanced-Mode DSG 1.2 for the Cisco CMTS Routers, on page 926

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 153: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Advanced-Mode DSG Issue 1.2

No special equipment or software is needed to use the Advanced-Mode DSG Issue 1.2 feature.

Restrictions for Advanced-Mode DSG Issue 1.2

This section contains restrictions that are specific to A-DSG 1.2 on a Cisco CMTS router.

DSG Configuration File Transfer Operations

DSG 1.2 does not support the copying of a DSG configuration file from a TFTP server, file system, or bootflash to the running configuration.

Multicast Configuration Restrictions

IP multicasting must be configured for correct operation of A-DSG 1.2. Specifically, IP multicast routing must be set in global configuration. Also, IP PIM must be configured on all bundle interfaces of cable interfaces that are to carry multicast traffic.

See the Configuring the Default Multicast Quality of Service, on page 906 and the Configuring IP Multicast Operations, on page 912 for additional Multicast information and global configurations supporting DSG.

NAT for DSG Unicast-only Mapping

A-DSG 1.2 supports multicast IP addressing. However, it also supports unicast IP destination addresses. On the Cisco cBR-8 router, DSG 1.2 support is provided with the configuration of Network Address Translation (NAT) on the router, to include these settings:

- WAN interface(s) are configured with the **ip nat outside** command.
- Cable interface(s) are configured with the **ip nat inside** command.
- For each mapping, additional configuration includes the source static multicast IP address and the unicast IP address.

The unicast IP address is the unicast destination IP address of the DSG packets arriving at the Cisco CMTS router. The multicast IP address is the new destination IP address that is configured to map to one or a set of DSG tunnels.

PIM and SSM for Multicast

When using Source Specific Multicast (SSM) operation in conjunction with A-DSG 1.2, the following system-wide configuration command must be specified:

- **ip pim ssm**

Refer to the Configuring IP Multicast Operations, on page 912.

Subinterfaces

A-DSG 1.2 supports subinterfaces on the Cisco CMTS router.

Information About Advanced-Mode DSG Issue 1.2

A-DSG 1.2 offers these new or enhanced capabilities:

- A-DSG client and agent modes
- Advanced-mode MIBs supporting DSG 1.2, including the DOCS-DSG-IF-MIB
- Advanced-mode tunnels with increased security
- Cable interface bundling through virtual interface bundling

- Downstream Channel Descriptor
- IP multicast support
- Quality of Service (QoS)

DSG 1.2 Clients and Agents

A-DSG 1.2 supports the DSG client and agent functions outlined by the CableLabs™ *DOCSIS Set-top Gateway (DSG) Interface Specification* , CM-SP-DSG-I05-050812.

FQDN Support

You can specify either a fully-qualified domain name (FQDN) or IP address for A-DSG classifier multicast group and source addresses using the **cable dsg cfr** command in global configuration mode. We recommend that you use an FQDN to avoid modification of multicast group and source addresses when network changes are implemented.

This feature allows you to use a hostname (FQDN) in place of the source IP address using the **cable dsg cfr** command. For example, you have two A-DSG tunnel servers, in two locations, sending multicast traffic to the same multicast address. In this scenario, you can specify a hostname for the source IP address and let the DNS server determine which source is sending the multicast traffic.

If you configure an A-DSG classifier with a hostname, the Cisco CMTS router immediately verifies if the hostname can be resolved against an IP address using the local host cache. If not, the router does not enable the classifier until the hostname is resolved. If the hostname cannot be resolved locally, the router performs a DNS query to verify the DSG classifiers.

The FQDN format does not support static Internet Group Management Protocol (IGMP) join requests initiated on the Cisco CMTS router. The IGMP static group IP address created automatically under a bundle interface at the time of A-DSG configuration is not displayed in the **show running-config interface** command output. To display the A-DSG static groups configured under a bundle interface, use the **show cable dsg static-group bundle** command in privileged EXEC mode.

DSG Name Process and DNS Query

Every DNS record contains a time to live (TTL) value set by the server administrator, and this may vary from seconds to weeks. The DSG name process supersedes the TTL value criterion to update A-DSG classifiers on the Cisco CMTS router.

The DSG name process enables the Cisco CMTS router to query the DNS server for faster classifier updates. To enable the Cisco CMTS router to perform a DNS query for an A-DSG classifier verification, you must configure one or more DNS servers using the **ip name-server** command in global configuration mode. You can also specify the DNS query interval using the **cable dsg name-update-interval** command in global configuration mode.

During a Cisco IOS software reload or a route processor switchover, the router may fail to query the DNS server if the interfaces are down, and the router may not wait for the interval specified using the **cable dsg name-update-interval** command to perform a DNS query. In this case, for an unresolved hostname, the router automatically performs a DNS query based on a system-defined (15 seconds) interval to facilitate faster DSG classifier updates. You cannot change the system-defined interval.

A-DSG Forwarding on the Primary Channel

You can disable A-DSG forwarding per primary capable interface using the **cable downstream dsg disable** command in interface configuration mode. Primary capable interfaces include modular, integrated cable interfaces, and Cisco cBR-8 CCAP cable interfaces.

For example, assume the cable interface 7/1/1 has A-DSG enabled and has four modular channels attached to it. However, you want A-DSG forwarding enabled only on two of these four modular channels. You can exclude the channels of your choice using the cable downstream dsg disable command. For details on how to disable modular channels, see the Disabling A-DSG Forwarding on the Primary Channel, on page 919.

Note If A-DSG downstream forwarding is disabled on a primary capable interface, the router does not create multicast service flows on the primary capable interface and stops sending Downstream Channel Descriptor (DCD) messages.

DOCSIS 3.0 DSG MDF Support

Support for DOCSIS 3.0 DSG Multicast DSID Forwarding (MDF) is introduced using DSG DA-to-DSID Association Entry type, length, value (TLV 13) in the MAC domain descriptor (MDD) message to communicate the association between a downstream service identifier (DSID) and a group MAC address used for DSG tunnel traffic. This is automatically supported on the Cisco CMTS router.

DOCSIS 2.0 hybrid CMs and DOCSIS 3.0 CMs use Dynamic Bonding Change (DBC) to get DSID information from the Cisco CMTS router, whereas DOCSIS 2.0 DSG hybrid embedded CMs and DOCSIS 3.0 DSG embedded CMs get DSID information from the Cisco CMTS router through MDD messages.

To disable MDF capability on all DSG embedded cable modems, including DOCSIS 3.0 DSG and DOCSIS 2.0 DSG hybrid modems, use the cable multicast mdf-disable command with the dsg keyword in global configuration mode.

Source Specific Multicast Mapping

Source Specific Multicast (SSM) is a datagram delivery model that best supports one-to-many applications, also known as broadcast applications. SSM is a core networking technology for the Cisco implementation of IP multicast solutions targeted for audio and video broadcast application environments.

The following two Cisco IOS components together support the implementation of SSM:

- Protocol Independent Multicast source-specific mode (PIM-SSM)
- Internet Group Management Protocol Version 3 (IGMPv3)

SSM mapping can be configured on Cisco CMTS routers.

For details on how to configure SSM mapping on a Cisco CMTS router, see the Source Specific Multicast (SSM) Mapping feature guide.

How to Configure Advanced-Mode DSG Issue 1.2

Advanced-mode DSG Issue 1.2 entails support for DSG tunnel configuration, to include global, WAN-side, and interface-level settings in support of Multicast.

Configuring the Default Multicast Quality of Service

According to DOCSIS 3.0, you must configure the default multicast quality of service (MQoS) when using the MQoS. This also applies to the DSG, which uses the MQoS by associating a service class name with the tunnel.

If the default MQoS is not configured, the DSG tunnel service class configuration is rejected. Similarly, if no DSG tunnel uses the MQoS, you are prompted to remove the default MQoS.

The CMTS selects the primary downstream channel to forward the multicast traffic when the default MQoS is configured and there is no matching MQoS group configuration. Otherwise, the wideband interface is used to forward the multicast traffic.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	Example:
	`Router(config)#`
		Enters global configuration mode.

Step 3	**cable multicast group-qos default scn** *service-class-name* **aggregate** **Example:** `Router(config)# cable multicast group-qos default scn name1 aggregate`	Configures a service class name for the QoS profile.
Step 4	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

What to do next

Note If you configure or remove the default MQoS while the CMTS is sending multicast traffic, duplicate traffic is generated for approximately 3 minutes (or 3 times the query interval).

Configuring Global Tunnel Group Settings for Advanced-Mode DSG 1.2

This procedure configures global and interface-level commands on the Cisco CMTS router to enable DSG tunnel groups. A DSG tunnel group is used to bundle some DSG channels together and associate them to a MAC domain interface.

Global A-DSG 1.2 Tunnel Settings

This procedure sets and enables global configurations to support both A-DSG 1.2 clients and agents. Additional procedures provide additional settings for these clients and agents.

Before you begin

When DOCSIS Set-top Gateway (DSG) is configured to have quality of service (QoS) for tunnel, ensure that the default multicast QoS (MQoS) is also configured. For more information, see Configuring the Default Multicast Quality of Service, on page 906.

Note The DSG tunnel service class configuration is rejected, if default MQoS is not configured.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**cable dsg tg***group-id* [**channel***channel-id* |**priority***DSG-rule-priority*] [**enable**|**disable**] **Example:** `Router(config)# cable dsg tg 1 channel 1 priority 1 enable`	Command allows the association of a group of tunnels to one or more downstream interfaces on the Cisco CMTS.

	Command or Action	Purpose
Step 4	**cabledsg tg***group-id* [**channel** *channel-id* [**ucid** *ID1*]] **Example:** `Router(config)# cable dsg tg 1 channel 1 ucid 1`	Sets the upstream channel or channels to which the DSG 1.2 tunnel applies.
Step 5	**cable dsg tg** *group-id* [**channel** *channel-id* [**vendor-param** *vendor-group-id*]] **Example:** `Router(config)# cable dsg tg 1 channel 1 vendor-param 1`	Sets the vendor-specific parameters for upstream DSG 1.2 channels.
Step 6	**cable dsg vendor-param** *group-id* **vendor** *vendor-index* **oui** *oui* **value** *value-in-TLV* **Example:** `Router(config)# cable dsg vendor-param 1 vendor 1 oui ABCDEA value 0101AB`	Configures vendor-specific parameters for A-DSG 1.2. To remove this configuration from the Cisco CMTS, use the no form of this command.
Step 7	**cable dsg chan-list** *list-index* **index** *entry-index* **freq** *freq* **Example:** `Router(config)# cable dsg chan-list 1 index 1 freq 47000000`	Configures the A-DSG 1.2 downstream channel list. The channel list is a list of DSG channels (downstream frequencies) that set-top boxes can search to find the DSG tunnel appropriate for their operation. To remove the A-DSG 1.2 channel list from the Cisco CMTS, us the no form of this command.
Step 8	**cable dsg timer inde** [**Tdsg1** *Tdsg1*]	[**Tdsg2** *Tdsg2*]	[**Tdsg3** *Tdsg3*]	[**Tdsg4** *Tdsg4*] **Example:** `Router(config)# cable dsg timer 1 Tdsg1 1 Tdsg2 2 Tdsg3 3 Tdsg4 4`	Configures the A-DSG 1.2 timer entry to be associated to the downstream channel, and encoded into the Downstream Channel Descriptor (DCD) message. To remove the cable DSG timer from the Cisco CMTS, use the no form of this command.
Step 9	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

What to do next

Troubleshooting Tips

Refer to **debug** and **show** commands in the How to Monitor and Debug the Advanced-mode DOCSIS Set-Top Gateway Feature, on page 920.

Adding DSG Tunnel Group to a Subinterface

This procedure adds a DSG tunnel group to a subinterface using the cable dsg tg group-id command. After adding the DSG tunnel-group to a subinterface, appropriate IP Internet Group Management Protocol (IGMP) static joins are created and forwarding of DSG traffic begins, if the downstream DSG is configured.

Before you begin

The downstream DSG should exist to create IGMP static joins.

Restriction You can associate a DSG tunnel group to only one subinterface within the same bundle interface.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**configureterminal**
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**interface bundle***bundle-subif-number*
	Example:
	`Router(config)# interface bundle 11.2`
	`Router(config-subif)#`
		Specifies the interface bundle and enters the subinterface configuration mode.

Step 4	**cable dsg tg***group-id* **Example:** `Router(config-subif)# cable dsg tg 1`	Adds a DSG tunnel group to a subinterface.
Step 5	end **Example:** `Router(config-subif)# end`	Returns to privileged EXEC mode.

Configuring the DSG Client Settings for Advanced-Mode DSG 1.2

After the global configurations and DSG client configurations are set for DSG 1.2 on the Cisco CMTS, use the following procedure to continue DSG 1.2 client configurations.

Restriction The **in-dcd ignore** option is not supported by DSG-IF-MIBS specification.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable dsg client-list** *client-list-id* **id-index** *id* {**application-id** *app-id* | **ca-system-id** *sys-id* | **mac-addr** *mac-addr* | **broadcast** [*broadcast-id*]} **Example:** `Router(config)# cable dsg client-list 1 id-index 1 mac-addr abcd.abcd.abcd`	Sets the DSG client parameters. This command is changed from earlier Cisco IOS Releases, and for DSG 1.2, this command specifies the optional broadcast ID to client ID broadcast type and vendor specific parameter index.
Step 4	**cable dsg client-list** *client-list-id* **id-index** *id* [**vendor-param** *vendor-group-id*] **Example:** `Router(config-if)# cable dsg client-list 1 id-index 1 vendor-param 1`	Sets vendor-specific parameters for the DSG client.
Step 5	**cable dsg tunnel** *tunnel id* **mac_addr** *mac addr* **tg** *tunnel-group* **clients** *client-list-id* [**enable** | **disable**] **Example:** `Router(config)# cable dsg tunnel mac-addr abcd.abcd.abcd tg 1 clients 1 enable`	This command is changed to associate a tunnel group and client-list ID to a DSG tunnel. Also, an optional QoS service class name can be associated to the tunnel. **Note** To associate a cable service class with an A-DSG tunnel on a Cisco CMTS router, use the cable dsg tunnel srv-class command in global configuration mode.
Step 6	**cable dsg cfr** *cfr index* [**dest-ip** {*ipaddr* |*hostname}]* [**tunnel** *tunnel-index*][**dest-port** *start end*]| [**priority** *priority*][**src-ip** {*ipaddr* |*hostname}* [**src-prefix-len** *length*]] [**enable** | **disable**] [**in-dcd {yes** | **no** | **ignore}**]	Specifies the DSG classifier index, with optional support for the DCD parameter, indicating whether or not to include the classifier in the DCD message.

	Command or Action	Purpose
	Example: `Router(config)# cable dsg cfr 1 dest-ip 224.225.225.225 tunnel 1 dest-port 4050 priority 2 src-ip ciscovideo.com src-prefix-len 24 enable`	**Note** When you use the **ignore** option, the DSG classifier is not included in the DCD message.

Step 7	**end**
	Example:
	`Router(config)# end`
	`Router#`
		Returns to privileged EXEC mode.

What to do next

Troubleshooting Tips

Refer to **debug** and **show** commands in the How to Monitor and Debug the Advanced-mode DOCSIS Set-Top Gateway Feature, on page 920.

Configuring Downstream DSG 1.2 Settings for Advanced-Mode DSG 1.2

When the global and client configurations are set for DSG 1.2 on the Cisco CMTS, use the following procedure to continue with DSG 1.2 downstream configurations.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configureterminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** {*slot* /*port* |*slot* /*subslot*/*port* } **Example:** `Router(config)# interface cable 8/1/1`	Enters interface configuration mode.
Step 4	**cable downstream dsg tg** *group-id* **[channel** *channel-id]* **Example:**	Associates the DSG tunnel group to the downstream interface. To remove this setting, use the **no** form of this command.

	Command or Action	Purpose
	`Router(config-if)# cable downstream dsg tg 1 channel 1`	
Step 5	**cable downstream dsg chan-list** *list-index* **Example:** `Router(config-if)# cable downstream dsg chan-list 2`	Associates the A-DSG channel list entry to a downstream channel, to be included in the DCD message. To remove this setting, use the **no** form of this command.
Step 6	**cable downstream dsg timer** *timer-index* **Example:** `Router(config-if)# cable downstream dsg timer 3`	Associates the DSG timer entry to a downstream channel, to be included in the DCD message. To remove this setting, use the no form of this command.
Step 7	**cable downstream dsg vendor-param** *vsif-grp-id* **Example:** `Router(config-if)# cable downstream dsg vendor-param 2`	Associates A-DSG vendor parameters to a downstream to be included in the DCD message. To remove this configuration from the Cisco CMTS, use the no form of this command.
Step 8	**cable downstream dsg** [**dcd-enable** | **dcd-disable**] **Example:** `Router(config-if)# cable downstream dsg dcd-enable`	Enables DCD messages to be sent on a downstream channel. This command is used when there are no enabled rules or tunnels for A-DSG currently on the Cisco CMTS. To disable DCD messages, use the **disable** form of this command.
Step 9	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring IP Multicast Operations

This section describes how to configure the operation of IP multicast transmissions on the cable and WAN interfaces on the Cisco CMTS. You should perform this configuration on each cable interface being used for DSG traffic and for each WAN interface that is connected to a network controller or Conditional Access (CA) server that is forwarding IP multicast traffic.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 2	**ip multicast-routing** **Example:** `Router(config)# ip multicast-routing`	Enables multicast routing on the router.
Step 3	**ip pim ssm** {**default** | **range**{*access-list* | *word*}} **Example:** `Router(config)# ip pim ssm range 4`	Defines the Source Specific Multicast (SSM) range of IP multicast addresses. To disable the SSM range, use the no form of this command. **Note** When an SSM range of IP multicast addresses is defined by the **ip pim ssm** command, no Multicast Source Discovery Protocol (MSDP) Source-Active (SA) messages will be accepted or originated in the SSM range.
Step 4	ip cef distributed **Example:** `Router(config)# ip cef distributed`	Enables Cisco Express Forwarding (CEF) on the route processor card. To disable CEF, use the no form of this command. For additional information about the **ip cef** command, refer to the following document on Cisco.com: • *Cisco IOS Switching Services Command Reference* , Release 12.3 http://www.cisco.com/en/US/docs/ios/12_3/switch/command/reference/swtch_i.html
Step 5	**interface bundle** *bundle-number* **Example:** `Router(config)# interface bundle 10`	Enters interface configuration mode for each interface bundle being used for DSG traffic.
Step 6	**ip pim** {**dense-mode** | **sparse-mode** | **sparse-dense-mode**} **Example:** `Router(config-if)# ip pim dense-mode`	Enables Protocol Independent Multicast (PIM) on the cable interface, which is required to use the DSG feature: **Note** You must configure this command on each interface that forwards multicast traffic.
Step 7	Repeat Step 5, on page 913 and Step 6, on page 913 for each cable interface that is being used for DSG traffic. Also repeat these steps on each W AN interface that is forwarding IP multicast traffic from the DSG network controllers and Conditional Access (CA) servers.	

	Command or Action	Purpose
Step 8	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Enabling DNS Query and DSG Name Process

The DSG name process enables the Cisco CMTS router to query the DNS server for faster classifier updates.

Before you begin

Ensure that the IP DNS-based hostname-to-address translation is configured on the Cisco CMTS router using the **ip domain-lookup** command in global configuration mode. This is configured by default, and the status is not displayed in the running configuration.

Procedure

	Command or Action	Purpose
Step 1	**configureterminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**ip domain-name** *name* **Example:** `Router(config)# ip domain-name cisco.com`	Sets the IP domain name that the Cisco IOS software uses to complete unqualified host names
Step 3	r **ip name-serve***server-address*[**multiple-server-addresses**] **Example:** `Router(config)# ip name-server 131.108.1.111`	Sets the server IP address.
Step 4	**cable dsg name-update-interval***minutes* **Example:** `Router(config)# cable dsg name-update-interval 10`	Sets the interval to check the DNS server for any FQDN classifier changes.
Step 5	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Configuring NAT to Support Unicast Messaging

This section describes how to configure a Cisco CMTS router for Network Address Translation (NAT) to enable the use of IP unicast addresses for DSG messaging. This allows the Cisco CMTS router to translate incoming IP unicast addresses into the appropriate IP multicast address for the DSG traffic.

For the Cisco cBR-8 router, A-DSG 1.2 can use an external router that is close to the Cisco CMTS to support unicast messaging. In this case, the nearby router must support NAT, and then send the address-translated multicast IP packets to the Cisco CMTS.

Tip This procedure should be performed after the cable interface has already been configured for DSG operations, as described in the Configuration Examples for Advanced-Mode DSG, on page 923.

Note The Cisco CMTS router supports NAT only when it is running an "IP Plus" (-i-) Cisco IOS software image. Refer to the release notes for your Cisco IOS release for complete image availability and requirements.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**interface** *wan-interface* **Example:** `Router(config)# interface FastEthernet0/0`	Enters interface configuration mode for the specified WAN interface.
Step 3	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Configures the WAN interface as the "outside" (public) NAT interface.
Step 4	**interface bundle** *bundle-number* **Example:** `Router(config-if)# interface bundle 10`	Enters interface configuration mode for the specified interface bundle. **Note** This interface bundle should have previously been configured for DSG operations.
Step 5	**ip address** *ip-address mask* **secondary** **Example:**	Configures the cable interface with an IP address and subnet that should match the unicast address being used for DSG traffic.

	Command or Action	Purpose
	`Router(config-if)# ip address 192.168.18.1 255.255.255.0 secondary`	This IP address and its subnet must not be used by any other cable interfaces, cable modems, or any other types of traffic in the cable network.
Step 6	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Configures the cable interface as the "inside" (private) NAT interface.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**ip nat inside source static** *ip-multicast-address cable-ip-address* **Example:** `Router(config)# ip nat inside source static 224.3.2.1 192.168.18.2`	Maps the unicast IP address assigned to the cable interface to the multicast address that should be used for the DSG traffic.
Step 9	Repeat Step 2, on page 915 and Step 8, on page 916 for each cable interface to be configured for DSG unicast traffic.	
Step 10	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring WAN Interfaces for Multicast Operations

In addition to basic WAN interface configuration on the Cisco CMTS, described in other documents, the following WAN interface commands should be configured on the Cisco CMTS to support IP multicast operations with A-DSG 1.2, as required.

- **ip pim**
- **ip pim ssm**
- **ip cef**

These commands are described in the Configuring IP Multicast Operations, on page 912, and in the following documents on Cisco.com.

For additional information about the **ip pim** command, refer to the following document on Cisco.com:

- *Cisco IOS IP Command Reference, Volume 3 of 4* : Multicast, Release 12.3

http://www.cisco.com/en/US/docs/ios/12_3/ipmulti/command/reference/iprmc_r.html

For additional information about the **ip pim ssm** command, refer to the following document on Cisco.com:

- *Cisco IOS IP Command Reference, Volume 3 of 4: Multicast* , Release 12.3 T

http://www.cisco.com/en/US/docs/ios/12_3t/ip_mcast/command/reference/ip3_i2gt.html

For additional information about the **ip cef** command, refer to the following document on Cisco.com:

- *Cisco IOS Switching Services Command Reference* , Release 12.3

http://www.cisco.com/en/US/docs/ios/12_3/switch/command/reference/swtch_r.html

Configuring a Standard IP Access List for Packet Filtering

This section describes how to configure a standard IP access list so that only authorized traffic is allowed on the cable interface.

Tip This procedure assumes a basic knowledge of how access lists use an IP address and bitmask to determine the range of IP addresses that are allowed access. For full details on configuring access lists, see the documents listed in the Additional References, on page 926.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**access-list** *access-list* **permit** *group-ip-address* [*mask*] **Example:** `Router(config)# access-list 90 permit 228.1.1.1`	Creates an access list specifying that permits access to the specific multicast address that matches the specified *group-ip-address* and *mask* .
Step 3	**access-list** *access-list* **deny** *group-ip-address* [*mask*] **Example:** `Router(config)# access-list 90 deny 224.0.0.0 15.255.255.255`	Configures the access list that denies access to any multicast address that matches the specified *group-ip-address* and *mask* .
Step 4	**access-list** *access-list* **deny any** **Example:** `Router(config)# access-list 90 deny any`	Configures the access list so that it denies access to any IP addresses other than the ones previously configured.
Step 5	**interface bundle** *bundle-number* **Example:**	Enters interface configuration mode for the specified interface bundle.

	Command or Action	Purpose
	`Router(config)# interface bundle 10`	
Step 6	**ip access-group** *access-list* **Example:** `Router(config-if)# ip access-group 90`	(Optional, but recommended) Configures the interface with the access list, so that packets are filtered by the list before being accepted on the interface. **Note** Standard Access lists only allow one address to be specified in the earlier step. If you apply an outbound access-list with only the multicast address of the tunnel denied, then the DSG traffic is not allowed to pass. **Note** On the Cisco cBR-8 router, inbound access lists on the cable interface do not apply to multicast traffic, so they do not apply here. As a result, the Cisco cBR-8 requires that you use extended access lists that are blocked in the outbound direction for packets originating from the cable modem or CPE device on the network, and destined to the multicast group. The multicast group contains the classifiers associated with A-DSG 1.1 rules enabled on the interface.
Step 7	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to Privileged EXEC mode.

Configuring a Standard IP Access List for Multicast Group Filtering

This section describes how to configure a standard IP access list so that non-DOCSIS devices, such as DSG set-top boxes, can access only the authorized multicast group addresses and DSG tunnels.

Tip This procedure assumes a basic knowledge of how access lists use an IP address and bitmask to determine the range of IP addresses that are allowed access. For full details on configuring access lists, see the documents listed in the Additional References, on page 926.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**access-list** *access-list* **permit** *group-ip-address* [*mask*] **Example:** `Router(config)# access-list 90 permit 228.1.1.1`	Creates an access list specifying that permits access to the specific multicast address that matches the specified *group-ip-address* and *mask* .
Step 3	**access-list** *access-list* **deny** *group-ip-address* [*mask*] **Example:** `Router(config)# access-list 90 deny 224.0.0.0 15.255.255.255`	Configures the access list that denies access to any multicast address that matches the specified *group-ip-address* and *mask* .
Step 4	**access-list** *access-list* **deny any** **Example:** `Router(config)# access-list 90 deny any`	Configures the access list so that it denies access to any IP addresses other than the ones previously configured.
Step 5	**interface cable** *interface* **Example:** `Router(config)# interface cable 3/0`	Enters interface configuration mode for the specified cable interface.
Step 6	**ip igmp access-group** *access-list* [*version*] **Example:** `Router(config-if)# ip igmp access-group 90`	(Optional, but recommended) Configures the interface to accept traffic only from the associated access list, so that only authorized devices are allowed to access the DSG tunnels.
Step 7	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Disabling A-DSG Forwarding on the Primary Channel

You can disable A-DSG forwarding per primary capable interface.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**interface modular-cable** *slot* /*subslot*/*port* :*interface-number* **Example:** `Router(config)# interface modular-cable 1/0/0:0`	Specifies the modular cable interface and enters cable interface configuration mode. Variables for this command may vary depending on the Cisco CMTS router and the Cisco IOS-XE software release.
Step 3	**cable downstream dsg disable** **Example:** `Router(config-if)# cable downstream dsg disable`	Disables A-DSG forwarding and DCD messages on the primary capable interface.
Step 4	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

How to Monitor and Debug the Advanced-mode DOCSIS Set-Top Gateway Feature

This section describes the following commands that you can use to monitor and display information about the Advanced-mode DOCSIS Set-Top Gateway feature:

Displaying Global Configurations for Advanced-Mode DSG 1.2

The following commands display globally-configured or interface-level DSG settings, status, statistics, and multiple types of DSG 1.2 tunnel information.

show cable dsg cfr

To verify all DSG classifier details, such as the classifier state, source, and destination IP addresses, use the **show cable dsg cfr** command.

To verify details of a particular DSG classifier, use the **show cable dsg cfr** *cfr-id* command.

To verify the detailed output for all DSG classifiers, use the **show cable dsg cfr verbose** command.

To verify the detailed output for a single DSG classifier, use the **show cable dsg cfr** *cfr-id* **verbose** command.

show cable dsg host

To verify the mapping of the DSG hostnames and IP addresses on a Cisco CMTS router, use the **show cable dsg host** command.

To verify the verbose output of the mapping of the DSG hostnames and IP addresses on a Cisco CMTS router, use the **show cable dsg host verbose** command.

show cable dsg tunnel

To display tunnel MAC address, state, tunnel group id, classifiers associated to tunnel and its state, use the **show cable dsg tunnel** command in privileged EXEC mode. This command also displays the number of interfaces to which a tunnel is associated, the clients associated, and the QoS service class name for all the configured tunnels.

To display information for a given DSG tunnel, use the **show cable dsg tunnel** *tunnel-id* command, specifying the tunnel for which to display information.

show cable dsg tunnel *tunnel-id* [**cfr** | **clients** | **interfaces** | **statistics** | **verbose**]

- **cfr**—Shows DSG tunnel classifiers.
- **clients**—Shows DSG tunnel clients.
- **interfaces**—Shows DSG tunnel interfaces.
- **statistics**—Shows DSG tunnel statistics.
- **verbose**—Shows DSG tunnel detail information.

show cable dsg tg

To display the configured parameters for all DSG tunnel groups, use **show cable dsg tg** command.

Note The **Chan state** column in the **show cable dsg tg** command output indicates that a channel belonging to a tunnel group is either enabled or diabled. It is possible that a tunnel group is enabled but a particular channel in that tunnel group is disabled.

To display the configured parameters for the specified tunnel group, use **show cable dsg tg** *tg-id* **channel** *channel-id* command.

To display detailed information for the specified tunnel group, use **show cable dsg tg** *tg-id* **channel** *channel-id* **verbose** command.

show running-config interface

To display a tunnel group attached to a subinterface, use the **show running-config interface** command in privileged EXEC mode, as shown in the example below:

```
Router# show running-config interface bundle 11.2
!
interface Bundle11.2
 ip address 4.4.2.1 255.255.255.0
 no ip unreachables
 ip pim sparse-mode
 ip igmp static-group 230.1.1.30
 no cable ip-multicast-echo
```

```
 cable dsg tg 61
end
```

Note The IGMP static group IP address created automatically at the time of DSG configuration is not displayed in the **show running-config interface** command output.

show cable dsg static-group bundle

To verify all DSG static groups configured under a bundle interface, use the **show cable dsg static-group bundle** command in privileged EXEC mode.

Displaying Interface-level Configurations for Advanced-Mode DSG 1.2

The following **show** commands display interface-level configurations for A-DSG 1.2.

show cable dsg tunnel interfaces

To display all interfaces and DSG rules for the associated tunnel, use the **show cable dsg tunnel interfaces** command in privileged EXEC mode.

show cable dsg tunnel (*tunnel-id*) **interfaces**

show interfaces cable dsg downstream

To display DSG downstream interface configuration information, to include the number of DSG tunnels, classifiers, clients, and vendor-specific parameters, use the **show interfaces cable dsg downstream** command in privileged EXEC mode.

show interfaces cable dsg downstream dcd

To display DCD statistics for the given downstream, use the **show interfaces cable dsg downstream dcd** command in privileged EXEC mode. This command only displays DCD Type/Length/Value information if the **debug cable dsg** command is previously enabled.

show interfaces cable dsg downstream tg

To display DSG tunnel group parameters, and rule information applying to the tunnel group, to include tunnels and tunnel states, classifiers, and client information, use the **show interfaces cable dsg downstream tg** command in privileged EXEC mode. You can display information for a specific tunnel, if specified.

show interfaces cable dsg downstream tunnel

To display DSG tunnel information associated with the downstream, use the **show interfaces cable dsg downstream tunnel** command in privileged EXEC mode.

Debugging Advanced-Mode DSG

To enable debugging for A-DSG on a Cisco CMTS router, use the debug cable dsg command in privileged EXEC mode.

Configuration Examples for Advanced-Mode DSG

This configuration example illustrates a sample DSG network featuring these components:

- Two Cisco universal broadband routers
- IP Multicast for each DSG implementation
- Two DSG Clients for each Cisco CMTS
- Two DSG Servers (one for each Cisco CMTS)

Each Cisco CMTS is configured as follows, and the remainder of this topic describes example configurations that apply to this architecture.

CMTS Headend 1

- DSG Server #1—Connected to Cisco CMTS via IP Multicast, with DSG Server having IP Address 12.8.8.1
- Destination IP Address for the Cisco CMTS—228.9.9.1
- DSG Tunnel Address—0105.0005.0005
- Downstream #1 Supporting two DSG Clients:
    - DSG Client #1—ID 101.1.1
    - DSG Client #2—ID 102.2.2

CMTS Headend 2

- DSG Server #2—Connected to Cisco CMTS via IP Multicast, with DSG Server having IP Address 12.8.8.2
- Destination IP Address for the Cisco CMTS—228.9.9.2
- DSG Tunnel Address—0106.0006.0006
- Downstream #2 Supporting two DSG Clients:
    - DSG Client #1—ID 101.1.1
    - DSG Client #2—ID 102.2.2

Example of Two DSG Tunnels with MAC DA Substitution

In this configuration, and given the two Cisco CMTS Headends cited above, below are the two sets of DSG rules, with each set applying to each Cisco CMTS, in respective fashion.

These settings apply to DSG #1 and two downstreams:

- DSG Rule ID 1
- DSG Client ID 101.1.1
- DSG Tunnel Address 105.5.5

These settings apply to DSG Rule #2 and two downstreams:

- DSG Rule ID 1
- DSG Client ID 102.2.2
- DSG Tunnel Address 106.6.6

DSG Example with Regionalization Per Downstream

In this configuration, and given the two Cisco CMTS Headends cited earlier in this topic, below are two downstream rules that can be configured in this architecture, for example:

- Downstream Rule #1
  - DSG Rule ID #1
  - DSG Client ID—101.1.1
  - DSG Tunnel Address—105.5.5
- Downstream Rule #2
  - DSG Rule ID #2
  - DSG Client ID—102.2.2
  - DSG Tunnel Address—106.6.6

DSG Example with Regionalization Per Upstream

In this configuration, and given the two Cisco CMTS Headends cited earlier in this topic, below are two upstream rules that can be configured in this architecture, for example:

- Upstream Rule #1
  - DSG Rule ID #1
  - DSG Client ID—101.1.1
  - DSG UCID Range—0 to 2
  - DSG Tunnel Address—105.5.5
- Upstream Rule #2
  - DSG Rule ID #2
  - DSG Client ID—102.2.2
  - DSG UCID Range—3 to 5
  - DSG Tunnel Address—106.6.6

Example of Two DSG Tunnels with Full Classifiers and MAC DA Substitution

In this configuration, and given the two Cisco CMTS Headends cited above, below are the two sets of DSG rules, with each set applying to each Cisco CMTS, in respective fashion.

These settings apply to DSG #1:

- DSG Rule ID 1
- Downstreams 1 and 2
- DSG Client ID 101.1.1
- DSG Tunnel Address 105.5.5
- DSG Classifier ID—10
- IP SA—12.8.8.1
- IP DA—228.9.9.1
- UDP DP—8000

These settings apply to DSG Rule #2:

- DSG Rule ID 2

- Downstreams 1 and 2
- DSG Client ID 102.2.2
- DSG Tunnel Address 106.6.6
- DSG Classifier ID—20
- IP SA—12.8.8.2
- IP DA—228.9.9.2
- UDP DP—8000

Example of One DSG Tunnel Supporting IP Multicast from Multiple DSG Servers

In this configuration, and given the two Cisco CMTS Headends cited earlier in this topic, below is an example of one DSG Tunnel with multiple DSG servers supporting IP Multicast:

- DSG Rule ID 1
- Downstreams 1 and 2
- DSG Client ID 101.1.1 and 102.2.2
- DSG Tunnel Address 105.5.5
- DSG Classifier ID—10
  - IP SA—12.8.8.1
  - IP DA—228.9.9.1
  - UDP DP—8000
- DSG Classifier ID—20
  - IP SA—12.8.8.2
  - IP DA—228.9.9.2
  - UDP DP—8000

Example: Enabling DNS Query

The following example shows how to enable a DNS query on the Cisco CMTS router:

```
Router# configure terminal
Router(config)# ip domain-lookup
Router(config)# ip domain-name cisco.com
Router(config)# ip name-server 131.108.1.111
Router(config)# cable dsg name-update-interval 10
Router(config)# end
```

Example: Disabling A-DSG Forwarding on the Primary Channel

The following example shows how to disable A-DSG forwarding on a primary capable modular interface on the Cisco CMTS router:

```
Router# configure terminal
Router(config)# interface modular-cable 1/0/0:0
Router(config-if)# cable downstream dsg disable
Router(config-if)# end
```

Additional References

The following sections provide references related to A-DSG 1.2.

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

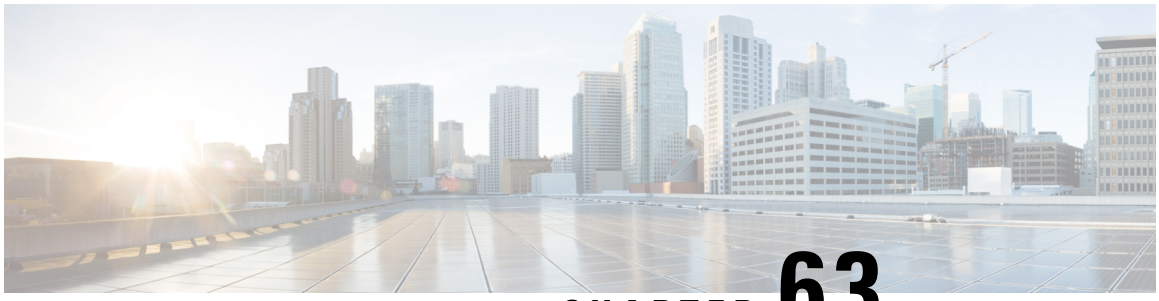
Feature Information for Advanced-Mode DSG 1.2 for the Cisco CMTS Routers

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 154: Feature Information for DOCSIS Set-Top Gateway and A-DSG for the Cisco CMTS Routers

Feature Name	Releases	Feature Information
DOCSIS Set-Top Gateway for the Cisco CMTS Routers	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Routers.



CHAPTER 63

Cisco Network Registrar for the Cisco CMTS Routers

This chapter supplements the Cisco Network Registrar (CNR) documentation by providing additional cable-specific instructions to provision a hybrid fiber-coaxial (HFC) network using Cisco universal broadband routers as CMTSs at the headend of the network.

Note For information about the IPv6 provisioning on CNR server, please refer to IPv6 on Cable.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 928
- Servers Required on the HFC Network, on page 928
- Cisco Network Registrar Description, on page 929
- Overview of DHCP Using CNR, on page 930
- How Cisco Converged Broadband Routers and Cable Modems Work, on page 931
- DHCP Fields and Options for Cable Modems, on page 931
- Cisco Network Registrar Sample Configuration, on page 932
- Overview of Scripts, on page 936
- Placement of Scripts, on page 936
- Activating Scripts in Cisco Network Registrar, on page 937
- Configuring the Cisco CMTS Routers to Use Scripts, on page 937
- Configuring the System Default Policy, on page 937
- Creating Selection Tag Scopes, on page 938

• Creating Network Scopes, on page 939
• Creating Policies for Class of Service or for Upgrading Cable Modem Cisco IOS Images, on page 940
• CNR Steps to Support Subinterfaces, on page 940
• Additional References, on page 941

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

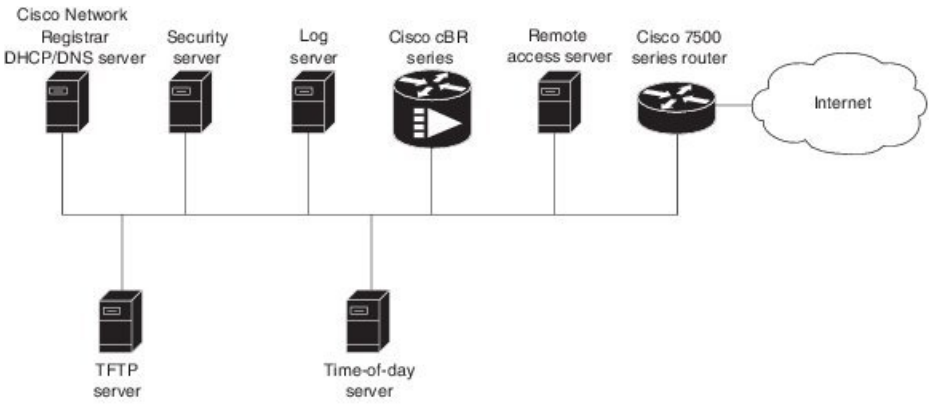
Table 155: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Servers Required on the HFC Network

A TFTP server, DHCP server, and time-of-day (TOD) server are required to support two-way data cable modems on an HFC network. A cable modem will not boot if these servers are not available. The log server and security servers are not required to configure and operate a cable modem. If the log server or security servers are not present, a cable modem will generate warning messages, but it will continue to boot and function properly.

Figure 29: Servers Required on a Two-Way HFC Network



The servers shown here can exist on the same platform. For example, the time-of-day server and the TFTP server can run on the same platform.

In this provisioning model, TOD and TFTP servers are standard Internet implementations of the RFC 868 and RFC 1350 specifications. Most computers running a UNIX-based operating system supply TOD and TFTP servers as a standard software feature. Typically, the TOD server is embedded in the UNIX *inetd* and it requires no additional configuration. The TFTP server is usually disabled in the standard software but can be enabled by the user. Microsoft NT server software includes a TFTP server that can be enabled with the services control panel. Microsoft NT does not include a TOD server. A public domain version of the TOD server for Microsoft NT can be downloaded from several sites.

The DHCP and Domain Name System (DNS) server shown in Figure above must be the DHCP/DNS server available in Cisco Network Registrar version 2.0 or later. CNR is the only DHCP server that implements policy-based assignment of IP addresses. The headend must be a Cisco cBR-8 converged broadband router. The remote access server is only required on HFC networks that are limited to one-way (downstream only) communication. In a one-way HFC network, upstream data from a PC through the headend to the Internet is carried over a dialup connection. This dialup connection for upstream data is referred to as telco return. For simplification, the model will not include a log or security server. Cable modems can be set up to use the logging and security servers by including the appropriate DHCP options in the cable modem policy as described in the *Cisco Network Registrar User Manual*.

Cisco Network Registrar Description

CNR is a dynamic IP address management system, running on Windows or Solaris, that uses the Dynamic Host Configuration Protocol (DHCP) to assign IP addresses to cable interfaces, PCs, and other devices on the broadband network. The CNR tool includes script extensions that allow a cable system administrator to define and view individual DHCP options, define the identity or type of device on the network, and assign the device to a predefined class or group.

Using the CNR tool, a cable system administrator can specify policies to provide:

- Integrated DHCP and Domain Name Server (DNS) services
- Time of Day (ToD) and Trivial File Transfer Protocol (TFTP) server based on the size of the network
- DHCP safe failover and dynamic DNS updates

Note This is available only in CNR 3.0 or higher.

Using the CNR tool and the extension scripts identified in the Overview of Scripts, on page 936 section, a cable system administrator can specify scopes, policies, and options for the network and each cable interface based on the services and configuration to support at each subscriber site.

Note Scopes refer to the administrative grouping of TCP/IP addresses; all IP addresses within a scope should be on the same subnet.

The cable system administrator defines system default policies for all standard options and uses scope-specific policies for options related to particular subnets, such as cable interfaces. This allows DHCP to send the information with the IP address.

Seven entry points exist for scripts:

- post-packet-decode
- pre-client-lookup
- post-client-lookup—Examines and takes action on results of the client-class process, places data items in the environment dictionary to use at the pre-packet-encode extension point, includes DHCP relay option
- check-lease-acceptable
- pre-packet-encode
- post-sent-packet
- pre-dns-add-forward

Overview of DHCP Using CNR

Cisco Network Registrar (CNR) is a dynamic IP address management system that uses the Dynamic Host Configuration Protocol (DHCP) and assigns IP addresses to PCs and other devices on a network based on a predefined set of policies, such as class of service. CNR assigns available IP addresses from address pools based on the identity or type of the requesting device and the policies in effect. For example, CNR can distinguish between registered devices, unregistered devices, and registered devices that have been assigned to a particular class of service.

CNR also provides extensions that can be customized (via programming or a script) so that you can view individual DHCP options, determine the identity or type of a device based on the content of the options, and assign a device to a predefined class or group. Using these extensions, you can determine the difference between PCs and cable modems and assign them IP addresses from different address pools.

In typical data-over-cable environments, service providers are interested in simplifying provisioning to limit the amount of information that must be collected about subscribers' customer premise equipment (CPEs). To support current provisioning models, a field technician must be sent to a subscriber's home or business to install and setup a cable modem. During this site visit, the technician might register the serial number and MAC address of the cable modem in the customer account database. Because a field technician must go to a subscriber's site to replace a cable modem, you can easily track modem information.

Manually registering and tracking information about a cable subscriber's PC is more difficult. A subscriber might purchase a new PC or exchange the network interface card (NIC) without notifying you of the change. Automatic provisioning with CNR reduces the amount of customer service involvement needed to track customer equipment. To use the provisioning model described in this document, you must still track serial numbers and MAC addresses for cable modems, but you do not need to track information about the PC or NIC cards installed at a subscriber site.

The remainder of this document describes how to configure CNR to support this model. The following sections describe the equipment and servers required for the cable headend, provide an overview of the interaction between DOCSIS-compatible cable modems and the Cisco universal broadband routers, and provide a guide on how to configure CNR to support this provisioning model.

How Cisco Converged Broadband Routers and Cable Modems Work

Cisco converged broadband routers and cable modems are based on the Data Over Cable Service Interface Specification (DOCSIS) standards. These standards were created by a consortium of cable service providers called Multimedia Cable Network Systems, Ltd. (MCNS) to that cable headend and cable modem equipment produced by different vendors will interoperate. The key DOCSIS standards provide the basis for a cable modem to communicate with any headend equipment and headend equipment to communicate with any cable modem.

Cable modems are assigned to operate on specific cable channels so activity can be balanced across several channels. Each Cisco cBR-8 router installed at the headend serves a specific channel. Part of network planning is to decide which channel each cable modem can use.

A cable modem cannot connect to the network until the following events occur:

- The cable modem initializes and ranges through available frequencies until it finds the first frequency that it can use to communicate to the headend. The cable modem might be another vendor's DOCSIS-compatible device and the headend might have a Cisco cBR-8 router installed. At this point on the initial connection, the cable modem cannot determine if it is communicating on the correct channel.
- The cable modem goes through the DHCP server process and receives a configuration file from the server.
- One of the parameters in the configuration file tells the cable modem which channel it can use.
- If the assigned channel is not available on the Cisco cBR-8 router to which the cable modem is currently connected, it resets itself and comes up on the assigned channel.
- During this second DHCP process, the modem will be connected to the correct CMTS. This time, the configuration file will be loaded. For a DOCSIS-compatible cable modem to access the network, it might go through the DHCP server two times on two different networks; therefore, one-lease-per-client IP addressing is critical.

DHCP Fields and Options for Cable Modems

DHCP options and packet fields are required to enable cable modems to boot and operate properly. Table below lists the required DHCP options and fields.

Table 156: Required DHCP Fields and Options

Required Field/Option	Field/Option In Cisco Network Registrar	Value/Description
Fields		
giaddr	-	IP address. As a DHCP packet passes through the relay agent to the DHCP server, the relay agent supplies a unique IP address to the packet and stores it in this field. The relay agent is a cBR-8 router with the iphelper attribute defined.
subnet-mask	-	Subnet mask for the IP address stored in the giaddr field. This value is also stored in the DHCP packet by the relay agent.
file	Packet-file-name	Name of the cable modem configuration file that will be read from a TFTP server.
siaddr	Packet-siaddr	IP address of the TFTP server where configuration files are stored.
Options		
Time-servers	-	List of hosts running the time server specified in the RFC 868 standard.
Time-offset	-	Time offset of a cable modem internal clock from Universal Time Coordinated (UTC). This value is used by cable modems to calculate the local time that is stored in time-stamping error logs.
MCNS-security-server	-	IP address of the security server. This should be set if security is required. See RFC 1533 for details.

Cisco Network Registrar Sample Configuration

You can use the following information to set up Cisco Network Registrar in a trial configuration. The configuration describes DHCP-related setup only; it does not cover setting up DNS or configuring dynamic DNS (DDNS). You should be familiar with important CNR concepts including scopes, primary and secondary scopes, scope selection tags, client classes, and CNR policies. See the Using Network Registrar publication for detailed information on these concepts.

In the trial configuration, you can configure CNR to perform the following operations:

- Receive DHCP requests from a cable modem and a PC on an HFC network via a port supporting multiple network numbers. The Cisco cBR-8 router at the headend must be configured as a forwarder (iphelper is configured).
- Serve IP addresses on two networks; a net-10 network (non-Internet routable) and a net-24 network (Internet routable).
- Tell the difference between a cable modem and a PC based on the MAC address of the device and provide net-24 addresses to the PC and net-10 addresses to the cable modem.
- Refuse to serve IP addresses to MAC addresses that it does not recognize.

To perform these options, you must implement the following CNR configuration items:

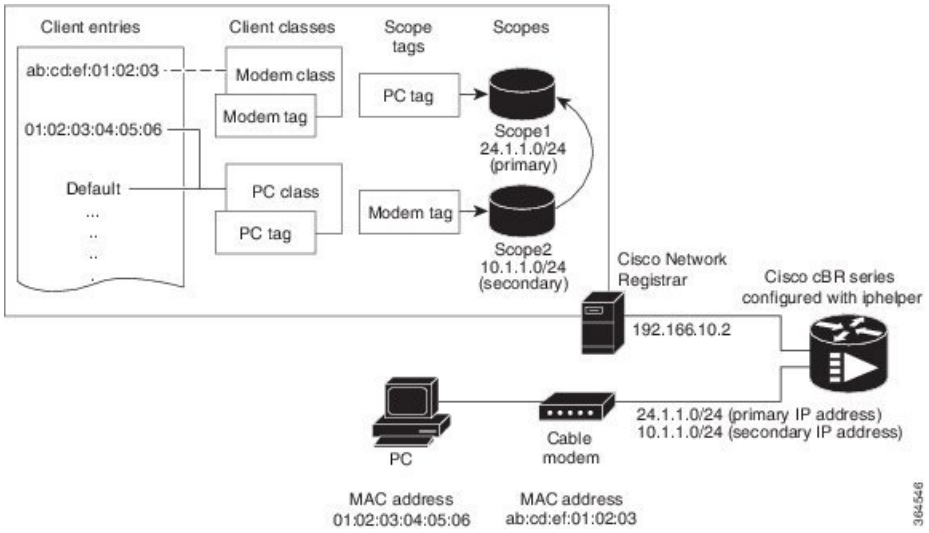
- Create two scope selection tags; one for PCs, one for cable modems.
- Create two client-classes; one for PCs , one for cable modems.
- Create a lease policy appropriate for the cable modem devices.
- Create a lease policy appropriate for the PC devices.
- Create a scope containing Class A net-24 (routable) addresses.
- Create a scope containing Class A net-10 (nonroutable) addresses.
- Identify the scope containing the net-24 addresses as the primary scope and configure the other scope containing the net-10 addresses as secondary to the net-24 scope.

Note The Cisco cBR-8 router upstream ports must be configured with the primary network address on the net-24 network; such as 24.1.1.1.

- Assign the policies to the appropriate scope.
- Add the MAC address of the cable modem and the PC to the client-entry list.
- Associate the PC tag with the scope containing routable addresses.
- Associate the cable modem tag with the scope containing nonroutable addresses.
- Associate the cable modem tag with the cable modem client-class.
- Associate the PC tag with the PC client-class.
- Assign the PC MAC to the PC class.
- Assign the cable modem MAC to the cable modem class.
- Enable client-class processing.

Figure below shows the trial CNR configuration in an HFC network.

Figure 30: Trial Configuration in an HFC Network



These configuration items and their associations can be created using either the CNR management graphical user interface (GUI) or command-line interface (CLI). The following sample script configures DHCP for a sample server:

```
File: cabledemo.rc
Command line: nrcmd -C <cluster> -N <user name> -P <password> -b < cabledemo.rc
-----------------------------------------------------------------------------------------
scope-selection-tag tag-CM create
scope-selection-tag tag-PC create
client-class create class-CM
client-class class-CM set selection-criteria=tag-CM
client-class create class-PC
client-class class-PC set selection-criteria=tag-PC
policy cmts-cisco create
policy cmts-cisco setleasetime 1800
policy cmts-cisco setoption domain-name-servers 192.168.10.2
policy cmts-cisco setoption routers 10.1.1.1
policy cmts-cisco setoption time-offset  604800
policy cmts-cisco setoption time-servers 192.168.10.20
policy cmts-cisco set packet-siaddr=192.168.10.2
policy cmts-cisco setoption log-servers 192.168.10.2
policy cmts-cisco setoption mcns-security-server 192.168.10.2
policy cmts-cisco set packet-file-name=golden.cfg
policy cmts-cisco set dhcp-reply-options=packet-file-name,packet-siaddr,mcns-security-server
policy pPC create
policy pPC set server-lease-time 1800
policy pPC setleasetime 1800
policy pPC setoption domain-name-servers 192.168.10.2
policy pPC setoption routers 24.1.1.1
scope S24.1.1.0 create 24.1.1.0 255.255.255.0
scope S24.1.1.0 addrange 24.1.1.5 24.1.1.254
scope S24.1.1.0 set policy=pPC
scope S24.1.1.0 set selection-tags=tag-PC
scope S10.1.1.0 create 10.1.1.0 255.255.255.0
scope S10.1.1.0 addrange 10.1.1.5 10.1.1.254
scope S10.1.1.0 set policy=cmts-cisco
scope S10.1.1.0 set selection-tags=tag-CM
scope S10.1.1.0 set primary-scope=S24.1.1.0
client 01:02:03:04:05:06 create client-class-name=class-PC
```

```
client ab:cd:ef:01:02:03 create client-class-name=class-CM
client default create action=exclude
dhcp enable client-class
dhcp enable one-lease-per-client
save
dhcp reload
```

In addition to the DHCP server setup, you might want to enable packet-tracing. When packet-tracing is enabled, the server parses both requests and replies, and then adds them to the logs. If you do enable tracing, performance will be adversely affected, and the logs will roll over quickly.

Use the following nrcmd command to set packet tracing.

DHCP set log-settings=incoming-packet-detail,outgoing-packet-detail

Cable Modem DHCP Response Fields

Each cable interface on the broadband network requires the following fields in the DHCP response:

- CM's IP address
- CM's subnet mask

Note For cable operators with less experience in networking, you can fill in a guess based on the network number and indicate how your IP network is divided.

- Name of the DOCSIS configuration file on the TFTP server intended for the cable interface
- Time offset of the cable interface from the Universal Coordinated Time (UTC), which the cable interface uses to calculate the local time when time-stamping error logs
- Time server address from which the cable interface obtains the current time

DOCSIS DHCP Fields

DOCSIS DHCP option requirements include:

- IP address of the next server to use in the TFTP bootstrap process; this is returned in the siaddr field
- DOCSIS configuration file that the cable interface downloads from the TFTP server

Note If the DHCP server is on a different network that uses a relay agent, then the relay agent must set the gateway address field of the DHCP response.

- IP address of the security server should be set if security is required

DHCP Relay Option (DOCSIS Option 82)

DOCSIS Option82 modifies DHCPDISCOVER packets to distinguish cable interfaces from the CPE devices or "clients" behind them. The DOCSIS Option82 is comprised of the following two suboptions:

- Suboption 1, Circuit ID:

```
Type 1 (1 byte)
Len 4 (1 byte)
Value (8 bytes)
<bit 31,30,.....................0)
<xYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY>
```

where the MSB indicates if the attached device is a cable interface.

x=1 Cable Modem REQ

x=0 CPE device (Behind the cable interface with the cable interface MAC address shown in suboption 2.)

The rest of the bits make up the SNMP index to the CMTS interface.

Y=0xYYYYYYY is the SNMP index to the CMTS interface.

- Suboption 2, MAC address of the cable interface:

```
Type 2 (1 byte)
Len 6 (1 byte)
Value xxxx.xxxx.xxxx (6 bytes)
```

Overview of Scripts

This section lists the scripts applicable to cable interface configuration.

Two-way Cable Modem Scripts

To support two-way configurations at a subscriber site, use these scripts:

- **Relay.tcl**
- **SetRouter.tcl**

Telco Return Cable Modem Scripts

To support telco return and two-way cable interface configurations on the same cable interface card or chassis, use these scripts:

- **PostClientLookup.tcl**
- **PrePacketEncode.tcl**

Placement of Scripts

Windows NT

For CNR running on Windows NT, place the appropriate scripts in the following directory:

```
\program files\network registrar\extensions\dhcp\scripts\tcl
```

Solaris

For CNR running on Solaris, place the appropriate scripts in the following directory:

```
/opt/nwreg2/extensions/dhcp/scripts/tcl
```

Activating Scripts in Cisco Network Registrar

To activate the scripts after you have placed them in the appropriate directory:

Procedure

Step 1 Open up a text editor.

Step 2 Open one of the scripts at the nrcmd> command prompt.

Step 3 Create the extension points and attach them to the system.

Note The easiest way to do this is to simply cut and paste the command lines from the scripts to the nrcmd> command line.

Step 4 After you have created and attached the extension points, do a dhcp reload.

The scripts are active.

Configuring the Cisco CMTS Routers to Use Scripts

Each cable interface must be set up as a BOOTP forwarder and have the relay option enabled. The primary and secondary IP addresses for each cable interface must be in sync with the CNR tool.

To properly communicate with scripts in the system, use the following commands on the Cisco CMTS router:

- To enable option 82, use the **ip dhcp relay info option** command.
- To disable the validation of DHPC relay agent information in forwarded BOOTREPLY messages, use the **no ip dhcp relay information option check** command.

Note You can also use the cable dhcp-giaddr command in cable interface configuration mode to modify the GIADDR field of DHCPDISCOVER and DHCPREQUEST packets to provide a relay IP address before packets are forwarded to the DHCP server. Use this command to set a “policy” option such that primary addresses are used for CMs and secondary addresses are used for hosts behind the CMs.

Configuring the System Default Policy

Add these options to the system default policy for:

- Cable modems to support on your network
- PCs to support behind each cable interface on your network

Cable Modems

Define these settings following the CNR tool documentation:

- TFTP server (IP address) for those cable interfaces using BOOTP
- Time-server (IP address)
- Time-offset (Hex value, 1440 for Eastern Standard Time)
- Packet-siaddr (IP address of CNR)
- Router (set to 0.0.0.0)
- Boot-file (name of .cm file for those cable interfaces using BOOTP)
- Packet-file-name (.cm file name)

PCs

Define these settings following the CNR tool documentation:

- Domain name
- Name servers (IP address of DNS servers)

Creating Selection Tag Scopes

General

When you create your scope selection tags:

Procedure

Step 1 Cut and paste the scope selection tag create commands from the scripts into the nrcmd> command line.

Note These names have to be exactly as they appear in the scripts.

Step 2 Then attach the selection tags to the appropriate scripts:

Example:

CM_Scope tagCablemodem

PC_Scope tagComputer

Telco Return for the Cisco cBR-8 Router

Before you begin

Note If you are using the prepacketencode and postclientlookup .tcl scripts for telco return, the telco return scope does not have a selection tag associated to the scope.

Procedure

Step 1 Put the tag Telcocablemodem on the primary cable interface scope to pull addresses from that pool instead.

Step 2 Follow the same procedure as above, but use a telco return policy which has a different .cm file with telco-specific commands in it.

Creating Network Scopes

Following is an example for creating scopes for your network. This example assumes two Cisco cBR-8 converged broadband routers in two locations, with one cable interface card on one Cisco cBR-8 configured for telco return.

```
cm-toledo1_2-0 10.2.0.0 255.255.0.0 assignable 10.2.0.10-10.2.254.254 tagCablemodem
tagTelcomodem Default GW=10.2.0.1 (assigned by scripts)
cm-toledo1_3-0 10.3.0.0 255.255.0.0 assignable 10.3.0.10-10.3.254.254 tagCablemodem
tagTelcomodem Default GW=10.3.0.1 (assigned by scripts)
pc-toledo1_2-0 208.16.182.0 255.255.255.248 assignable 208.16.182.2-208.16.182.6 tagComputer
 Default GW=208.16.182.1 (assigned by scripts)
pc-toledo1_3-0 208.16.182.8 255.255.255.248 assignable 208.16.182.10-208.16.182.14 tagComputer
 Default GW=208.16.182.9 (assigned by scripts)
telco_return_2-0 192.168.1.0 255.255.255.0 (No assignable addresses, tag was put on cable
modem primary scope to force telco-return cable modem to pull address from primary scope)
cm-arlington1_2-0 10.4.0.0 255.255.0.0 assignable 10.4.0.10-10.4.254.254 tagCablemodem
Default GW=10.4.0.1 (assigned by scripts)
cm-arlington1_3-0 10.5.0.0 255.255.0.0 assignable 10.5.0.10-10.5.254.254 tagCablemodem
Default GW=10.5.0.1 (assigned by scripts)
pc-arlington1_2-0 208.16.182.16 255.255.255.248 assignable 208.16.182.17-208.16.182.22
tagComputer Default GW=208.16.182.17 (assigned by scripts)
pc-toledo1_3-0 208.16.182.24 255.255.255.248 assignable 208.16.182.2-208.16.182.30 tagComputer
 Default GW=208.16.182.25 (assigned by scripts)
```

Note Remember the last valid address in the .248 subnet range is the broadcast address; do not use this.

Creating Policies for Class of Service or for Upgrading Cable Modem Cisco IOS Images

To support Class of Service (CoS), define:

- Scope selection tags—Identifiers that describe types of scope configurations

Note This is needed for Option82.

- Client classes—Class with which a group of clients is associated

Note Scope selection tags are excluded from or included in client-classes.

- Client—Specific DHCP clients and the defined class to which they belong

To assign the CoS or use Option82, make a client entry with a MAC address and point to the appropriate policy. To use client-based MAC provisioning, add a client entry "default - exclude," then put in MAC addresses for all devices (for example, cable interfaces and PCs) in the client tab and select the policy to use, including the appropriate tag.

CNR Steps to Support Subinterfaces

The CNR configuration is done differently if subinterfaces are configured. Here is an example. If you have configured two ISP subinterfaces and one management subinterface on a Cisco cBR-8 router, make sure that the management subinterface is the first subinterface that is configured. If cable interface three—c3/0/0—is being used, create c3/0/0.1, c3/0/0.2 and c3/0/0.3 as three subinterfaces and c3/0/0.1 as the first subinterface configured as the management subinterface.

Note The Cisco cBR-8 router requires management subinterfaces to route DHCP packets from CMs when they first initialize because the Cisco cBR-8 router does not know the subinterfaces they belong to until it has seen the IP addresses assigned to them by gleaning DHCP reply message from CNR.

In CNR, complete the following steps for such a configuration:

Procedure

Step 1 Create two scope selection tags such as: isp1-cm-tag and isp2-cm-tag

Step 2 Configure three scopes; for example, mgmt-scope, isp1-cm-scope, and isp2-cm-scope such that isp1-cm-scope and isp2-cm-scope each define mgmt-scope to be the primary scope

Step 3 Also configure two scopes for PCs for each of the ISPs; isp1-pc-scope and isp2-pc-scope. For scope isp1-cm-scope, configure isp1-cm-tag to be the scope selection tag. For scope isp2-cm-scope, configure isp2-cm-tag to be the scope selection tag

Step 4 Configure two client classes; for example, isp1-client-class and isp2-client-class

Step 5 Create client entries with their MAC addresses for CMs that belong to ISP1 and assign them to isp1-client-class. Also assign the scope selection tag isp1-cm-tag

Step 6 Create client entries for CMs that belong to ISP2 and assign them to isp2-client-class. Also assign the scope selection tag isp2-cm-tag

Step 7 Enable client class processing from the scope-selection-tag window

Overlapping address ranges cannot be configured on these subinterfaces because software gleans the DHCP reply to figure out the subinterface it really belongs to. Although CNR can be configured with overlapping address range scopes, it cannot be used to allocate addresses from these scopes.

Additional References

The following sections provide references related to Cisco Network Registrar for use with the Cisco CMTS routers.

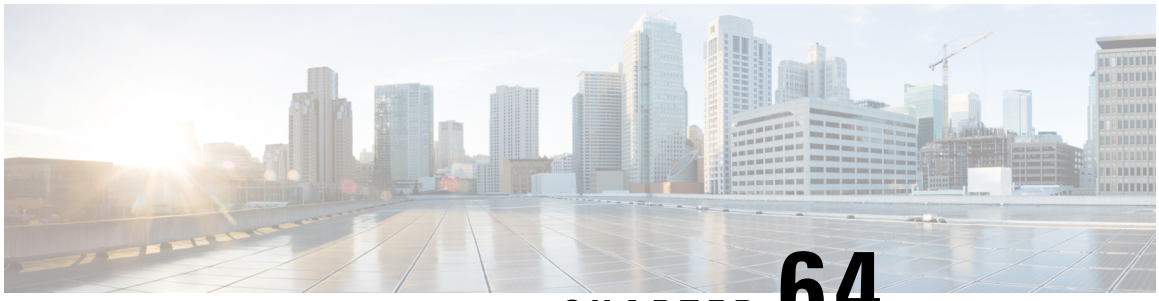
Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/cisco/web/support/index.html

PART IX

PacketCable and PacketCable Multimedia Configuration

- PacketCable and PacketCable Multimedia, on page 945
- COPS Engine Operation, on page 981



CHAPTER 64

PacketCable and PacketCable Multimedia

This document describes how to configure the Cisco CMTS for PacketCable and PacketCable Multimedia operations over an existing DOCSIS (1.1and later versions) network.

- Finding Feature Information, on page 945
- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 946
- Restrictions for PacketCable Operations, on page 946
- Information About PacketCable Operations, on page 947
- How to Configure PacketCable Operations, on page 953
- Configuration Examples for PacketCable, on page 960
- Verifying PacketCable Operations, on page 962
- Information About PacketCable Multimedia Operations, on page 966
- How to Configure PCMM Operations, on page 969
- Configuration Examples for PacketCable Multimedia, on page 972
- Verifying PCMM Operations, on page 972
- High Availability Stateful Switchover (SSO) for PacketCable and PacketCable MultiMedia, on page 974
- Voice MGPI Support, on page 975
- Additional References, on page 977
- Feature Information for PacketCable and PacketCable Multimedia, on page 978

Finding Feature Information

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 157: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for PacketCable Operations

- Supports only embedded multimedia terminal adapter (E-MTA) clients. Standalone MTA (S-MTA) clients are not supported.
- PacketCable operations can be configured together with HCCP N+1 redundancy, but the PacketCable states are not synchronized between the Working and Protect interfaces. If a switchover occurs, existing voice calls continue, but when the user hangs up, PacketCable event messages are not generated because the Protect interface is not aware of the previous call states. However, new voice calls can be made and proceed in the normal fashion.
- The 200,000 Hz channel width cannot be used on upstreams that support PacketCable voice calls, or on any upstreams that use Unsolicited Grant Service (UGS) or UGS with Activity Detection (UGS-AD)

service flows. Using this small a channel width with voice and other UGS/UGS-AD service flows results in calls being rejected because of "DSA MULTIPLE ERRORS".

Information About PacketCable Operations

This section provides an overview and other information about PacketCable operations, the components of a PacketCable network, and how they interact with the other components of a DOCSIS cable networks.

Feature Overview

PacketCable is a program initiative from Cablelabs and its associated vendors to establish a standard way of providing packet-based, real-time video and other multimedia traffic over hybrid fiber-coaxial (HFC) cable networks. The PacketCable specification is built upon the Data-over-Cable System Interface Specifications (DOCSIS) 1.1, but it extends the DOCSIS protocol with several other protocols for use over noncable networks, such as the Internet and the public switched telephone network (PSTN).

This allows PacketCable to be an end-to-end solution for traffic that originates or terminates on a cable network, simplifying the task of providing multimedia services over an infrastructure composed of disparate networks and media types. It also provides an integrated approach to end-to-end call signaling, provisioning, quality of service (QoS), security, billing, and network management.

Emergency 911 Features

PacketCable Emergency 911 Cable Interface Line Card Prioritization

The PacketCable Emergency 911 cable interface line card prioritization feature enables cable interface line cards that are supporting an Emergency 911 call to be given automatic priority over cable interface line cards supporting non-emergency voice calls, even in the case of HCCP switchover events. In such cases, Protect HCCP line card interfaces automatically prioritize service to Emergency 911 voice calls, should Working HCCP cable interface line cards be disrupted. This feature is enabled by default, and may not be disabled with manual configuration.

Note Emergency 911 cable interface line card prioritization applies only to PacketCable voice calls.

During HCCP switchover events, cable modems recover in the following sequence:

1. Cable modems supporting Emergency 911 voice traffic
2. Cable modems supporting non-emergency voice traffic
3. Cable modems that are nearing a T4 timeout event, in which service would be disrupted
4. Remaining cable modems

To view information about Emergency 911 voice events and cable interface line card prioritization on the Cisco CMTS router, use the show hccp, show cable calls, and show hccp event-history commands in privileged EXEC mode.

PacketCable Emergency 911 Services Listing and History

The enhanced informational support for PacketCable Emergency 911 calls on the Cisco CMTS router provides the following information and related history:

- active Emergency 911 calls
- recent Emergency 911 calls
- regular voice calls
- voice calls made after recent Emergency 911 calls

This feature is enabled and supported with the following configuration and show commands:

- cable high-priority-call-window
- **show cable calls**
- **show cable modem calls**

To set the call window (in minutes) during which the Cisco CMTS router maintains records of Emergency 911 calls, use the cable high-priority-call-window command in global configuration mode. To remove the call window configuration from the Cisco CMTS router, use the **no** form of this command:

PacketCable Network Components

A PacketCable network contains a number of components. Some components are the same as those that exist in a DOCSIS 1.1 network, while other components are new entities that create the end-to-end infrastructure that the PacketCable network needs to establish calls. Wherever possible, the PacketCable components and protocols build on existing protocols and infrastructures to simplify implementation and interoperability.

- Cable modem—A customer premises equipment (CPE) device that connects to a DOCSIS 1.0 or DOCSIS 1.1 cable network. All DOCSIS cable modems provide high-speed data connectivity to the Internet, while other cable modems can provide additional features, such as telephone connectivity.
- Cable Modem Termination System (CMTS)—A headend-based router that connects a DOCSIS cable network to the IP backbone network. The CMTS controls the DOCSIS 1.1 MAC layer and enforces the quality of service (QoS) limits that the cable operator guarantees to its subscribers. A typical CMTS services between several hundred and several thousand cable modems.

Note See the DOCSIS 1.1 specifications for information about cable modem and CMTS operations.

- Multimedia terminal adapter (MTA)—A CPE device that connects telephones and other end-user devices to the PacketCable network. The PacketCable specification defines two MTA types, an embedded MTA (E-MTA) and a standalone MTA (S-MTA). The E-MTA is an MTA integrated into a DOCSIS 1.1 cable modem, while the S-MTA is a separate MTA that requires a DOCSIS 1.1 cable modem to connect to the cable network.
- Call management server (CMS)—A centrally located server that provides the signaling functions that allow MTAs to establish calls over the network. The CMS uses the Network-based call signaling (NCS) protocol to provide authentication and authorization, call routing, and support for special features such as three-way calling. A PacketCable network could have multiple CMS servers, depending on its size and complexity.

Note The CMS implements several protocols on top of the Common Open Policy Service (COPS) protocol to communicate with the rest of the PacketCable network.

- Gate controller (GC)—A server that controls the establishment of gates in the PacketCable network. A gate is a logical entity in the CMTS that ensures that a service flow is authorized for the QoS features it is requesting. A separate gate controls the upstream and downstream directions of a service flow. When a call is established, the GC instructs the CMTS to create each gate and supplies the set of authorized parameters for each gate, which the CMTS uses to authorize the QoS requests that the MTA is making for the call. The GC is also responsible for coordinating the creation of the two sets of gates at each end of the call so that the call can be authorized and established.

Note A PacketCable network can contain multiple GCs, although only one server at a time is in control of any particular call. Typically, the same workstation provides both the CMS and GC servers.

- Record keeping server (RKS)—Billing server that collects the information about each call as it is made. The RKS uses the Remote Authentication Dial-In User Service (RADIUS) protocol to collect the billing data from the CMTS and other PacketCable servers. The RKS generates a call data record (CDR) for every call and forwards that information to the appropriate application server at the service provider's data processing center for further processing.

Dynamic Quality of Service

A key feature of a PacketCable network is a dynamic quality of service (DQoS) capability that is similar to the dynamic services provided by DOCSIS 1.1. However, DOCSIS 1.1 DQoS authorizes and provisions services only in the cable network and does not reserve the resources needed to propagate a call from one endpoint to another across the network.

The PacketCable DQoS extends the DOCSIS 1.1 services across the entire network, so that resources can be dynamically authorized and provisioned from one endpoint to another. This prevents possible theft-of-service attacks and guarantees customers the services they are authorized to use.

Note PacketCable 1.0 requires that DOCSIS 1.1 be used for resource reservation within the cable network for E-MTA clients.

Two-Stage Resource Reservation Process

The PacketCable DQoS model uses a two-stage resource reservation process, in which resources are first reserved and then committed. This allows a bidirectional reservation process that ensures that resources are available at both endpoints of the connection before actually placing the call.

When an MTA makes a call request, the local CMTS communicates with the gate controller to authorize the call's resources. After the resources are authorized, the CMTS reserves the local resources while it negotiates with the remote end for the resources that are required at that end.

Note The CMTS uses DOCSIS 1.1 Dynamic Service Addition (DSA) messages to reserve the resources, and then uses Dynamic Service Change (DSC) messages to commit the resources.

When all required resources are available, the local CMTS and remote CMTS both commit the resources, allowing traffic to flow. Usage accounting and billing do not begin until the remote MTA picks up and the call is actually in progress.

The DQoS model ensures that both endpoints of a call, as well as the backbone network, have reserved the same bandwidth, and that the bandwidth is reserved only while the call is in progress. When a call terminates, all portions of the network can release the call's resources and make them available for other users.

Making a Call Using DQoS

DOCSIS 1.1 networks use service flows to implement different QoS policies, but service flows exist only within the cable network. To control the service flows and to extend them across the entire network, a PacketCable network creates and maintains "gates."

A gate is a logical entity created on the CMTS at each side of a connection that authorizes and establishes a particular DQoS traffic flow. The CMTS communicates with the gate controller to coordinate the creation of matching gates at each side of the connection.

Gates are unidirectional, so separate gates are required for the downstream and upstream traffic flows. The same gate ID, however, is usually used for the downstream and upstream gates for a call. Each CMTS maintains its own set of gates, so a bidirectional traffic flow requires four gates to be created, two gates on the local CMTS and two gates on the remote CMTS.

For a typical call, gates progress through the following stages to create a DQoS traffic flow:

1. The local MTA makes a call request, and the gate controller sends a Gate-Allocation command to the CMTS, which creates a gate in response and puts it into the Allocated state.
2. The call management server, which might be the same server as the gate controller, parses the call request to translate the destination phone number into the appropriate destination gateway.
3. The gate controller verifies that the MTA making the call request is authorized for the required resources and sends a Gate-Set command to the CMTS, which puts the gate into the Authorized state.
4. The CMTS on each side of the connection reserves the local resources needed for the call, putting the gate into the Reserved state.
5. As the remote CMTS and local CMTS perform gate coordination, their respective gates get put into the Local_Committed and Remote_Committed states.
6. When both sides have reserved all required resources, each CMTS puts its gates into the Committed state, allowing traffic to flow.

DQoSLite Based IPv6 Voice Support

DQoSLite is a modem centric solution without notion of gates, to validate and deliver residential voice services over IPv6 to reclaim IPv4 address space. CMTS does not participate in resource reservation and authorization.

DQoSLite leverages elements from PacketCable 2.0. It is SIP based, it's provision mechanism is similar to PacketCable 2.0 and it can be part of an IP Multimedia Subsystem (IMS) infrastructure for the ISP.

The key factors for deploying IPv6 voice solution on this new DQoSLite infrastructure are as follows:

- It is SIP or IMS based.

- Support for a wide range of multimedia services.
- To reclaim some IPv4 address space.

This feature is enabled and supported with the following configuration and show commands:

- **packetcale authorize vanilla-docsis-mta**
- **show cable modem** { *ip-address* | *mac-address* } **qos**
- **show cable modem** { *ip-address* | *mac-address* } **service-flow**
- **show interface cable** *slot*/*subslot*/*cable-interface-index* **sid** *sid*
- **show interface cable** *slot*/*subslot*/*cable-interface-index* **service-flow** *sfid*

Dynamic Service Transaction ID Support

DOCSIS 2.0 mandates unique Transaction IDs (TAIDs) across transactions. The TAIDs must be unique and not incremented. The TAIDs are assigned by the senders and sometimes the TAID timeout is mismatched between senders and receivers. This affects the uniqueness of the TAID.

A TAID can be reused when the sender finishes a transaction. Similarly, DOCSIS allows the receiver to identify a transaction by TAID without the SFID. Problems arise in DSD transaction and DSA/DSC interrupted transactions, when these two requirements are combined.

The uniqueness of TAID must be ensured to resolve the interoperability issue. This is done by making the CMTS wait until T10 to reuse the same TAID. A new TAID allocation algorithm is used to fulfill this requirement.

It creates a TAID pool to replace the existing 16-bit counter. This TAID pool is monitored by timers to track the TAID expiration. A flag is assigned to each TAID in the pool to indicate its availability. When new TAID is requested, the dynamic service process checks the availability of the TAID. If the TAID is available, it is allocated to the new service flow, else the request is rejected.

Once the TAID is allocated, the timer starts with T10 expiration time and the TAID flag is set to FALSE to indicate the unavailability of TAID. The dynamic service process keeps track of the timer. When the time expires, the timer stops and the flag is set to TRUE to indicate the availability of TAID.

The TAID pool is allocated and initialized at the process initialization. All timers associated with the TAIDs are added as leaf timers to the process' parent timer.

PacketCable Subscriber ID Support

The PacketCable Subscriber ID feature adds a subscriber ID to all Gate Control messages and enhances error codes returned from the Cisco CMTS router.

Previously, the Gate ID was unique only to individual CMTS systems, with the CMTS proxying all CMS Gate Control messaging through a central device, which manages the CMTS connections on behalf of the CMS. The CMS had a single Common Open Policy Service (COPS) association to the proxy device. Therefore, the Gate IDs could be duplicated when using multiple CMTS systems.

A subscriber ID is added to each Gate Control message to disambiguate the Gate IDs between the CMS and proxy device. The subscriber ID parameter is added to the following COPS messages:

- GATE-INFO

- GATE-DELETE
- GATE-OPEN
- GATE-CLOSE

The subscriber ID is available at the CMS and is used in the Gate-Set messages. Additionally, the error codes returned from CMTS or its proxy are enhanced to include more specific information about gate operation failures.

To enable this feature, use the **packetcable gate send-subscriberID** command in global configuration mode.

Benefits

The PacketCable feature offers the following benefits to service providers and their customers:

Integrated Services on a Cable Network

PacketCable allows cable operators the ability to offer multimedia, real-time services, in addition to data connectivity, across their entire network. These services could include basic telephony with lifeline support, as well as telephony that offers competitive extended calling services. Operators can deploy new services while heavily leveraging their existing network infrastructures.

The widespread use of IP as the standard transport mechanism for data networks today is enabling many advanced Internet applications such as multimedia e-mail, real-time chat, streaming media (including music and video), and videoconferencing. The PacketCable initiative provides the network architecture for a cable operator to deliver these services quickly and economically.

Standardized Provisioning

PacketCable provides a standardized, efficient method to provision IP services for individual subscribers, because PacketCable specifications define a uniform, open, and interoperable network. Cable operators are assured of standardized provisioning and the associated lower costs of deployment.

Interoperability

Customer premises equipment (CPE) devices account for a major portion of the capital expense in deploying a VoIP solution at a cable plant. The PacketCable specifications ensure that vendors will build MTA clients that support the voice and other services that cable operators plan to deploy. Because these CPE devices are based on existing DOCSIS-compliant cable modems, time and cost of development is minimized.

Interoperability with the other components of the PacketCable network is also guaranteed because of the standards-based approach to the specifications. Any PacketCable-certified component will be able to interoperate within a network that conforms to the PacketCable standards.

Secure Architecture

Because PacketCable is built upon the security features available in DOCSIS 1.1, cable operators will be assured of networks that are secure from end to end, with a high standard of security that prevents the most common theft-of-service attacks. The comprehensive, standards-based PacketCable specifications are designed to create a network that is as secure as the public switched telephone network (PSTN).

CALEA Support

The PacketCable architecture was designed to accommodate the 1994 Communications Assistance for Law Enforcement Act (CALEA), which requires telecommunications carriers to assist law-enforcement agencies

in conducting court-ordered electronic surveillance. PacketCable networks will be able to provide the two types of information that a carrier must provide, depending on the type of court order:

- Call-identifying information—The carrier must provide the call-identifying information for calls to or from an intercept target. For telephone calls, this information includes the phone numbers called by the target or calling the target.
- Call content—The carrier must provide the content of calls to or from an intercept target. For telephone calls, this real-time content is the voice conversation.

How to Configure PacketCable Operations

This section contains the following tasks to configure the PacketCable feature. Each task is required unless otherwise identified as optional.

Enabling PacketCable Operation

To enable PacketCable operation, use the following commands beginning in user EXEC mode. This is a required procedure.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**packetcable** **Example:** `Router(config)# packetcable`	Enables PacketCable operation on all cable interfaces.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Disabling PacketCable Operation

To disable PacketCable operation, use the following commands beginning in user EXEC mode. This procedure is required only when you no longer want the Cisco CMTS to support PacketCable signaling.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no packetcable** **Example:** `Router(config)# no packetcable`	Disables PacketCable operation on all cable interfaces.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Configuring PacketCable Operation

To configure the different parameters that affect PacketCable operations, use the following commands beginning in user EXEC mode. All of these procedures are optional, because each parameter is set to a default that is appropriate for typical PacketCable operations.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**packetcable element-id** *n* **Example:** `Router(config)# packetcable element-id 23`	Configures the Event Message Element ID for the Cisco CMTS. If you do not manually configure the Element ID, the CMTS defaults to a random value between 0 and 99,999 when PacketCable operations are enabled.

	Command or Action	Purpose
Step 4	**packetcable gate maxcount** *n* **Example:** `Router(config)# packetcable gate maxcount 524288`	Sets the maximum number of gate IDs to be allocated in the gate database on the Cisco CMTS.
Step 5	**packetcable timer T0** *timer-value* **Example:** `Router(config)# packetcable timer T0 40000`	Sets the T0 timer in milliseconds.
Step 6	**packetcable timer T1** *timer-value* **Example:** `Router(config)# packetcable timer T1 300000`	Sets the T1 timer in milliseconds.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Enabling Both PacketCable and Non-PacketCable UGS Service Flows

By default, when PacketCable operations are enabled using the **packetcable** command, cable modems must follow the PacketCable protocol when requesting Unsolicited Grant Service (UGS) service flows. This prevents DOCSIS cable modems that do not support PacketCable operations from using DOCSIS-style UGS service flows.

If you have a mixed network that contains both PacketCable and non-PacketCable DOCSIS CMs, you can use the **packetcable authorize vanilla-docsis-mta** command to enable both types of UGS service flows. This is an optional procedure.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**packetcable** **Example:** `Router(config)# packetcable`	Enables PacketCable operations.
Step 4	**packetcable authorize vanilla-docsis-mta** **Example:** `Router(config)# packetcable authorize vanilla-docsis-mta`	Enables the use of DOCSIS-style UGS service flow requests.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

What to do next

Tip Use the **show packetcable global** command to display whether non-PacketCable UGS service flows have been enabled.

Enabling PacketCable Subscriber ID Support

To include subscriber identification in GATE-OPEN and GATE-CLOSE Gate Control messages, use the **packetcable gate send-subscriberID** command in global configuration mode.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**packetcable** **Example:** `Router(config)# packetcable`	Enables PacketCable operations.

	Command or Action	Purpose
Step 4	**packetcable gate send-subscribeID** **Example:** `Router(config)# packetcable gate send-subscriberID`	Enables the use of gate control subscriber identification information.
Step 5	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Configuring RADIUS Accounting for RKS Servers

To enable the Cisco CMTS router to communicate with the Record Keeping Servers (RKS servers) using the RADIUS protocol, use the following commands. This is a required procedure.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the authentication, authorization, and accounting (AAA) access control model.
Step 4	**aaa group server radius** *group-name* **Example:** `Router(config)# aaa group server radius packetcable`	Creates a group of RADIUS servers for authentication and enters RADIUS group configuration mode. The value of *group-name* is a unique, arbitrary string that identifies this group.
Step 5	**server** {*hostname* | *ip-address*} [**auth-port** *udp-port*] [**acct-port** *udp-port*] **Example:**	Specifies the host name or IP address for the RADIUS server that is providing the RKS services.

	Command or Action	Purpose
	`Router(config-sg-radius)# server radius-server1`	**Note** Repeat this command as needed to enter multiple RADIUS servers. The Cisco CMTS uses the servers in the order given with this command.
Step 6	**exit** **Example:** `Router(config-sg-radius)# exit`	Exits RADIUS group configuration mode.
Step 7	**aaa accounting network default start-stop group radius group** *group-name* **Example:** `Router(config)# aaa accounting network default start-stop group radius group packetcable`	Enables AAA services using the group of RADIUS servers that are defined in the previously created group. The *group-name* parameter should be the same name specified in Step 4 .
Step 8	**radius-server host** {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**timeout** *seconds*] [**retransmit** *retries*] **key 0000000000000000** **Example:** `Router(config)# radius-server host radius-server1 key 0000000000000000`	Specifies a RADIUS host. Use the same values for *hostname* or *ip-address* as for one of the servers specified in Step 5 . If you also specified the **auth-port** or **acct-port** values in Step 5 , you must also specify those here, as well. The **key** value is required and must be 16 ASCII zeros, as shown. **Note** Repeat this command for each RADIUS server entered in Step 5 .
Step 9	**radius-server vsa send accounting** **Example:** `Router(config)# radius-server vsa send accounting`	Configures the Cisco CMTS to recognize and use accounting-related vendor-specific attributes (VSA).
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

What to do next

Troubleshooting Tips

If the connection between a PacketCable CMS and the Cisco CMTS router is not completely established, and the PacketCable CMS does not correctly terminate the session by sending a TCP FIN message, the connection shows a COPS server in the output of the **show cops server** command.

PacketCable Client Accept Timeout

The PacketCable Client Accept Timeout feature supports COPS for PacketCable on the Cisco CMTS router. This feature also allows you to set timeout values for COPS Telnet connections on the Cisco CMTS router, and for clearing COPS Telnet sessions.

Telnet errors on the network or Cisco CMTS router might cause incomplete COPS sessions to be created. In order to address this issue, the timeout timer enables clearing and cleaning of allocated resources for the stale COPS Telnet sessions on the Cisco CMTS router.

The timeout timer applies to each COPS Telnet connection on the Cisco CMTS router. When this timeout setting expires, it terminates the Telnet session and clears supporting resources on the Cisco CMTS router.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**packetcable timer** {**T0** *timer-value* | **T1** *timer-value* | **multimedia T1** *timer-value*}
	Example:
	`Router(config)# packetcable timer T0 300000`
	Example:
	`Router(config)# packetcable timer T1 400000`
	Example:
	`Router(config)# packetcable timer multimedia T1 400000`
		Sets the PacketCable timer value.

Step 4	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

What to do next

Troubleshooting Tips

If the connection between a PacketCable CMS and the Cisco CMTS router is not completely established, and the PacketCable CMS does not correctly terminate the session by sending a TCP FIN message, the connection shows a COPS server in the output of the **show cops server** command.

Configuration Examples for PacketCable

This section provides a PacketCable configuration example.

Example: Typical PacketCable Configuration

This section provides a typical configuration for a Cisco CMTS router that has been configured for PacketCable operations, using default parameters. To use this configuration, you must change the IP addresses for the RADIUS and RKS servers to match the addresses for the servers in your network.

```
!
version 15.5
no parser cache
no service pad
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
no service password-encryption
service internal
service udp-small-servers max-servers no-limit
service tcp-small-servers max-servers no-limit
!
hostname Router
!
no logging rate-limit
aaa new-model
!
!
aaa group server radius a
 server 10.9.62.12 auth-port 1813 acct-port 1812
 server 10.9.62.13 auth-port 1813 acct-port 1812
!
aaa accounting network default start-stop group radius group a
aaa session-id common
enable password <delete>
!
cable modulation-profile 2 request 0 16 0 8 qpsk scrambler 152 no-diff 64 fixed uw16
cable modulation-profile 2 initial 5 34 0 48 qpsk scrambler 152 no-diff 128 fixed uw16
cable modulation-profile 2 station 5 34 0 48 qpsk scrambler 152 no-diff 128 fixed uw16
cable modulation-profile 2 short 6 75 6 8 16qam scrambler 152 no-diff 144 shortened uw8
cable modulation-profile 2 long 8 220 0 8 16qam scrambler 152 no-diff 160 shortened uw8
cable modulation-profile 5 request 0 16 2 8 qpsk scrambler 152 no-diff 64 fixed uw16
cable modulation-profile 5 initial 5 34 0 48 qpsk scrambler 152 no-diff 128 fixed uw16
cable modulation-profile 5 station 5 34 0 48 qpsk scrambler 152 no-diff 128 fixed uw16
cable modulation-profile 5 short 6 78 7 8 16qam scrambler 152 no-diff 144 shortened uw16
cable modulation-profile 5 long 8 220 0 8 16qam scrambler 152 no-diff 160 shortened uw16
cable qos profile 5 max-burst 1200
cable qos profile 5 max-downstream 2000
cable qos profile 5 max-upstream 128
cable qos profile 5 priority 5
cable qos profile 5 privacy
cable qos profile 7 guaranteed-upstream 87
cable qos profile 7 max-upstream 87
cable qos profile 7 privacy
no cable qos permission create
```

```
no cable qos permission update
cable qos permission modems
cable qos permission enforce 5
cable time-server
no cable privacy accept-self-signed-certificate
ip subnet-zero
!
!
no ip domain-lookup
ip domain-name cisco.com
ip host tftp 10.8.8.8
ip host cnr 10.9.62.17
!
packetcable
packetcable element-id 12456
!
!
!
interface Tunnel0
 ip address 10.55.66.3 255.255.255.0
 load-interval 30
 tunnel source TenGigabitEthernet 4/1/0
 tunnel destination 172.27.184.69
!
interface Tunnel10
 ip address 10.0.1.1 255.255.0.0
!
interface TenGigabitEthernet 4/1/0
 ip address 10.9.60.10 255.255.0.0
 no ip redirects
 no ip mroute-cache
 full-duplex
!
interface TenGigabitEthernet 4/1/0
 ip address 172.22.79.44 255.255.254.0
 no ip redirects
 no ip mroute-cache
 full-duplex
!
interface Cable3/0
 ip address 10.3.1.33 255.255.255.0 secondary
 ip address 10.4.1.1 255.255.255.0 secondary
 ip address 10.4.1.33 255.255.255.0 secondary
 ip address 10.3.1.1 255.255.255.0
 ip helper-address 10.9.62.17
 load-interval 30
 no keepalive
 cable downstream annex B
 cable downstream modulation 64qam
 cable downstream interleave-depth 32
 cable downstream frequency 55500000
 cable upstream 0 modulation-profile 2
 no cable upstream 0 shutdown
 cable upstream 1 frequency 12000000
 cable upstream 1 power-level 0
 cable upstream 1 channel-width 3200000
 cable upstream 1 data-backoff automatic
 cable upstream 1 modulation-profile 2
 cable upstream 1 shutdown
 cable upstream 2 frequency 16000000
 cable upstream 2 power-level 0
 cable upstream 2 channel-width 3200000
 cable upstream 2 data-backoff automatic
 cable upstream 2 modulation-profile 2
```

```
 no cable upstream 2 shutdown
 cable upstream 3 frequency 20000000
 cable upstream 3 power-level 0
 cable upstream 3 channel-width 3200000
 cable upstream 3 data-backoff automatic
 cable upstream 3 modulation-profile 2
 no cable upstream 3 shutdown
 cable upstream 4 frequency 24000000
 cable upstream 4 power-level 0
 cable upstream 4 channel-width 3200000
 cable upstream 4 data-backoff automatic
 no cable upstream 4 shutdown
 cable upstream 5 frequency 28000000
 cable upstream 5 power-level 0
 cable upstream 5 channel-width 3200000
 cable upstream 5 data-backoff automatic
 cable upstream 5 modulation-profile 2
 no cable upstream 5 shutdown
 cable dhcp-giaddr policy
!
router eigrp 48849
 network 1.0.0.0
 network 10.0.0.0
 auto-summary
 no eigrp log-neighbor-changes
!
ip default-gateway 10.9.0.1
ip classless
ip route 0.0.0.0 0.0.0.0 172.22.78.1
ip route 10.8.0.0 255.255.0.0 10.9.0.1
ip route 192.168.80.0 255.255.255.0 Tunnel0
ip route 192.168.80.0 255.255.255.0 172.27.184.69
ip route 10.255.254.254 255.255.255.255 10.9.0.1
no ip http server
ip pim bidir-enable
!
!
cdp run
!
!
radius-server host 10.9.62.12 auth-port 1813 acct-port 1812 key 0000000000000000
radius-server retransmit 3
radius-server vsa send accounting
!
line con 0
 exec-timeout 0 0
 privilege level 15
line aux 0
line vty 0 4
 session-timeout 33
 exec-timeout 0 0
 password <deleted>
!
ntp clock-period 17179976
ntp server 1.9.35.8
end
```

Verifying PacketCable Operations

To verify and maintain information about PacketCable operations, use one or more of the following commands:

- **show packetcable global**

- **show packetcable gate**
- **show packetcable gate ipv6**
- **show packetcable gate dqos**
- **show packetcable gate counter commit**

To verify the PacketCable configuration, values for the Element ID, maximum number of gates, and the different CMTS-based DQoS timers, use the **show packetcable global** command in privileged EXEC mode.

```
Router# show packetcable global
Packet Cable Global configuration:
Enabled   : Yes
Element-ID: 12456
Max Gates : 1048576
Allow non-PacketCable UGS
Default Timer value -
  T0      : 30000 msec
  T1      : 300000 msec
```

To verify information about one or more gates in the gate database, use the **show packetcable gate** command as shown in the following example:

```
Router# show packetcable gate summary
GateID     i/f          SubscriberID   GC-Addr        State     Type  SFID(us) SFID(ds)
13582      Ca8/1/0      3.18.1.4       20.5.0.254     RECOVERY  Dqos  74
29962      Ca8/1/0      3.18.1.5       20.5.0.254     RECOVERY  Dqos  73
46354      Ca8/1/0      ------         20.5.0.254     RECOVERY  Dqos  72
62738      Ca8/1/0      ------         20.5.0.254     RECOVERY  Dqos          69
Total number of  gates = 4
Total Gates committed(since bootup or clear counter) = 8
```

To verify information about one or more PacketCable gates associated with IPv6 subscriber IDs in the gate database, use the **show packetcable gate ipv6** command as shown in the following example:

```
Router# show packetcable gate ipv6 summary
GateID     i/f                    SubscriberID               State   SFID(us) SFID(ds)
13582      Ca8/1/0      2001:40:1:42:C0B4:84E5:5081:9B5C      COMMIT    74
29962      Ca8/1/0      2001:40:1:42:C0B4:84E5:5081:9B5C      COMMIT    73
46354      Ca8/1/0      2001:40:1:42:C0B4:84E5:5081:9B5C      COMMIT    72
62738      Ca8/1/0      2001:40:1:42:C0B4:84E5:5081:9B5C      COMMIT             69

Total number of  gates = 4
Total Gates committed(since bootup or clear counter) = 8
```

To verify information about one or more PacketCable gates associated with IPv4 subscriber IDs in the gate database, use the **show packetcable gate dqos** command as shown in the following example:

```
Router# show packetcable gate dqos summary
GateID     i/f          SubscriberID   GC-Addr        State     Type  SFID(us) SFID(ds)
13576      Ca8/1/0      40.1.43.60     10.74.58.5     COMMIT    DQoS  527      528
29956      Ca8/1/0      40.1.43.56     10.74.58.5     COMMIT    DQoS  525      526
Total number of DQOS gates = 2
Total Gates committed(since bootup or clear counter) = 346
```

To verify the total number of gates that the Cisco CMTS router has moved to the Committed state since the router was last reset, or since the counter was last cleared, use the show packetcable gate counter commit command as shown in the following example:

```
Router# show packetcable gate counter commit
```

Total Gates committed (since bootup or clear counter) = 132

Verifying Emergency 911 Calls

This section provides a few examples to illustrate how you can use the show cable calls and show cable modem calls commands to verify different scenarios associated with Emergency 911 calls.

The following example displays Emergency 911 calls made on the Cable8/1/1 interface on the Cisco CMTS router during the window set for high priority calls:

```
Router# show cable calls
Interface   ActiveHiPriCalls  ActiveAllCalls  PostHiPriCallCMs  RecentHiPriCMs
C5/0/0      0                 0               0                 0
C5/0/1      0                 0               0                 0
C5/1/0      0                 0               0                 0
C5/1/1      0                 0               0                 0
C5/1/2      0                 0               0                 0
C5/1/3      0                 0               0                 0
C5/1/4      0                 0               0                 0
C6/0/0      0                 0               0                 0
C6/0/1      0                 0               0                 0
C7/0/0      0                 0               0                 0
C7/0/1      0                 0               0                 0
C8/1/0      0                 0               0                 0
C8/1/1      1                 1               0                 0
C8/1/2      0                 0               0                 0
C8/1/3      0                 0               0                 0
C8/1/4      0                 0               0                 0
Total       1                 1               0                 0
```

The following example displays the change on the Cisco CMTS router when this Emergency 911 calls ends:

```
Router# show cable calls
Interface   ActiveHiPriCalls  ActiveAllCalls  PostHiPriCallCMs  RecentHiPriCMs
C5/0/0      0                 0               0                 0
C5/0/1      0                 0               0                 0
C5/1/0      0                 0               0                 0
C5/1/1      0                 0               0                 0
C5/1/2      0                 0               0                 0
C5/1/3      0                 0               0                 0
C5/1/4      0                 0               0                 0
C6/0/0      0                 0               0                 0
C6/0/1      0                 0               0                 0
C7/0/0      0                 0               0                 0
C7/0/1      0                 0               0                 0
C8/1/0      0                 0               0                 0
C8/1/1      0                 0               0                 1
C8/1/2      0                 0               0                 0
C8/1/3      0                 0               0                 0
C8/1/4      0                 0               0                 0
Total       0                 0               0                 1
```

The following example displays information that is available when making a voice call from the same MTA to another MTA on the same interface:

```
Router# show cable calls
Interface   ActiveHiPriCalls  ActiveAllCalls  PostHiPriCallCMs  RecentHiPriCMs
C5/0/0      0                 0               0                 0
C5/0/1      0                 0               0                 0
C5/1/0      0                 0               0                 0
```

```
C5/1/1      0                0                0                0
C5/1/2      0                0                0                0
C5/1/3      0                0                0                0
C5/1/4      0                0                0                0
C6/0/0      0                0                0                0
C6/0/1      0                0                0                0
C7/0/0      0                0                0                0
C7/0/1      0                0                0                0
C8/1/0      0                0                0                0
C8/1/1      0                2                1                1
C8/1/2      0                0                0                0
C8/1/3      0                0                0                0
C8/1/4      0                0                0                0
Total       0                2                1                1
```

The following example displays information that is available when a voice call from the same MTA to another MTA on the same interface ends:

```
Router# show cable calls
Interface   ActiveHiPriCalls  ActiveAllCalls  PostHiPriCallCMs  RecentHiPriCMs
C5/0/0      0                 0               0                 0
C5/0/1      0                 0               0                 0
C5/1/0      0                 0               0                 0
C5/1/1      0                 0               0                 0
C5/1/2      0                 0               0                 0
C5/1/3      0                 0               0                 0
C5/1/4      0                 0               0                 0
C6/0/0      0                 0               0                 0
C6/0/1      0                 0               0                 0
C7/0/0      0                 0               0                 0
C7/0/1      0                 0               0                 0
C8/1/0      0                 0               0                 0
C8/1/1      0                 0               0                 1
C8/1/2      0                 0               0                 0
C8/1/3      0                 0               0                 0
C8/1/4      0                 0               0                 0
Total       0                 0               0                 1
```

The following examples display the show cable modem calls command output on the Cisco CMTS router over a period of time, with changing call status information. The call information disappears when a call ends.

```
Router# show cable modem calls
Cable Modem Call Status Flags:
H: Active high priority calls
R: Recent high priority calls
V: Active voice calls (including high priority)
MAC Address    IP Address      I/F       Prim  CMCallStatus  LatestHiPriCall
                                         Sid                    (min:sec)
0000.cab7.7b04 10.10.155.38    C8/1/1/U0 18    R                   0:39
Router# show cable modem calls
Cable Modem Call Status Flags:
H: Active high priority calls
R: Recent high priority calls
V: Active voice calls (including high priority)
MAC Address    IP Address      I/F       Prim  CMCallStatus  LatestHiPriCall
                                         Sid                    (min:sec)
```

The following example displays a new Emergency 911 call on the Cisco CMTS router:

```
Router# show cable modem calls
Cable Modem Call Status Flags:
H: Active high priority calls
```

```
R: Recent high priority calls
V: Active voice calls (including high priority)
MAC Address    IP Address      I/F       Prim  CMCallStatus  LatestHiPriCall
                                         Sid                    (min:sec)
0000.cab7.7b04 10.10.155.38    C8/1/1/U0 18    HV                1:30
```

The following example displays the end of the Emergency 911 call on the Cisco CMTS router:

```
Router# show cable modem calls
Cable Modem Call Status Flags:
H: Active high priority calls
R: Recent high priority calls
V: Active voice calls (including high priority)
MAC Address    IP Address      I/F       Prim  CMCallStatus  LatestHiPriCall
                                         Sid                    (min:sec)
0000.cab7.7b04 10.10.155.38    C8/1/1/U0 18    R                 0:3
```

The following example displays a non-emergency voice call on the Cisco CMTS router from the same MTA:

```
Router# show cable modem calls
Cable Modem Call Status Flags:
H: Active high priority calls
R: Recent high priority calls
V: Active voice calls (including high priority)
MAC Address    IP Address      I/F       Prim  CMCallStatus  LatestHiPriCall
                                         Sid                    (min:sec)
0000.ca36.f97d 10.10.155.25    C8/1/1/U0 5     V                  -
0000.cab7.7b04 10.10.155.38    C8/1/1/U0 18    RV                0:30
```

The following example displays the end of the non-emergency voice call on the Cisco CMTS router:

```
Router# show cable modem calls
Cable Modem Call Status Flags:
H: Active high priority calls
R: Recent high priority calls
V: Active voice calls (including high priority)
MAC Address    IP Address      I/F       Prim  CMCallStatus  LatestHiPriCall
                                         Sid                    (min:sec)
0000.cab7.7b04 10.10.155.38    C8/1/1/U0 18    R                 0:36
```

Information About PacketCable Multimedia Operations

The PacketCable Multimedia (PCMM) feature is a powerful implementation of the CableLabs® standards for PacketCable Multimedia. PCMM provides enhanced QoS for multimedia applications, voice, and bandwidth-intensive services over a DOCSIS (DOCSIS 1.1 and later versions) network.

The Cisco CMTS router supports DOCSIS QoS for SIP-based telephones and SIP video phones, Bandwidth-on-Demand applications, and network-based gaming applications, all of which place extensive bandwidth demands on the network.

This section provides information about the following aspects of PacketCable Multimedia for the Cisco CMTS router, emphasizing PCMM components that are configured with the Cisco IOS command-line interface later in this document:

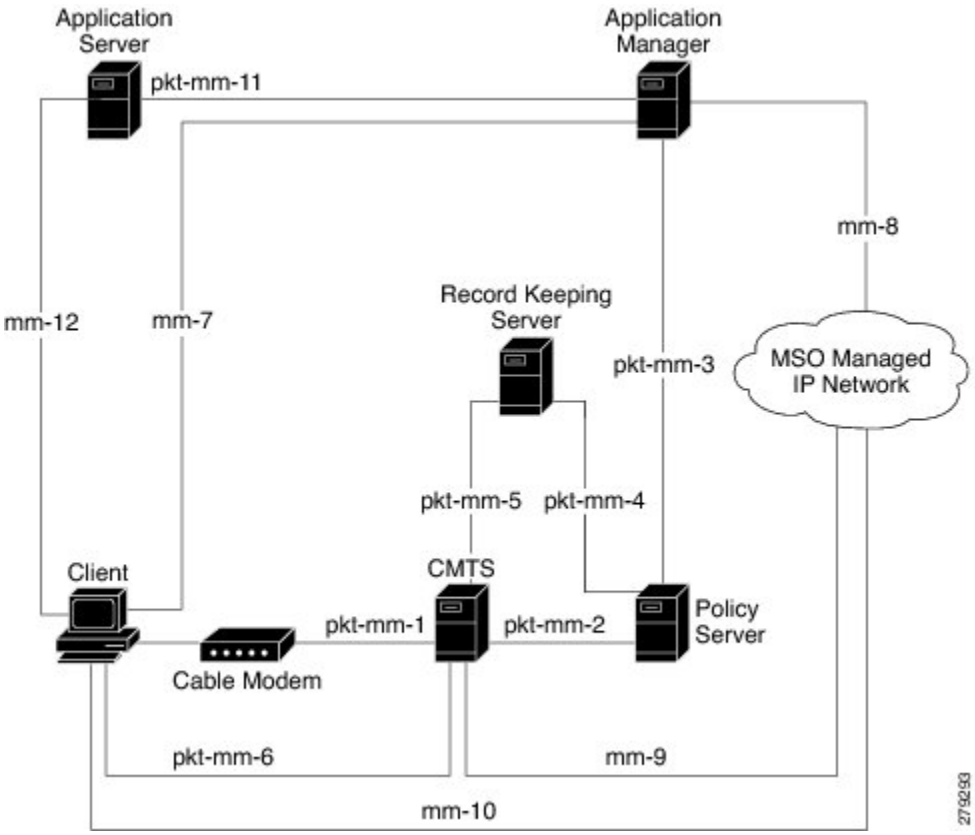
PCMM Overview

The following network components are required to support the PCMM feature:

- Application Server—Responsible for relaying client requests to the Application Manager.
- Application Manager—Responsible for application or session-level state and for applying session control domain (SCD) policy.
- Policy Server—Responsible for applying the RCD policy and for managing relationships between the Application Manager and a Cisco CMTS router.
- Cisco CMTS router—Responsible for performing admission control and managing network resources through DOCSIS service flows.

Figure below provides an architectural overview of the PCMM functionality:

Figure 31: PCMM Architectural Overview



PCMM Enhancements over PacketCable 1.x

PacketCable Multimedia is a service delivery framework that leverages and uses as much of existing PacketCable 1.x deployments and functionality as possible. Furthermore, PCMM offers powerful enhancements to the VoIP service delivery framework with straightforward CLI implementation. The key enhancements that the PCMM provides are:

- Time and volume based network resource authorizations are based on DOCSIS 1.1 Quality of Service (QoS) mechanisms.
- Event-based network resource auditing and management functions.
- Secure infrastructure that protects all interfaces at appropriate levels.

- Preauthorized model from PacketCable 1.x, where the PCMM gate installation and management is supplemented with service flow creation, modification and deletion functions. Together, these provide a secure, network-based QoS.

PCMM and High Availability Features on the Cisco CMTS Router

High Availability on the Cisco CMTS router accommodates synchronization of service flows created for the PCMM applications and the PCCM gate.

PCMM Gates

PCMM Gate Overview and PCMM Dynamic Quality of Service

A PacketCable 1.x gate defines QoS parameters and policy-based authorization for subscribers, and a specific envelope of network resources. A PacketCable 1.x gate also maintains classifiers for originating and terminating IP addresses and ports.

The subscriber ID can identify both IPv4 and IPv6 addresses of either the cable modem or the client CPE.

PacketCable 1.x defines a preauthorization model. The PacketCable gates are created and installed at the Cisco CMTS router prior to network resource reservation or activation requests. This process, termed gate control, is managed through a COPS-based policy interface on the Cisco CMTS router.

In PCMM, this COPS-based interface is enhanced for QoS life-cycle management. PCMM gates maintain service flow creation, modification and deletion functions to provide for network-based QoS. Multiple PCMM gates and service flow policies can be maintained on the Cisco CMTS router at a given time, and these PCMM gates are fully interoperable with PacketCable 1.x gates.

When a cable modem subscriber requests bandwidth for a network-intensive application, the network Policy Server sends a Gate-Set message to the Cisco CMTS router. This message contains QoS, service flow, and billing information for this subscriber. This gate profile information is maintained on the Cisco CMTS router, to include PCMM gate states and PCMM state transitions.

The Cisco CMTS router initiates service flows with cable modems, and optimizes DOCSIS resource availability on the Cisco CMTS router for bandwidth-intensive service flows characteristic to PCMM.

Restrictions

On some upstream paths, best effort service flows are configured on some modems with Committed Information Rate (CIR). When a number of bandwidth requests are queued in the modems, only a few requests are sent to the CMTS. This occurs due to congestion of sending requests caused by higher number of service flows, greater traffic and small size of packets. Therefore, only a few best effort service flow requests are satisfied by the CMTS.

PCMM Persistent Gate

Persistent Gate is a feature by which PCMM gate information is maintained for cable modems that go offline. This gate information is quickly enabled after a cable modem returns online. When a cable modem returns online, the Cisco CMTS router scans PCMM gates previously stored, and initiates service to the cable modem according to the respective PCMM gate. This re-enabled service maintains traffic support profiles for that gate, and allocates DOCSIS resources based on the new online subscriber.

PCMM Interfaces

PCMM optimizes the IPC handshake between the cable interface line card and the Route Processor (RP) for the Cisco CMTS router. Additional PCMM interface changes from PacketCable 1.x include the handling for COPS interface and distributed cable interface line cards.

PCMM to COPS Interface

PCMM differs from PacketCable 1.x in handling COPS sessions. The COPS sessions on PCMM use TCP port number 3918 by default. Whereas, PacketCable uses the DQoS specification for TCP port requirements and COPS sessions.

When the PCMM module initializes for the first time, a PCMM registry is added to the cable interface line card and the route processor. The PCMM module also registers the PCMM COPS client with the COPS layer on the Cisco CMTS router.

PCMM and Distributed Cable Interface Line Cards

As with PacketCable 1.x, PCMM uses IPC messages for voice support. When PCMM gates are created on the Network Processing Engine (NPE) or route processor (RP), the PCMM gate parameters are sent to cable interface line cards. IPC maintains all communication between the NPE or RP, and the cable interface line cards.

Event messaging is used with PCMM to support billing information based on Gate-Set messages. Event messaging for distributed cable interface line cards originates from the line cards, based on the success of DSX operation.

The PCMM module also registers the PCMM COPS client with the COPS layer.

PCMM Unicast and Multicast

In unicast transmission, content is sent to a unique user. In multicast transmission, content is sent to multiple users simultaneously.

PCMM Multicast Session Range

You can configure a PCMM multicast session range by specifying IPv4 IP addresses and a mask for a PCMM multicast group. The PCMM multicast session range enables the Cisco CMTS router to accept Gate-Set messages from the PCMM Policy Server. If a PCMM multicast session range is configured, the Cisco CMTS router does not allow you to create multicast sessions using other sources such as Internet Group Management Protocol (IGMP) and DOCSIS Set-Top Gateway (DSG).

How to Configure PCMM Operations

The following tasks describe how to enable PCMM operations and configure its related features on the Cisco CMTS router:

Enabling PCMM Operations on the Cisco CMTS Router

To enable PCMM operations on the Cisco CMTS router:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**packetcable multimedia** **Example:** `Router(config)# packetcable multimedia`	Enables and displays PCMM processing on the Cisco CMTS router. This command enables the Cisco CMTS router to start or stop responding to PCMM COPS messages received from the PCMM Policy Server.
Step 4	**packetcable authorize vanilla-docsis-mta** **Example:** `Router(config)# packetcable authorize vanilla-docsis-mta`	Allows non-DQoS MTAs to send DOCSIS DSX messages.
Step 5	**packetcable gate maxcount** *n* **Example:** `Router(config)# packetcable gate maxcount 890`	Sets the maximum number of PCMM gates in the gate database.
Step 6	**packetcable timer multimedia T1** *timer-value* **Example:** `Router(config)# packetcable timer multimedia T1 300000`	Sets the timeout value for T1 timer used in PCMM gate processing.
Step 7	**clear packetcable gate counter commit [dqos | multimedia]** **Example:** `Router(config)# clear packetcable gate counter commit multimedia`	(Optional) Clears the specified PCMM gate counter.
Step 8	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Configuring a PCMM Multicast Session Range

A PCMM multicast session range enables the Cisco CMTS router to use a range of IP addresses for a PCMM multicast group.

Before you begin

Ensure that PCMM is configured using the **packetcable multimedia command.**

Note

- You can configure only one PCMM multicast group on the Cisco CMTS router. You can configure a maximum of ten multicast sessions for a single multicast group.
- The PCMM multicast feature is supported only with the cable modems that are capable of Multicast DSID-based Forwarding (MDF).

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable multicast source pcmm** **Example:** `Router(config)# cable multicast source pcmm`	Enables PCMM-based multicast service on the Cisco CMTS router and enters multicast session range configuration mode.
Step 4	**session-range** *ip-addressip-mask* **Example:** `Router(config)# session-range 229.0.0.0 255.0.0.0`	Configures a session range for the PCMM multicast group.
Step 5	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Configuration Examples for PacketCable Multimedia

The following sections provide configuration examples for PCMM operations on the Cisco CMTS router:

Example: Enabling PCMM Operations on the Cisco CMTS Router

```
Router# configure terminal
Router(config)# packetcable multimedia
Router(config)# packetcable authorize vanilla-docsis-mta
Router(config)# packetcable gate maxcount 890
Router(config)# packetcable timer multimedia 30000
```

Example: Enabling a Multicast Session Range on the Cisco CMTS Router

```
Router# configure terminal
Router(config)# cable multicast source pcmm
Router(config)# session-range 229.0.0.0 255.0.0.0
```

Verifying PCMM Operations

Use the following **show** commands to verify PCMM operations:

- **show packetcable gate multimedia**
- **show cable multicast db**
- **show interface wideband-cable**
- **show cable multicast qos**

To verify the PCMM multicast gates, use the **show packetcable gate multimedia** command as shown in the following example:

```
Router# show packetcable gate multimedia multicast summary
GateID     i/f          SubscriberID   GC-Addr        State     Type  SFID(us) SFID(ds)
134        Ca5/0/0      60.1.1.202     2.39.26.19     COMMIT    MM             4
Total number of Multimedia-MCAST gates = 1
Total Gates committed(since bootup or clear counter) = 1
```

To verify the PCMM IPv6 gates, use the **show packetcable gate multimedia ipv6**command as shown in the following example:

```
Router# show packetcable gate multimedia ipv6 summary
Load for five secs: 10%/1%; one minute: 9%; five minutes: 9%
Time source is NTP, 03:29:42.153 EST Mon Nov 9 2015

GateID     i/f                     SubscriberID                State   SFID(us) SFID(ds)
409        Ca5/0/2      2001:420:2C7F:FC38:58AF:E36A:80:213A   COMMIT             1326

16789      Ca5/0/2      2001:420:2C7F:FC38:AC40:A49A:F80A:8D0B COMMIT    1321
33177      Ca5/0/2      2001:420:2C7F:FC38:DD49:72A3:2ECC:8770 COMMIT             1322
```

```
49577      Ca5/0/2      2001:420:2C7F:FC38:485:31DF:C88B:E315  COMMIT    1308
65953      Ca5/0/2      2001:420:2C7F:FC38:5AB:AA0B:34AD:ACCF  COMMIT    1336
82337      Ca5/0/2      2001:420:2C7F:FC38:5AB:AA0B:34AD:ACCF  COMMIT              1337

98721      Ca5/0/2      2001:420:2C7F:FC38:5570:EF2E:7565:D36A COMMIT              1316

115097     Ca5/0/2      2001:420:2C7F:FC38:6009:EF26:F573:7356 COMMIT              1318

131489     Ca5/0/2      2001:420:2C7F:FC38:7D4A:BC50:3FD:CA7   COMMIT    1312
147873     Ca5/0/2      2001:420:2C7F:FC38:E83E:8259:AEF6:5624 COMMIT    1332


Total number of Multimedia gates = 10
Total Gates committed(since bootup or clear counter) = 1024
```

To verify all the PCMM client entries available with the multicast database, use the **show cable multicast db** command as shown in the following example:

```
Router# show cable multicast db client pcmm
Interface : Bundle1
Session (S,G) : (*,229.2.2.12)
Fwd Intf     Bundle Intf  Host Intf    CM MAC         CPE IP           Gate-ID SFID
Wi1/1/0:0    Bundle1      Ca5/0/0      0018.6852.8056 60.1.1.202       134 4
```

To verify multicast sessions on a specific wideband cable interface, use the **show interface wideband-cable** command as shown in the following example:

```
Router# show interface wideband-cable 1/1/0:0 multicast-sessions
Default Multicast Service Flow 3 on Wideband-Cable1/1/0:0
Multicast Group    : 229.2.2.12
         Source    : N/A
         Act GCRs  : 1
         Interface : Bu1                   State: A      GI: Bu1         RC: 0
  GCR       : GC   SAID    SFID    Key   GQC    GEn
                    512  8196    4       0     512    0
```

To verify the attribute-based assignment of service flows on a specific wideband cable interface, use the **show interface wideband-cable** command as shown in the following example:

```
Router# show interface wideband-cable 1/1/0:0
service-flow 4 verbose
Sfid                                  : 4
Mac Address                           : ffff.ffff.ffff
Type                                  : Secondary(Static)
Direction                             : Downstream
Current State                         : Active
Current QoS Indexes [Prov, Adm, Act]  : [4, 4, 4]
Active Time                           : 05:26
Required Attributes                   : 0x00000000
Forbidden Attributes                  : 0x00000000
Aggregate Attributes                  : 0x00000000
Multicast Sid                         : 8196
Traffic Priority                      : 0
Maximum Sustained rate                : 0 bits/sec
Maximum Burst                         : 3044 bytes
Minimum Reserved Rate                 : 250000 bits/sec
Minimum Packet Size                   : 0 bytes
Maximum Latency                       : 0 usecs
```

```
Peak Rate                                : 0 bits/sec
Admitted QoS Timeout                     : 200 seconds
Active QoS Timeout                       : 0 seconds
Packets                                  : 0
Bytes                                    : 0
Rate Limit Delayed Packets               : 0
Rate Limit Dropped Packets               : 0
Current Throughput                       : 0 bits/sec, 0 packets/sec
Application Priority                     : 0
Low Latency App                          : No
Blaze/JIB3 DS Statistic Index            : 0
Forwarding Interface                     : Wi1/1/0:0
Classifiers: NONE
```

To verify that the PCMM-based MQoS gate controllers are created using the correct session ranges, use the **show cable multicast qos**command as shown in the following example:

```
Router# show cable multicast qos group-qos
Group QOS Index    Service Class    Control    Igmp Limit    Override  App
 DEFAULT           mcast_default     Aggregate  NO-LIMIT
 1                 SDV_SD            Single      ---           No        CLI
 512               SDV_HD            Single      ---           No        PCMM
```

High Availability Stateful Switchover (SSO) for PacketCable and PacketCable MultiMedia

Enhanced high availability support enables the synchronization of PacketCable and PacketCable MultiMedia (PCMM) gates during switchover events on the Cisco CMTS router. This enhancement is enabled by default.

This functionality uses the existing per-interface HCCP commands that are used to associate the working and protect interfaces in the case of N+1 redundancy.

PacketCable and PCMM with Admission Control

A PacketCable or PacketCable Multimedia network contains a number of components that benefit from Admission Control QoS. Admission Control manages and optimizes QoS for PacketCable and PCMM in these ways:

- QoS (based on DOCSIS 1.1 or later versions) for voice and data
- Cable modem registration
- CMS
- Gateway controllers (GC)
- Record keeping servers (RKS)
- Video telephony

When configuring Admission Control with either PacketCable or PCMM, PacketCable or PCMM must be fully operational on the Cisco CMTS headend prior to gaining the benefits from Admission Control.

Voice MGPI Support

The multiple grants per interval (MGPI) feature enables the Cisco CMTS router to map multiple PacketCable Multimedia gates (application flows) to a single DOCSIS service flow using UGS traffic profiles of the same cable modem. In other words, the Cisco CMTS router increases the number of grants per interval for each application flow based on a single service flow, resulting in multiple grants per interval.

The MGPI feature supports the flow-aggregated voice MGPI functionality based on CableLabs PacketCable Specification (PKT-SP-MM-I05-091029). The flow-aggregated MGPI functionality allows the application manager to use the UGS traffic profile to explicitly set the number of grants per interval and place several application flows on a single gate. This results in an aggregated view for event messages, volume, and time usage limits.

Voice Support Over DOCSIS 3.0 E-MTAs

PacketCable and PCMM services are supported on embedded multimedia terminal adapters (E-MTAs). An E-MTA is a network element that contains the interface to a physical voice device, a network interface, and all signaling and encapsulation functions required for the VoIP transport, class features signaling, and QoS signaling.

PacketCable and PCMM Call Trace

To effectively capture signaling information, this functionality buffers signaling for a configured number of PacketCable or PCMM gates. By default, only ten user-configured gate traces are saved in a buffer. After the specified number is reached, any subsequent gate signaling information does not get buffered. When one of the gates being traced is deleted, gate signaling of a new gate is buffered.

Use the **cable dynamic-qos trace** command in global configuration mode to enable the call trace functionality for PacketCable and PacketCable Multimedia gates on the Cisco CMTS router. You will have to specify the number of subscribers for whom call trace needs to be enabled.

Verifying PacketCable and PCMM Statistics

Use the following commands to verify PacketCable and PCMM statistics on the Cisco CMTS router:

- **show interface cable dynamic-service statistics**
- **show interface cable packetcable statistics**
- **show packetcable cms**

To verify dynamic service statistics based on the cable interface, use the show interface cable dynamic-service statistics command as shown in the following example:

```
Router# show interface cable 7/1/0 dynamic-service statistics
  Upstream        Downstream
DSA REQ      0               5
DSA RSP      5               0
DSA ACK      0               5
DSC REQ      0               5
DSC RSP      5               0
DSC ACK      0               5
DSD REQ      0               0
```

```
DSD RSP      0               0
Retransmission counts
             Upstream        Downstream
DSA REQ      0               0
DSA RSP      0               0
DSA ACK      0               0
DSC REQ      0               5
DSC RSP      5               0
DSC ACK      0               0
DSD REQ      0               0
DSD RSP      0               0
```

To verify PacketCable IPC statistics based on the cable interface, use the show interface cable packetcable statistics command as shown in the following example:

```
Router# show interface cable 7/1/0 packetcable statistics
Packetcable IPC Statistics on RP
Msg   create     gate       gate       gate set   dsd
      gie        set        del        notify     notify
Sent  0          10         0          0          0
Rcvd  0          0          0          10         0
Packetcable IPC Statistics on LC
Msg   create     gate       gate       gate set   dsd
      gie        set        del        notify     notify
Sent  0          0          0          10         0
Rcvd  0          10         0          0          0
```

To verify all gate controllers that are currently connected to the PacketCable client, use the show packetcable cms command as shown in the following example:

```
Router# show packetcable cms
GC-Addr        GC-Port  Client-Addr    COPS-handle  Version PSID Key PDD-Cfg
1.100.30.2     47236    2.39.34.1      0x2FF9E268/1    4.0   0    0   0
2.39.26.19     55390    2.39.34.1      0x2FF9D890/1    1.0   0    0   2
```

To verify all gate controllers including the COPS servers for which the PacketCable connection is gone down, use the **show packetcable cms** command with the all keyword as shown in the following example:

```
Router# show packetcable cms all
GC-Addr        GC-Port  Client-Addr    COPS-handle  Version PSID Key PDD-Cfg
1.100.30.2     47236    2.39.34.1      0x2FF9E268/1    4.0   0    0   0
2.39.26.19     55390    2.39.34.1      0x2FF9D890/1    1.0   0    0   2
1.10.30.22     42307    2.39.34.1      0x0       /0    4.0   0    0   0
```

To verify gate controller statistics, use the **show packetcable cms** command with the keyword, verbose, as shown in the following example:

```
Router# show packetcable cms verbose
Gate Controller
     Addr         :     1.100.30.2
     Port         :     47236
     Client Addr  :     2.39.34.1
     COPS Handle  :     0x2FF9E268
     Version      :     4.0
     Statistics   :
        gate del =  0  gate del ack =  0   gate del err = 0
        gate info = 0  gate info ack = 0  gate info err = 0
        gate open =  0  gate report state = 0
        gate set =  0  gate set ack =  0   gate set err = 0
        gate alloc = 0 gate alloc ack = 0  gate alloc err = 0
        gate close =  0
```

```
Gate Controller
     Addr          :     2.39.26.19
     Port          :     55390
     Client Addr   :     2.39.34.1
     COPS Handle   :     0x2FF9D890
     Version       :     1.0
     Statistics    :
        gate del =  0  gate del ack =  0   gate del err = 0
        gate info = 0  gate info ack = 0  gate info err = 0
        gate open =  0  gate report state = 0
        gate set =  2  gate set ack =  2   gate set err = 0
        PCMM Timers Expired
        Timer T1 =  0  Timer T2 =  0 Timer T3 = 0 Timer T4 = 0
GC-Addr         GC-Port  Client-Addr    COPS-handle  Version PSID Key PDD-Cfg
1.100.30.2      47236    2.39.34.1      0x2FF9E268/1    4.0   0    0   0
2.39.26.19      55390    2.39.34.1      0x2FF9D890/1    1.0   0    0   2
```

Additional References

Related Documents

Related Topic	Document Title
CMTS commands	*Cisco CMTS Cable Command Reference* http://www.cisco.com/en/US/docs/ios/cable/command/reference/cbl_book.html
N+1 redundancy	*N+1 Redundancy for the Cisco CMTS Routers* http://www.cisco.com/en/US/docs/ios/cable/configuration/guide/cmts_nplus1_redun_ps2209_TSD_Pro
NTP or SNTP Configuration	To configure the Cisco CMTS router to use Network Time Protocol (NTP) or Simple Network Time P the "Performing Basic System Management" chapter in the "System Management" section of the Cisc Configuration Guide.

Standards

Standards[4]	Title
PKT-SP-MM-I06-110629	PacketCable™ Specification Multimedia Specification
ITU X.509 V3	*International Telecommunications Union (ITU) X.509 Version 3.0* standard
PKT-EM-I03-011221	*PacketCable™ Event Message Specification*
PKT-SP-DQOS-I04-021018	*PacketCable™ Dynamic Quality-of-Service Specification*
PKT-SP-EC-MGCP-I04-011221	*PacketCable™ Network-Based Call Signaling Protocol Specification*
PKT-SP-ESP-I01-991229	*PacketCable™ Electronic Surveillance Specification*
PKT-SP-ISTP-I02-011221	*PacketCable™ Internet Signaling Transport Protocol (ISTP) Specification*

Standards[4]	Title
PKT-SP-PROV-I03-011221	*PacketCable™ MTA Device Provisioning Specification*

[4] Not all supported standards are listed.

MIBs

MIBs	MIBs Link
No new or changed MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 1321	*The MD5 Message-Digest Algorithm*
RFC 1510	*The Kerberos Network Authentication Service (V5)*
RFC 2138	*Remote Authentication Dial In User Service (RADIUS)*
RFC 2205	*Resource ReSerVation Protocol (RSVP)*
RFC 2327	*SDP: Session Description Protocol*
RFC 2748	*The COPS (Common Open Policy Service) Protocol*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

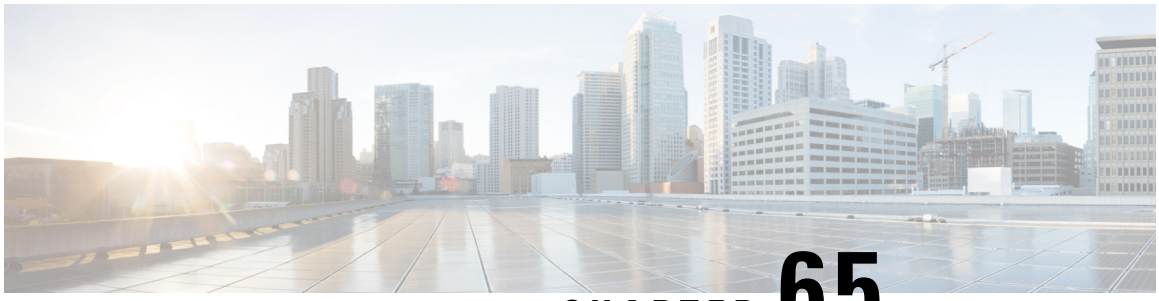
Feature Information for PacketCable and PacketCable Multimedia

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release,

feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Feature Name	Releases	Feature Information
PacketCable and PacketCable Multimedia Unicast	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Routers.
PacketCable and PacketCable Multimedia Multicast	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Routers.
DQoSLite Based IPv6 Voice Support	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Routers.



CHAPTER 65

COPS Engine Operation

This document describes the Common Open Policy Service (COPS) engine feature on the Cisco CMTS routers. The Cisco CMTS routers also support Access control lists (ACLs) with the COPS engine.

- Finding Feature Information, on page 981
- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 981
- Prerequisites for the COPS Engine on the Cisco CMTS Routers, on page 982
- Restrictions for the COPS Engine on the Cisco CMTS, on page 982
- Information About the COPS Engine on the Cisco CMTS, on page 982
- How to Configure the COPS Engine on the Cisco CMTS, on page 983
- COPS Engine Configuration Examples for Cable, on page 988
- Additional References, on page 989
- Feature Information for COPS Engine Operation, on page 990

Finding Feature Information

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 158: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for the COPS Engine on the Cisco CMTS Routers

- A compatible policy server must be connected to the network, such as the Cisco COPS QoS Policy Manager.
- Compliance with administrative policy, such as the Computer Assisted Law Enforcement Act (CALEA) or other lawful intercept (LI), is required for use of this feature on the Cisco CMTS routers.

Restrictions for the COPS Engine on the Cisco CMTS

- Resource Reservation Protocol (RSVP) is not configured on the Cisco CMTS. COPS engine configuration on the Cisco CMTS is limited to networks in which separate RSVP and COPS Servers are configured and operational.

Information About the COPS Engine on the Cisco CMTS

Common Open Policy Service (COPS) is a protocol for communicating network traffic policy information to network devices.

COPS works in correspondence with the Resource Reservation Protocol (RSVP), which is a means for reserving network resources—primarily bandwidth—to guarantee that applications sending end-to-end across the Internet will perform at the desired speed and quality. RSVP is not configured on the Cisco CMTS, but the Cisco CMTS presumes RSVP on the network for these configurations.

Refer to the Additional References, on page 989 for further information about COPS for RSVP.

How to Configure the COPS Engine on the Cisco CMTS

This section describes the tasks for configuring the COPS for RSVP feature on the Cisco CMTS.

To configure the COPS engine on the Cisco CMTS, perform the following tasks:

Configuring COPS TCP and DSCP Marking

This feature allows you to change the Differentiated Services Code Point (DSCP) marking for COPS messages that are transmitted or received by the Cisco router. The **cops ip dscp** command changes the default IP parameters for connections between the Cisco router and COPS servers in the cable network.

DSCP values are used in Quality of Service (QoS) configurations on a Cisco router to summarize the relationship between DSCP and IP precedence. This command allows COPS to remark the packets for either incoming or outbound connections.

The default setting is 0 for outbound connections. On default incoming connections, the COPS engine takes the DSCP value from the COPS server initiating the TCP connection.

Note This feature affects all TCP connections with all COPS servers.

- For messages transmitted by the Cisco router, the default DSCP value is 0.
- For incoming connections to the Cisco router, the COPS engine takes the DSCP value used by the COPS server that initiates the TCP connection, by default.
- The **cops ip dscp** command allows the Cisco router to re-mark the COPS packets for either incoming or outbound connections.
- This command affects all TCP connections with all COPS servers.
- This command does not affect existing connections to COPS servers. Once you issue this command, this function is supported only for new connections after that point in time.

Perform the following steps to enable optional DSCP marking for COPS messages on the Cisco CMTS.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cops ip dscp** [<**0-63**> | **default** | **af11-af43** | **cs1-cs7**] **Example:** `Router(config)# cops ip dscp default`	Specifies the marking for COPS messages that are transmitted by the Cisco router. The values for this command specify the markings with which COPS messages are transmitted. The following values are supported for the Cisco CMTS router: • **0-63**—DSCP value ranging from 0-63. • **af11**—Use AF11 dscp (001010) • **af12**—Use AF12 dscp (001100) • **af13**—Use AF13 dscp (001110) • **af21**—Use AF21 dscp (010010) • **af22**—Use AF22 dscp (010100) • **af23**—Use AF23 dscp (010110) • **af31**—Use AF31 dscp (011010) • **af32**—Use AF32 dscp (011100) • **af33**—Use AF33 dscp (011110) • **af41**—Use AF41 dscp (100010) • **af42**—Use AF42 dscp (100100) • **af43**—Use AF43 dscp (100110) • **cs1**—Use CS1 dscp (001000) [precedence 1] • **cs2**—Use CS2 dscp (010000) [precedence 2] • **cs3**—Use CS3 dscp (011000) [precedence 3] • **cs4**—Use CS4 dscp (100000) [precedence 4] • **cs5**—Use CS5 dscp (101000) [precedence 5] • **cs6**—Use CS6 dscp (110000) [precedence 6] • **cs7**—Use CS7 dscp (111000) [precedence 7] • **default**—Use default dscp (000000) • **ef**—Use EF dscp (101110)
Step 4	exit **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config)# exit` `Router#`	

Configuring COPS TCP Window Size

This feature allows you to override the default TCP receive window size that is used by COPS processes. This setting can be used to prevent the COPS server from sending too much data at one time.

Perform the following steps to change the TCP Window size on the Cisco CMTS.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cops tcp window-size** *bytes* **Example:** `Router(config)# cops tcp window-size 64000`	Overrides the default TCP receive window size on the Cisco CMTS. To return the TCP window size to a default setting of 4K, use the **no** form of this command. **Note** The default COPS TCP window size is 4000 bytes. **Note** This command does not affect existing connections to COPS servers. Once you issue this command, this function is supported only for new connections after that point in time. **Note** This command affects all TCP connections with all COPS servers.

Step 4	**exit**
	Example:
	`Router(config)# exit`
	`Router#`
		Returns to privileged EXEC mode.

Configuring Access Control List Support for COPS Engine

Perform the following steps to configure COPS ACLs on the Cisco CMTS.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cops listeners access-list** { *acl-num* | *acl-name* } **Example:** `Router# cops listeners access-list 40`	Configures access control lists (ACLs) for inbound connections to all COPS listener applications on the Cisco CMTS. To remove this setting from the Cisco CMTS, us the **no** form of this command.

Step 4	**exit**
	Example:
	`Router(config)# exit`
	`Router#`
		Returns to privileged EXEC mode.

What to do next

Access lists can be displayed by using the **show access-list** command in privileged EXEC mode.

Restricting RSVP Policy to Specific Access Control Lists

Perform the following steps to restrict the RSVP policy to specific ACLs, as already configured on the Cisco CMTS.

For ACL configuration, refer to the Configuring Access Control List Support for COPS Engine, on page 986.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**interface cable** (*slot* /*subslot* /*port* }
	Example:
	`Router(config)#interface cable 8/0/1`
	`Router(config-if)#`
		Enters interface configuration mode.

Step 4	**ip rsvp policy cops** *ACL-1 ACL-2* **servers** *iP-addr1 IP-addr2* **Example:** `Router(config-if)# ip rsvp policy cops 40 160 servers 161.44.130.164 161.44.129.2`	Tells the router to apply RSVP policy to messages that match the specified ACLs, and specifies the COPS server or servers for those sessions.

Step 5	exit
	Example:
	`Router(config)# exit`
	`Router#`
		Returns to privileged EXEC mode.

Displaying and Verifying COPS Engine Configuration on the Cisco CMTS

Once COPS is enabled and configured on the Cisco CMTS, you can verify and track configuration by using one or all of the **show** commands in the following steps.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**show cops servers** **Example:** `Router# show cops servers`	Displays server addresses, port, state, keepalives, and policy client information.
Step 3	**show ip rsvp policy cops** **Example:** `Router# show ip rsvp policy cops`	Displays policy server addresses, ACL IDs, and client/server connection status.

	Command or Action	Purpose
Step 4	**show ip rsvp policy** **Example:** `Router# show ip rsvp policy`	Displays ACL IDs and their connection status.

Show Commands for COPS Engine Information

The following examples display three views of the COPS engine configuration on the Cisco router. These respective show commands verify the COPS engine configuration.

Displaying COPS Servers on the Network

This example displays the policy server address, state, keepalives, and policy client information:

```
Router# show cops servers
COPS SERVER: Address: 161.44.135.172. Port: 3288. State: 0. Keepalive: 120 sec
Number of clients: 1. Number of sessions: 1.
    COPS CLIENT: Client type: 1.  State: 0.
```

Displaying COPS Policy Information on the Network

This example displays the policy server address, the ACL ID, and the client/server connection status:

```
Router# show ip rsvp policy cops
COPS/RSVP entry. ACLs: 40 60
PDPs: 161.44.135.172
Current state: Connected
Currently connected to PDP 161.44.135.172, port 0
```

Displaying Access Lists for COPS

This example displays the ACL ID numbers and the status for each ACL ID:

```
Router# show ip rsvp policy
Local policy: Currently unsupported
COPS:
ACLs: 40 60 . State: CONNECTED.
ACLs: 40 160 . State: CONNECTING.
```

COPS Engine Configuration Examples for Cable

The following sections provide COPS for RSVP configuration examples on the Cisco CMTS:

Example: COPS Server Specified

The following example specifies the COPS server and enables COPS for RSVP on the server. Both of these functions are accomplished by using the **ip rsvp policy cops** command. By implication, the default settings for all remaining COPS for RSVP commands are accepted.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip rsvp policy cops servers 161.44.130.168 161.44.129.6
Router(config)# exit
```

Example: COPS Server Display

The following examples display three views of the COPS for RSVP configuration on the router, which can be used to verify the COPS for RSVP configuration.

This example displays the policy server address, state, keepalives, and policy client information:

```
Router# show cops servers
COPS SERVER: Address: 161.44.135.172. Port: 3288. State: 0. Keepalive: 120 sec
Number of clients: 1. Number of sessions: 1.
    COPS CLIENT: Client type: 1.  State: 0.
```

This example displays the policy server address, the ACL ID, and the client/server connection status:

```
Router# show ip rsvp policy cops
COPS/RSVP entry. ACLs: 40 60
PDPs: 161.44.135.172
Current state: Connected
Currently connected to PDP 161.44.135.172, port 0
```

This example displays the ACL ID numbers and the status for each ACL ID:

```
Router# show ip rsvp policy
Local policy: Currently unsupported
COPS:
ACLs: 40 60 . State: CONNECTED.
ACLs: 40 160 . State: CONNECTING.
```

Additional References

Related Documents

Related Topic	Document Title
Cisco CMTS Commands	Cisco CMTS Cable Command Reference
COPS for RSVP	• *Configuring COPS for RSVP* http://www.cisco.com/en/US/docs/ios-xml/ios/qos_rsvp/configuration/12-4t/cops_rsvp.htm • *COPS for RSVP* http://www.cisco.com/en/US/docs/ios/12_1t/12_1t1/feature/guide/CopsRSVP.html

Standards

Standard	Title
PKT-SP-ESP-I01-991229	PacketCable™ Electronic Surveillance Specification (http://www.packetcable.com)

MIBs

MIB	MIBs Link
• No MIBs have been introduced or enhanced for support of this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
General RFC Resources	• *RFC Index Search Engine* http://www.rfc-editor.org/rfcsearch.html • *SNMP: Frequently Asked Questions About MIB RFCs* http://www.cisco.com/en/US/tech/tk648/tk362/technologies_q_and_a_item09186a00800c2612.

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Feature Information for COPS Engine Operation

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

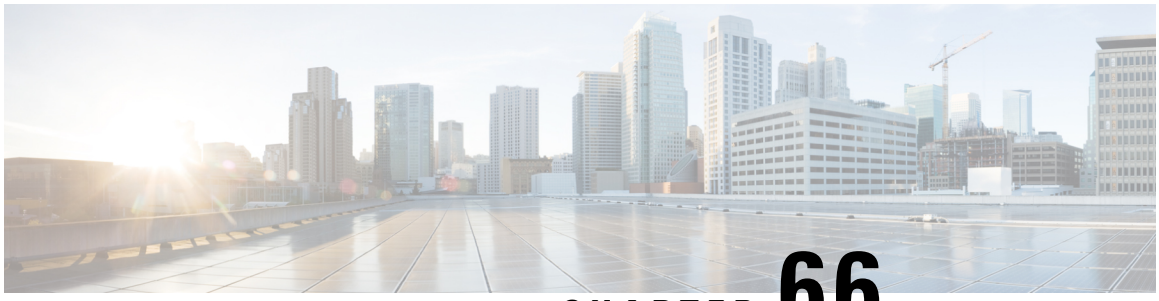
Table 159: Feature Information for COPS Engine Operation

Feature Name	Releases	Feature Information
COPS Engine Operation	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Routers.

PART X

Quality of Services Configuration

- Dynamic Bandwidth Sharing, on page 995
- Modular Quality of Service Command-Line Interface QoS, on page 1003
- DOCSIS 1.1 for the Cisco CMTS Routers, on page 1023
- Default DOCSIS 1.0 ToS Overwrite, on page 1063
- DOCSIS WFQ Scheduler on the Cisco CMTS Routers, on page 1069
- Fairness Across DOCSIS Interfaces, on page 1079
- Service Group Admission Control, on page 1093
- Subscriber Traffic Management, on page 1107



CHAPTER 66

Dynamic Bandwidth Sharing

The Cisco cBR series router enables dynamic bandwidth sharing (DBS) on integrated cable (IC) and wideband (WB) cable interfaces.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 995
- Information About Dynamic Bandwidth Sharing, on page 996
- How to Configure Dynamic Bandwidth Sharing, on page 996
- Verifying the Dynamic Bandwidth Sharing Configuration, on page 998
- Additional References, on page 1001
- Feature Information for Dynamic Bandwidth Sharing, on page 1002

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 160: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About Dynamic Bandwidth Sharing

DBS for Integrated and Wideband Cable Interfaces

Prior to DOCSIS 3.0 standards, cable service flows were associated with a single cable interface, which in turn corresponded to a physical downstream on a line card. Under DOCSIS 3.0 standards, cable service flows can be associated with more than one downstream channel.

DBS is the dynamic allocation of bandwidth for IC and WB cable interfaces sharing the same downstream channel. The bandwidth available to each IC, WB cable, or narrowband channel is not a fixed value—it depends on the configuration and the traffic load on the IC or WB cable.

DBS enables high burst rates with DOCSIS 2.0 cable modems as well as DOCSIS 3.0 cable modems. The DBS feature continues working across line card and Supervisor switchovers with no loss of functionality.

How to Configure Dynamic Bandwidth Sharing

Dynamic bandwidth sharing is enabled by default on the integrated and wideband cable interfaces on the Cisco cBR router. You can configure the bandwidth allocation for the WB and IC interfaces.

Important Dynamic bandwidth sharing cannot be disabled on the Cisco cBR router.

This section contains the following procedures:

Configuring DBS for a Wideband Cable Interface

Perform the following to configure the bandwidth allocation for a wideband cable interface.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface wideband-cable** *slot/subslot/portwideband-channel* **Example:** `Router(config)# interface wideband-cable 1/0/0:0`	Configures a wideband cable interface.
Step 4	**cable rf-channel channel-list** *grouplist* [**bandwidth-percent** *bw-percent*] **Example:** `Router(config-if)# cable rf-channel channel-list 10 bandwidth-percent 50`	Configures the bandwidth allocation for the wideband channel interface.

Configuring DBS for an Integrated Cable Interface

Perform this procedure to configure the bandwidth allocation for an integrated cable interface.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface integrated-cable** *slot/subslot/portrf-channel* **Example:** `Router(config)# interface integrated-cable 1/0/0:0`	Enters the cable interface mode.
Step 4	**cable rf-bandwidth-percent** *bw-percent* **Example:** `Router(config-if)# cable rf-bandwidth-percent 50`	Configures the bandwidth allocation for the integrated cable interface.

Verifying the Dynamic Bandwidth Sharing Configuration

Use the following commands to verify the dynamic bandwidth sharing information:

- **show controllers Integrated-Cable** *slot/subslot/port* **bandwidth rf-channel**—Displays the bandwidth information for RF channels.

  Following is a sample output of the command:

```
Router# show controllers integrated-Cable 2/0/0 bandwidth rf-channel

Ctrlr    RF       IF          CIR(Kbps)    Guar(Kbps)
2/0/0    0        In2/0/0:0   7500         13750
                  Wi2/0/0:0   7500         13750
                  Wi2/0/0:1   3750         10000
2/0/0    1        In2/0/0:1   7500         13750
                  Wi2/0/0:0   7500         13750
                  Wi2/0/0:1   3750         10000
2/0/0    2        In2/0/0:2   7500         12500
                  Wi2/0/0:0   7500         12500
                  Wi2/0/0:1   7500         12500
2/0/0    3        In2/0/0:3   7500         12500
                  Wi2/0/0:0   7500         12500
                  Wi2/0/0:1   7500         12500
2/0/0    4        In2/0/0:4   7500         12500
                  Wi2/0/0:0   7500         12500
                  Wi2/0/0:1   7500         12500
2/0/0    5        In2/0/0:5   7500         12500
                  Wi2/0/0:0   7500         12500
                  Wi2/0/0:1   7500         12500
2/0/0    6        In2/0/0:6   7500         12500
                  Wi2/0/0:0   7500         12500
                  Wi2/0/0:1   7500         12500
```

```
2/0/0    7        In2/0/0:7    7500         12500
                  Wi2/0/0:0    7500         12500
                  Wi2/0/0:1    7500         12500
2/0/0    8        In2/0/0:8    7500         18750
                  Wi2/0/0:1    7500         18750
                  Wi2/0/0:2    7500         0
2/0/0    9        In2/0/0:9    7500         18750
                  Wi2/0/0:1    7500         18750
                  Wi2/0/0:2    7500         0
2/0/0    10       In2/0/0:10   7500         18750
                  Wi2/0/0:1    7500         18750
                  Wi2/0/0:2    7500         0
2/0/0    11       In2/0/0:11   7500         18750
                  Wi2/0/0:1    7500         18750
                  Wi2/0/0:2    7500         0
2/0/0    12       In2/0/0:12   7500         37500
                  Wi2/0/0:2    7500         0
                  Wi2/0/0:3    7500         0
2/0/0    13       In2/0/0:13   7500         37500
                  Wi2/0/0:2    7500         0
                  Wi2/0/0:3    7500         0
```

- **show controllers Integrated-Cable** *slot/subslot/port* **bandwidth wb-channel**—Displays the bandwidth information for wideband channels.

  Following is a sample output of the command:

```
Router# show controllers Integrated-Cable 2/0/0 bandwidth wb-channel

Ctrlr    WB       RF           CIR(Kbps)    Guar(Kbps)
2/0/0    0                     60000        102500
                  2/0/0:0      7500         13750
                  2/0/0:1      7500         13750
                  2/0/0:2      7500         12500
                  2/0/0:3      7500         12500
                  2/0/0:4      7500         12500
                  2/0/0:5      7500         12500
                  2/0/0:6      7500         12500
                  2/0/0:7      7500         12500
2/0/0    1                     82500        170000
                  2/0/0:0      3750         10000
                  2/0/0:1      3750         10000
                  2/0/0:2      7500         12500
                  2/0/0:3      7500         12500
                  2/0/0:4      7500         12500
                  2/0/0:5      7500         12500
                  2/0/0:6      7500         12500
                  2/0/0:7      7500         12500
                  2/0/0:8      7500         18750
                  2/0/0:9      7500         18750
                  2/0/0:10     7500         18750
                  2/0/0:11     7500         18750
                  2/0/0:32     0            0
                  2/0/0:33     0            0
                  2/0/0:34     0            0
                  2/0/0:35     0            0
2/0/0    2                     60000        0
                  2/0/0:8      7500         0
                  2/0/0:9      7500         0
                  2/0/0:10     7500         0
                  2/0/0:11     7500         0
                  2/0/0:12     7500         0
                  2/0/0:13     7500         0
```

```
                2/0/0:14   7500         0
                2/0/0:15   7500         0
                2/0/0:64   0            0
                2/0/0:65   0            0
                2/0/0:66   0            0
                2/0/0:67   0            0
```

- **show controllers Integrated-Cable** *slot/subslot/port* **mapping rf-channel**—Displays the mapping for RF channels.

  Following is a sample output of the command:

```
Router# show controllers integrated-Cable 2/0/0 mapping rf-channel

Ctrlr    RF       IC %       IC Rem     WB         WB %       WB Rem
2/0/0    0        20         1          2/0/0:0    20         1
                                        2/0/0:1    10         1
2/0/0    1        20         1          2/0/0:0    20         1
                                        2/0/0:1    10         1
2/0/0    2        20         1          2/0/0:0    20         1
                                        2/0/0:1    20         1
2/0/0    3        20         1          2/0/0:0    20         1
                                        2/0/0:1    20         1
2/0/0    4        20         1          2/0/0:0    20         1
                                        2/0/0:1    20         1
2/0/0    5        20         1          2/0/0:0    20         1
                                        2/0/0:1    20         1
2/0/0    6        20         1          2/0/0:0    20         1
                                        2/0/0:1    20         1
2/0/0    7        20         1          2/0/0:0    20         1
                                        2/0/0:1    20         1
2/0/0    8        20         1          2/0/0:1    20         1
                                        2/0/0:2    20         1
2/0/0    9        20         1          2/0/0:1    20         1
                                        2/0/0:2    20         1
2/0/0    10       20         1          2/0/0:1    20         1
                                        2/0/0:2    20         1
```

- **show controllers Integrated-Cable** *slot/port/interface-number* **mapping wb-channel**—Displays the mapping for wideband channels.

  Following is a sample output of the command:

```
Router# show controllers integrated-Cable 2/0/0 mapping wb-channel

Ctrlr    WB       RF         WB %   WB Rem
2/0/0    0        2/0/0:0    20     1
                  2/0/0:1    20     1
                  2/0/0:2    20     1
                  2/0/0:3    20     1
                  2/0/0:4    20     1
                  2/0/0:5    20     1
                  2/0/0:6    20     1
                  2/0/0:7    20     1
2/0/0    1        2/0/0:0    10     1
                  2/0/0:1    10     1
                  2/0/0:2    20     1
                  2/0/0:3    20     1
                  2/0/0:4    20     1
                  2/0/0:5    20     1
                  2/0/0:6    20     1
                  2/0/0:7    20     1
                  2/0/0:8    20     1
                  2/0/0:9    20     1
```

```
                    2/0/0:10    20      1
                    2/0/0:11    20      1
                    2/0/0:32    20      1
                    2/0/0:33    20      1
                    2/0/0:34    20      1
                    2/0/0:35    20      1
2/0/0       2       2/0/0:8     20      1
                    2/0/0:9     20      1
                    2/0/0:10    20      1
                    2/0/0:11    20      1
                    2/0/0:12    20      1
                    2/0/0:13    20      1
                    2/0/0:14    20      1
                    2/0/0:15    20      1
                    2/0/0:64    20      1
                    2/0/0:65    20      1
                    2/0/0:66    20      1
                    2/0/0:67    20      1
2/0/0       3       2/0/0:12    20      1
                    2/0/0:13    20      1
                    2/0/0:14    20      1
                    2/0/0:15    20      1
                    2/0/0:16    20      1
                    2/0/0:17    20      1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco CMTS cable commands	*Cisco CMTS Cable Command Reference*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

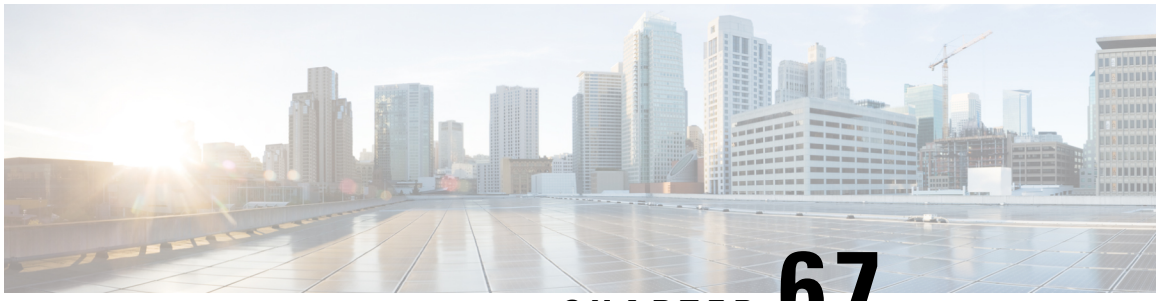
Feature Information for Dynamic Bandwidth Sharing

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 161: Feature Information for Dynamic Bandwidth Sharing

Feature Name	Releases	Feature Information
Dynamic bandwidth sharing	Cisco IOS-XE 16.5.1	This feature was integrated into Cisco <polaris release label> on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 67

Modular Quality of Service Command-Line Interface QoS

This module contains the concepts about applying QoS features using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC) and the tasks for configuring the MQC. The MQC allows you to define a traffic class, create a traffic policy (policy map), and attach the traffic policy to an interface. The traffic policy contains the QoS feature that will be applied to the traffic class.

- Finding Feature Information, on page 1003
- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1003
- Restrictions for Applying QoS Features Using the MQC, on page 1004
- About, on page 1005
- How to Apply QoS Features Using the MQC, on page 1010
- Configuration Examples for Applying QoS Features Using the MQC, on page 1014
- How to Configure Input MQC on the Port-Channel Interfaces, on page 1018
- Example: Configuring Input MQC on the Port-Channel Interfaces, on page 1020
- Additional References, on page 1020
- Feature Information for Modular Quality of Service Command-Line Interface QoS, on page 1021

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 162: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for Applying QoS Features Using the MQC

The MQC-based QoS does not support classification of legacy Layer 2 protocol packets such as Internetwork Packet Exchange (IPX), DECnet, or AppleTalk. When these types of packets are being forwarded through a generic Layer 2 tunneling mechanism, the packets can be handled by MQC but without protocol classification. As a result, legacy protocol traffic in a Layer 2 tunnel is matched only by a "match any" class or class-default.

The number of QoS policy maps and class maps supported varies by platform and release.

Note The policy map limitations do not refer to the number of applied policy map instances, only to the definition of the policy maps.

About

The MQC Structure

The MQC (Modular Quality of Service (QoS) Command-Line Interface (CLI)) enables you to set packet classification and marking based on a QoS group value. MQC CLI allows you to create traffic classes and policies, enable a QoS feature (such as packet classification), and attach these policies to interfaces.

The MQC structure necessitates developing the following entities: traffic class, policy map, and service policy.

Elements of a Traffic Class

A traffic class contains three major elements: a traffic class name, a series of **match** commands, and, if more than one **match** command is used in the traffic class, instructions on how to evaluate these **match** commands.

The **match** commands are used for classifying packets. Packets are checked to determine whether they meet the criteria specified in the **match**commands; if a packet meets the specified criteria, that packet is considered a member of the class. Packets that fail to meet the matching criteria are classified as members of the default traffic class.

Available match Commands

The table below lists *some* of the available **match** commands that can be used with the MQC. The available **match** commands vary by Cisco IOS XE release. For more information about the commands and command syntax, see the *Cisco IOS Quality of Service Solutions* Command Reference.

Table 163: match Commands That Can Be Used with the MQC

Command	Purpose
match access-group	Configures the match criteria for a class map on the basis of the specified access control list (ACL).
match any	Configures the match criteria for a class map to be successful match criteria for all packets.
match cos	Matches a packet based on a Layer 2 class of service (CoS) marking.
match destination-address mac	Uses the destination MAC address as a match criterion.
match discard-class	Matches packets of a certain discard class.
match [**ip**] **dscp**	Identifies a specific IP differentiated service code point (DSCP) value as a match criterion. Up to eight DSCP values can be included in one match statement.
match input-interface	Configures a class map to use the specified input interface as a match criterion.
match ip rtp	Configures a class map to use the Real-Time Transport Protocol (RTP) port as the match criterion.

Command	Purpose
match mpls experimental	Configures a class map to use the specified value of the Multiprotocol Label Switching (MPLS) experimental (EXP) field as a match criterion.
match mpls experimental topmost	Matches the MPLS EXP value in the topmost label.
match not	Specifies the single match criterion value to use as an unsuccessful match criterion. **Note** The **match not** command, rather than identifying the specific match parameter to use as a match criterion, is used to specify a match criterion that prevents a packet from being classified as a member of the class. For instance, if the **match not qos-group 6** command is issued while you configure the traffic class, QoS group 6 becomes the only QoS group value that is not considered a successful match criterion. All other QoS group values would be successful match criteria.
match packet length	Specifies the Layer 3 packet length in the IP header as a match criterion in a class map.
match port-type	Matches traffic on the basis of the port type for a class map.
match [**ip**] **precedence**	Identifies IP precedence values as match criteria.
match protocol	Configures the match criteria for a class map on the basis of the specified protocol. **Note** A separate **match protocol** (NBAR) command is used to configure network-based application recognition (NBAR) to match traffic by a protocol type known to NBAR.
match protocol fasttrack	Configures NBAR to match FastTrack peer-to-peer traffic.
match protocol gnutella	Configures NBAR to match Gnutella peer-to-peer traffic.
match protocol http	Configures NBAR to match Hypertext Transfer Protocol (HTTP) traffic by URL, host, Multipurpose Internet Mail Extension (MIME) type, or fields in HTTP packet headers.
match protocol rtp	Configures NBAR to match RTP traffic.
match qos-group	Identifies a specific QoS group value as a match criterion.
match source-address mac	Uses the source MAC address as a match criterion.

Multiple match Commands in One Traffic Class

If the traffic class contains more than one **match** command, you need to specify how to evaluate the **match** commands. You specify this by using either the **match-any** or **match-all** keyword of the **class-map** command. Note the following points about the **match-any** and **match-all** keywords:

- If you specify the **match-any** keyword, the traffic being evaluated by the traffic class must match *one* of the specified criteria.
- If you specify the **match-all** keyword, the traffic being evaluated by the traffic class must match *all* of the specified criteria.
- If you do not specify either keyword, the traffic being evaluated by the traffic class must match *all* of the specified criteria (that is, the behavior of the **match-all** keyword is used).

Elements of a Traffic Policy

A traffic policy contains three elements: a traffic policy name, a traffic class (specified with the **class** command), and the command used to enable the QoS feature.

The traffic policy (policy map) applies the enabled QoS feature to the traffic class once you attach the policy map to the interface (by using the **service-policy** command).

Note A packet can match only *one* traffic class within a traffic policy. If a packet matches more than one traffic class in the traffic policy, the *first* traffic class defined in the policy will be used.

Commands Used to Enable QoS Features

The commands used to enable QoS features vary by Cisco IOS XE release. The table below lists *some* of the available commands and the QoS features that they enable. For complete command syntax, see the *Cisco IOS QoS Command Reference*.

For more information about a specific QoS feature that you want to enable, see the appropriate module of the Cisco IOS XE Quality of Service Solutions Configuration Guide.

Table 164: Commands Used to Enable QoS Features

Command	Purpose
bandwidth	Configures a minimum bandwidth guarantee for a class.
bandwidth remaining	Configures an excess weight for a class.
fair-queue	Enables the flow-based queueing feature within a traffic class.
fair-queue pre-classify	Configures and checks whether the **qos pre-classify** command can be used for fair queue. When the **qos pre-classify** command is enabled on the tunnel inteface, and then the **fair-queue pre-classify** command is enabled for the policy-map, the policy-map is attached to either the tunnel interface or the physical interface. The inner IP header of the tunnel will be used for the hash algorithm of the fair queue.
drop	Discards the packets in the specified traffic class.
police	Configures traffic policing.

Command	Purpose
police (percent)	Configures traffic policing on the basis of a percentage of bandwidth available on an interface.
police (two rates)	Configures traffic policing using two rates, the committed information rate (CIR) and the peak information rate (PIR).
priority	Gives priority to a class of traffic belonging to a policy map.
queue-limit	Specifies or modifies the maximum number of packets the queue can hold for a class configured in a policy map.
random-detect	Enables Weighted Random Early Detection (WRED).
random-detect discard-class	Configures the WRED parameters for a discard-class value for a class in a policy map.
random-detect discard-class-based	Configures WRED on the basis of the discard class value of a packet.
random-detect exponential-weighting-constant	Configures the exponential weight factor for the average queue size calculation for the queue reserved for a class.
random-detect precedence	Configure the WRED parameters for a particular IP Precedence for a class policy in a policy map.
service-policy	Specifies the name of a traffic policy used as a matching criterion (for nesting traffic policies [hierarchical traffic policies] within one another).
set atm-clp	Sets the cell loss priority (CLP) bit when a policy map is configured.
set cos	Sets the Layer 2 class of service (CoS) value of an outgoing packet.
set discard-class	Marks a packet with a discard-class value.
set [**ip**] **dscp**	Marks a packet by setting the differentiated services code point (DSCP) value in the type of service (ToS) byte.
set fr-de	Changes the discard eligible (DE) bit setting in the address field of a Frame Relay frame to 1 for all traffic leaving an interface.
set mpls experimental	Designates the value to which the MPLS bits are set if the packets match the specified policy map.
set precedence	Sets the precedence value in the packet header.
set qos-group	Sets a QoS group identifier (ID) that can be used later to classify packets.
shape	Shapes traffic to the indicated bit rate according to the algorithm specified.

Nested Traffic Classes

The MQC does not necessarily require that you associate only one traffic class to one traffic policy.

In a scenario where packets satisfy more than one match criterion, the MQC enables you to associate multiple traffic classes with a single traffic policy (also termed nested traffic classes) using the **match class-map** command. (We term these *nested class maps* or *MQC Hierarchical class maps*.) This command provides the only method of combining match-any and match-all characteristics within a single traffic class. By doing so, you can create a traffic class using one match criterion evaluation instruction (either match-any or match-all) and then use that traffic class as a match criterion in a traffic class that uses a different match criterion type. For example, a traffic class created with the match-any instruction must use a class configured with the match-all instruction as a match criterion, or vice versa.

Consider this likely scenario: Suppose A, B, C, and D were all separate match criterion, and you wanted traffic matching A, B, or C and D (i.e., A or B or [C and D]) to be classified as belonging to a traffic class. Without the nested traffic class, traffic would either have to match all four of the match criterion (A and B and C and D) or match any of the match criterion (A or B or C or D) to be considered part of the traffic class. You would not be able to combine "and" (match-all) and "or" (match-any) statements within the traffic class; you would be unable to configure the desired configuration.

The solution: Create one traffic class using match-all for C and D (which we will call criterion E), and then create a new match-any traffic class using A, B, and E. The new traffic class would have the correct evaluation sequence (A or B or E, which is equivalent to A or B or [C and D]).

match-all and match-any Keywords of the class-map Command

One of the commands used when you create a traffic class is the **class-map** command. The command syntax for the **class-map** command includes two keywords: **match-all** and **match-any** . The **match-all** and **match-any** keywords need to be specified only if more than one match criterion is configured in the traffic class. Note the following points about these keywords:

- The **match-all** keyword is used when *all* of the match criteria in the traffic class must be met in order for a packet to be placed in the specified traffic class.
- The **match-any** keyword is used when only *one* of the match criterion in the traffic class must be met in order for a packet to be placed in the specified traffic class.
- If neither the **match-all** keyword nor **match-any** keyword is specified, the traffic class will behave in a manner consistent with the **match-all** keyword.

input and output Keywords of the service-policy Command

As a general rule, the QoS features configured in the traffic policy can be applied to packets entering the interface or to packets leaving the interface. Therefore, when you use the **service-policy** command, you need to specify the direction of the traffic policy by using the **input** or **output** keyword.

For instance, the **service-policy output policy-map1** command would apply the QoS features in the traffic policy to the interface in the output direction. All packets leaving the interface (output) are evaluated according to the criteria specified in the traffic policy named policy-map1.

Note For Cisco releases, queueing mechanisms are not supported in the input direction. Nonqueueing mechanisms (such as traffic policing and traffic marking) are supported in the input direction. Also, classifying traffic on the basis of the source MAC address (using the **match source-address mac** command) is supported in the input direction only.

Benefits of Applying QoS Features Using the MQC

The MQC structure allows you to create the traffic policy (policy map) once and then apply it to as many traffic classes as needed. You can also attach the traffic policies to as many interfaces as needed.

How to Apply QoS Features Using the MQC

Creating a Traffic Class

To create a traffic class, use the **class-map** command to specify the traffic class name. Then use one or more **match** commands to specify the appropriate match criteria. Packets matching the criteria that you specify are placed in the traffic class. For more information about the **match-all** and **match-any** keywords of the class-map comand, see the "match-all and match-any Keywords of the class-map Command" section.

Note The **match cos** command is shown in Step 4. The **match cos** command is simply an example of one of the **match** commands that you can use. For information about the other available **match** commands, see the "match-all and match-any Keywords of the class-map Command" section.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map** [**match-all** | **match-any**] *class-map-name* **Example:** `Router(config)# class-map match-any class1`	Creates a class to be used with a class map and enters class-map configuration mode. • The class map is used for matching packets to the specified class. • Enter the class name. **Note** The **match-all** keyword specifies that all match criteria must be met. The **match-any** keyword specifies that one of the match criterion must be met. Use these keywords only if you will be specifying more than one **match** command.

	Command or Action	Purpose
Step 4	**match cos** *cos-number* **Example:** `Router(config-cmap)# match cos 2`	Matches a packet on the basis of a Layer 2 class of service (CoS) number. • Enter the CoS number. **Note** The **match cos** command is an example of the **match** commands you can use. For information about the other **match** commands that are available, see the "match-all and match-any Keywords of the class-map Command" section.
Step 5	Enter additional match commands, if applicable; otherwise, continue with step 6.	--
Step 6	**end** **Example:** `Router(config-cmap)# end`	(Optional) Exits QoS class-map configuration mode and returns to privileged EXEC mode.

Creating a Traffic Policy

Note The **bandwidth** command is shown in Step 5. The **bandwidth** command is an example of the commands that you can use in a policy map to enable a QoS feature (in this case, Class-based Weighted Fair Queuing (CBWFQ). For information about other available commands, see the "Elements of a Traffic Policy" section.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map policy1`	Creates or specifies the name of the traffic policy and enters QoS policy-map configuration mode. • Enter the policy map name.

	Command or Action	Purpose
Step 4	**class** { *class-name* | **class-default** } **Example:** `Router(config-pmap)# class class1`	Specifies the name of a traffic class and enters QoS policy-map class configuration mode. **Note** This step associates the traffic class with the traffic policy.
Step 5	**bandwidth** {*bandwidth-kbps* | **percent** *percent* } **Example:** `Router(config-pmap-c)# bandwidth 3000`	(Optional) Specifies a minimum bandwidth guarantee to a traffic class in periods of congestion. • A minimum bandwidth guarantee can be specified in kb/s or by a percentage of the overall available bandwidth. **Note** The **bandwidth** command enables CBWFQ. The **bandwidth** command is an example of the commands that you can use in a policy map to enable a QoS feature. For information about the other commands available, see the "Elements of a Traffic Policy" section.
Step 6	Enter the commands for any additional QoS feature that you want to enable, if applicable; otherwise, continue with Step 7.	--
Step 7	**end** **Example:** `Router(config-pmap-c)# end`	(Optional) Exits QoS policy-map class configuration mode and returns to privileged EXEC mode.

Attaching a Traffic Policy to an Interface Using the MQC

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Router(config)# interface TenGigabitEthernet 4/1/0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and interface number.
Step 4	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy input policy1`	Attaches a policy map to an interface. • Enter either the **input** or **output** keyword and the policy map name.
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

Verifying the Traffic Class and Traffic Policy Information

The show commands described in this section are optional and can be entered in any order.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show class-map** **Example:** `Router# show class-map`	(Optional) Displays all class maps and their matching criteria.
Step 3	**show policy-map** *policy-map-name* **class** *class-name* **Example:** `Router# show policy-map policy1 class class1`	(Optional) Displays the configuration for the specified class of the specified policy map. • Enter the policy map name and the class name.
Step 4	**show policy-map** **Example:**	(Optional) Displays the configuration of all classes for all existing policy maps.

	Command or Action	Purpose
	`Router# show policy-map`	
Step 5	**show policy-map interface** *type number* **Example:** `Router# show policy-map interface TengigabitEthernet 4/1/0`	(Optional) Displays the statistics and the configurations of the input and output policies that are attached to an interface. • Enter the interface type and number.
Step 6	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for Applying QoS Features Using the MQC

Creating a Traffic Class

In the following example, we create traffic classes and define their match criteria. For the first traffic class (class1), we use access control list (ACL) 101 as match criteria; for the second traffic class (class2), ACL 102. We check the packets against the contents of these ACLs to determine if they belong to the class.

```
class-map class1
  match access-group 101
  exit
class-map class2
  match access-group 102
  end
```

Creating a Policy Map

In the following example, we define a traffic policy (policy1) containing the QoS features that we will apply to two classes: class1 and class2. The match criteria for these classes were previously defined in Creating a Traffic Class, on page 1014).

For class1, the policy includes a bandwidth allocation request and a maximum packet count limit for the queue reserved for that class. For class2, the policy specifies only a bandwidth allocation request.

```
policy-map policy1
  class class1
    bandwidth 3000
    queue-limit 30
    exit
  class class2
    bandwidth 2000
    end
```

Example: Attaching a Traffic Policy to an Interface

The following example shows how to attach an existing traffic policy to an interface. After you define a traffic policy with the **policy-map** command, you can attach it to one or more interfaces by using the **service-policy** command in interface configuration mode. Although you can assign the same traffic policy to multiple interfaces, each interface can have only one traffic policy attached in the input direction and only one traffic policy attached in the output direction.

```
Router(config)# interface TengigabitEthernet 4/1/0
Router(config-if)# service-policy output policy1
Router(config-if)# exit
Router(config)# interface TengigabitEthernet 4/1/0
Router(config-if)# service-policy output policy1
Router(config-if)# end
```

Using the match not Command

Use the **match not** command to specify a QoS policy value that is not used as a match criterion. All other values of that QoS policy become successful match criteria. For instance, if you issue the **match not qos-group 4** command in QoS class-map configuration mode, the specified class will accept all QoS group values except 4 as successful match criteria.

In the following traffic class, all protocols except IP are considered successful match criteria:

```
class-map noip
  match not protocol ip
  end
```

Configuring a Default Traffic Class

Traffic that does not meet the match criteria specified in the traffic classes (i.e., *unclassified traffic*) is treated as belonging to the default traffic class.

If you do not configure a default class, packets are still treated as members of that class. The default class has no QoS features enabled so packets belonging to this class have no QoS functionality. Such packets are placed into a first-in, first-out (FIFO) queue managed by tail drop, which is a means of avoiding congestion that treats all traffic equally and does not differentiate between classes of service. Queues fill during periods of congestion. When the output queue is full and tail drop is active, packets are dropped until the congestion is eliminated and the queue is no longer full.

The following example configures a policy map (policy1) for the default class (always called class-default) with these characteristics: 10 queues for traffic that does not meet the match criteria of other classes whose policy is defined by class policy1, and a maximum of 20 packets per queue before tail drop is enacted to handle additional queued packets.

```
policy-map policy1
  class class-default
    fair-queue
    queue-limit 20
```

In the following example, we configure a policy map (policy1) for the default class (always termed class-default) with these characteristics: 10 queues for traffic that does not meet the match criterion of other classes whose policy is defined by the traffic policy policy1.

```
policy-map policy1
  class class-default
    shape average 100m
```

How Commands "class-map match-any" and "class-map match-all" Differ

This example shows how packets are evaluated when multiple match criteria exist. It illustrates the difference between the **class-map match-any** and **class-map match-all** commands. Packets must meet either all of the match criteria (**match-all**) or one of the match criteria (**match-any**) to be considered a member of the traffic class.

The following examples show a traffic class configured with the **class-map match-all** command:

```
class-map match-all cisco1
  match protocol ip
  match qos-group 4
  match access-group 101
```

If a packet arrives on a router with traffic class cisco1 configured on the interface, we assess whether it matches the IP protocol, QoS group 4, and access group 101. If all of these match criteria are met, the packet is classified as a member of the traffic class cisco1 (a logical AND operator; Protocol IP AND QoS group 4 AND access group 101).

```
class-map match-all vlan
  match vlan 1
  match vlan inner 1
```

The following example illustrates use of the **class-map match-any** command. Only one match criterion must be met for us to classify the packet as a member of the traffic class (i.e., a logical OR operator; protocol IP OR QoS group 4 OR access group 101):

```
class-map match-any cisco2
  match protocol ip
  match qos-group 4
  match access-group 101
```

In the traffic class cisco2, the match criterion are evaluated consecutively until a successful match is located. The packet is first evaluated to determine whether the IP protocol can be used as a match criterion. If so, the packet is matched to traffic class cisco2. If not, then QoS group 4 is evaluated as a match criterion and so on. If the packet matches none of the specified criteria, the packet is classified as a member of the default traffic class (*class default-class*).

Establishing Traffic Class as a Match Criterion (Nested Traffic Classes)

There are two reasons to use the **match class-map** command. One reason is maintenance; if a large traffic class currently exists, using the traffic class match criterion is easier than retyping the same traffic class configuration. The second and more common reason is to mix match-all and match-any characteristics in one

traffic policy. This enables you to create a traffic class using one match criterion evaluation instruction (either match-any or match-all) and then use that traffic class as a match criterion in a traffic class that uses a different match criterion type.

Consider this likely scenario: Suppose A, B, C, and D were all separate match criterion, and you wanted traffic matching A, B, or C and D (i.e., A or B or [C and D]) to be classified as belonging to a traffic class. Without the nested traffic class, traffic would either have to match all four of the match criterion (A and B and C and D) or match any of the match criterion (A or B or C or D) to be considered part of the traffic class. You would not be able to combine "and" (match-all) and "or" (match-any) statements within the traffic class; you would be unable to configure the desired configuration.

The solution: Create one traffic class using match-all for C and D (which we will call criterion E), and then create a new match-any traffic class using A, B, and E. The new traffic class would have the correct evaluation sequence (A or B or E, which is equivalent to A or B or [C and D]).

Example: Nested Traffic Class for Maintenance

In the following example, the traffic class called class1 has the same characteristics as the traffic class called class2, with the exception that traffic class class1 has added a destination address as a match criterion. Rather than configuring traffic class class1 line by line, you can enter the **match class-map class2** command. This command allows all of the characteristics in the traffic class called class2 to be included in the traffic class called class1, and you can add the new destination address match criterion without reconfiguring the entire traffic class.

```
Router(config)# class-map match-any class2
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 3
Router(config-cmap)# match access-group 2
Router(config-cmap)# exit
Router(config)# class-map match-all class1
Router(config-cmap)# match class-map class2
Router(config-cmap)# match destination-address mac 0000.0000.0000
Router(config-cmap)# exit
```

Example: Nested Traffic Class to Combine match-any and match-all Characteristics in One Traffic Class

The only method of including both match-any and match-all characteristics in a single traffic class is to use the **match class-map** command. To combine match-any and match-all characteristics into a single class, use the match-any instruction to create a traffic class that uses a class configured with the match-all instruction as a match criterion (through the **match class-map** command).

The following example shows how to combine the characteristics of two traffic classes, one with match-any and one with match-all characteristics, into one traffic class with the **match class-map** command. The result requires a packet to match one of the following three match criteria to be considered a member of traffic class class4: IP protocol *and* QoS group 4, destination MAC address 00.00.00.00.00.00, or access group 2.

In this example, only the traffic class called class4 is used with the traffic policy called policy1.

```
Router(config)# class-map match-all class3
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 4
Router(config-cmap)# exit
Router(config)# class-map match-any class4
Router(config-cmap)# match class-map class3
```

```
Router(config-cmap)# match destination-address mac 00.00.00.00.00.00
Router(config-cmap)# match access-group 2
Router(config-cmap)# exit
Router(config)# policy-map policy1
Router(config-pmap)# class class4
Router(config-pmap-c)# police 8100 1500 2504 conform-action transmit exceed-action
set-qos-transmit 4
Router(config-pmap-c)# end
```

Example: Traffic Policy as a QoS Policy (Hierarchical Traffic Policies)

A traffic policy can be included in a QoS policy when the **service-policy** command is used in QoS policy-map class configuration mode. A traffic policy that contains a traffic policy is called a hierarchical traffic policy.

A hierarchical traffic policy contains a child policy and a parent policy. The child policy is the previously defined traffic policy that is being associated with the new traffic policy through the use of the **service-policy** command. The new traffic policy using the preexisting traffic policy is the parent policy. In the example in this section, the traffic policy called child is the child policy and traffic policy called parent is the parent policy.

Hierarchical traffic policies can be attached to subinterfaces. When hierarchical traffic policies are used, a single traffic policy (with a child and parent policy) can be used to shape and priority traffic on subinterfaces.

```
Router(config)# policy-map child
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50
Router(config)# policy-map parent
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 10000000
Router(config-pmap-c)# service-policy child
```

The value used with the **shape** command is provisioned from the committed information rate (CIR) value from the service provider.

How to Configure Input MQC on the Port-Channel Interfaces

To configure input MQC on a port-channel interface to differentiate traffic flow and set corresponding "qos-group" features, follow the steps given below.

Restriction

- QoS actions like policing, shaping, WRED, and queuing are not supported.
- Input MQC cannot be configured on cable physical interfaces.

Creating a Traffic Class

The **class-map** command is used to create a traffic class. A traffic class contains three major elements: a name, a series of **match** commands, and, if more than one **match** command exists in the traffic class, an instruction on how to evaluate these **match** commands.

The **match** commands are used to specify various criteria for classifying packets. Packets are checked to determine whether they match the criteria specified in the **match** commands; if a packet matches the specified criteria, that packet is considered a member of the class and is forwarded according to the QoS specifications set in the traffic policy. Packets that fail to meet any of the matching criteria are classified as members of the default traffic class.

To create traffic classes and define their match criteria, complete the following procedure:

```
configure terminal
class-map class
match type
```

Creating a Policy Map

After creating traffic classes, you can configure traffic policies to configure marking features to apply certain actions to the selected traffic in those classes.

The **policy-map** command is used to create a traffic policy. The purpose of a traffic policy is to configure the QoS features that should be associated with the traffic that has been classified in a user-specified traffic class.

Note A packet can match only one traffic class within a traffic policy. If a packet matches more than one traffic class in the traffic policy, the first traffic class defined in the policy will be used.

To define a traffic policy, complete the following procedure:

```
configure terminal
policy-map policy
class class
```

Defining QoS Actions in a Policy Map

Action commands can be added from within class mode on a policy map.

Set Actions

Set commands allow traffic to be marked such that other network devices along the forwarding path can quickly determine the proper class of service to apply to a traffic flow.

To define a set action, complete the following procedure:

```
configure terminal
policy-map policy
class class
set option
```

Configuring Aggregate Port-Channel Interface

To configure port-channel interface, complete the following procedure:

```
configure terminal
platform qos port-channel-aggregate port_channel_number
```

```
interface port-channel port_channel_number
ip address ip musk
interface name
channel-group number
```

Attaching a Traffic Policy to an Interface

After you define a traffic policy with the **policy-map**command, you can attach it to one or more interfaces by using the **service-policy** command in interface configuration mode. Although you can assign the same traffic policy to multiple interfaces, each interface can have only one traffic policy attached in the input direction and only one traffic policy attached in the output direction.

To attach a traffic policy to an interface, complete the following procedure:

```
configure terminal
interface port-channel port_channel_number
service-policy input policy
```

Example: Configuring Input MQC on the Port-Channel Interfaces

The following example shows how to configure input MQC on the port-channel interfaces.

```
Router# configure terminal
Router(config)# class-map class1
Router(config-cmap)# match any
Router(config-cmap)# exit
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# set dscp af11
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# platform qos port-channel-aggregate 2 Router(config)# interface port-channel
 2
Router(config-if)# ip address 192.168.0.1 255.255.255.0
Router(config-if)# no shut
Router(config-if)# interface tenGigabitEthernet 4/1/1
Router(config-if)# no ip address
Router(config-if)# no shut
Router(config-if)# channel-group 2
Router(config-if)# interface port-channel 2
Router(config-if)# service-policy input policy1
Device(config-if)# end
```

Additional References

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*

Related Topic	Document Title
Packet classification	"Classifying Network Traffic" module
Frame Relay Fragmentation (FRF) PVCs	"FRF .20 Support" module
Selective Packet Discard	"IPv6 Selective Packet Discard" module
Scaling and performance information	"Broadband Scalability and Performance" module of the Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide .

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Modular Quality of Service Command-Line Interface QoS

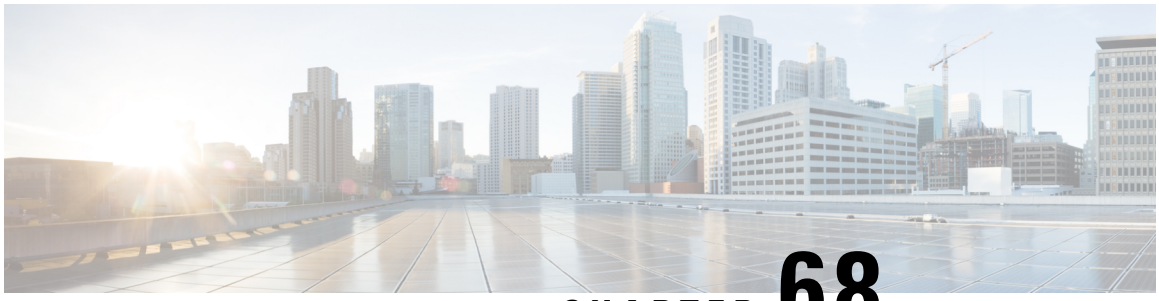
Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 165: Feature Information for Modular Quality of Service Command-Line Interface QoS

Feature Name	Releases	Feature Information
Modular Quality of Service Command-Line Interface QoS	Cisco IOS-XE 16.5.1	This feature was integrated into Cisco <polaris release label> on Cisco cBR Series Converged Broadband Router s.

Feature Name	Releases	Feature Information
Service Policy on Port-Channel Interfaces	Cisco IOS-XE 16.5.1	This feature was integrated into Cisco <polaris release label> on Cisco cBR Series Converged Broadband Router s.



CHAPTER 68

DOCSIS 1.1 for the Cisco CMTS Routers

This document describes how to configure the Cisco CMTS router for Data-over-Cable Service Interface Specifications (DOCSIS) 1.1 operations.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1023
- Prerequisites for DOCSIS 1.1 Operations, on page 1024
- Restrictions for DOCSIS 1.1 Operations, on page 1025
- Information about DOCSIS 1.1, on page 1027
- How to Configure the Cisco CMTS for DOCSIS 1.1 Operations, on page 1039
- Monitoring DOCSIS Operations, on page 1051
- Configuration Examples for DOCSIS 1.1 Operations, on page 1057
- Additional References, on page 1060
- Feature Information for DOCSIS 1.1 for Cisco CMTS Routers, on page 1061

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 166: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for DOCSIS 1.1 Operations

To support DOCSIS 1.1 operations, the cable modem must also support the DOCSIS 1.1 feature set. In addition, before you power on and configure the Cisco CMTS, check the following points:

- Ensure that your network supports reliable broadband data transmission. Your plant must be swept, balanced, and certified, based on NTSC or appropriate international cable plant recommendations. Ensure that your plant meets all DOCSIS downstream and upstream RF requirements.
- Ensure that your Cisco CMTS is installed according to the instructions provided in the appropriate Hardware Installation Guide. The chassis must contain at least one port adapter to provide backbone connectivity and one Cisco cable line card to serve as the RF cable TV interface.
- Ensure that all other required headend or distribution hub routing and network interface equipment is installed, configured, and operational, based on the services to support. This includes all routers, servers (DHCP, TFTP, and ToD), network management systems, and other configuration or billing systems. This includes IP telephony equipment including gatekeepers and gateways; backbone and other equipment if supporting virtual private networks (VPNs); and dialup access servers, telephone circuits and connections and other equipment if supporting telco return.
- Ensure that DHCP and DOCSIS configuration files have been created and pushed to appropriate servers such that each cable modem, when initialized, can transmit a DHCP request, receive an IP address, obtain TFTP and ToD server addresses, and download DOCSIS configuration files. Optionally, ensure that your servers can also download updated software images to DOCSIS 1.0 and DOCSIS 1.1 cable modems.

- Ensure that customer premises equipment (CPE)—cable modems or set-top boxes, PCs, telephones, or facsimile machines—meet the requirements for your network and service offerings.
- Familiarize yourself with your channel plan to ensure assigning of appropriate frequencies. Outline your strategies for setting up bundling or VPN solution sets, if applicable, to your headend or distribution hub. Know your dial plan if using H.323 for VoIP services and setting up VoIP-enabled cable modem configuration files. Obtain passwords, IP addresses, subnet masks, and device names, as appropriate.
- Ensure that the system clocks on the Cisco CMTS and on the time-of-day (ToD) servers are synchronized. If this does not occur, the clocks on the CMs will not match the clocks on the Cisco CMTS, which could interfere with BPI+ operations. In particular, this could prevent the proper verification of the digital certificates on the cable modem (CM).

After these prerequisites are met, you are ready to configure the Cisco CMTS. This includes, at a minimum, configuring a host name and password for the Cisco CMTS and configuring the Cisco CMTS to support IP over the cable plant and network backbone.

Caution If you plan to use service-class-based provisioning, the service classes must be configured at the Cisco CMTS before cable modems attempt to make a connection. Use the **cable service class** command to configure service classes.

Restrictions for DOCSIS 1.1 Operations

DOCSIS 1.1 operations includes the following restrictions:

Baseline Privacy Interface Plus Requirements

BPI+ encryption and authentication must be supported and enabled by both the cable modem and CMTS. In addition, the cable modem must contain a digital certificate that conforms to the DOCSIS 1.1 and BPI+ specifications.

Also, ensure that the system clocks on the CMTS and on the time-of-day (ToD) servers are synchronized. If this does not occur, the clocks on the CMs will not match the clocks on the CMTS, which could interfere with BPI+ operations. In particular, this could prevent the proper verification of the digital certificates on the CM.

Note Ensure that the system clocks on the CMTS and on the time-of-day (ToD) servers are synchronized. If this does not occur, the clocks on the CMs will not match the clocks on the CMTS, which could interfere with BPI+ operations. In particular, this could prevent the proper verification of the digital certificates on the CM.

BPI+-Encrypted Multicast Not Supported with Bundled Subinterfaces on the Cisco cBR-8 Router

The current Cisco IOS-XE releases do not support using BPI+ encrypted multicast on bundled cable subinterfaces on the Cisco cBR-8 router. Encrypted multicast is supported on bundled cable interfaces or on non-bundled cable subinterfaces, but not when a subinterface is bundled on the Cisco cBR-8 router.

BPI+ Not Supported with High Availability Configurations

The current Cisco IOS-XE releases do not support using BPI+ encrypted multicast on a cable interface when the interface has also been configured for N+1 (1:n) High Availability or Remote Processor Redundancy Plus (RPR+) High Availability redundancy.

In addition, BPI+ is not automatically supported after a switchover from the Working cable interface to the Protect cable interface, because the cable interface configurations that are required for BPI+ encryption are not automatically synchronized between the two interfaces. A workaround for this is to manually configure the Protect cable interfaces with the required configurations.

DOCSIS Root Certificates

The Cisco CMTS supports only one DOCSIS Root CA certificate.

Maximum Burst Size

Previously, the maximum concatenated burst size parameter could be set to zero to specify an unlimited value. In a DOCSIS 1.1 environment, this parameter should be set to a nonzero value, with a maximum value of 1522 bytes for DOCSIS 1.0 cable modems.

If a cable modem attempts to register with a maximum concatenation burst size of zero, the DOCSIS 1.1 CMTS refuses to allow the cable modem to come online. This avoids the possibility that a DOCSIS 1.0 cable modem could interfere with voice traffic on the upstream by sending extremely large data packets. Since DOCSIS 1.0 does not support fragmentation, transmitting such data packets could result in unwanted jitter in the voice traffic.

In addition, DOCSIS 1.1 requires that the maximum transmit burst size be set to either 1522 bytes or the maximum concatenated burst size, whichever is larger. Do not set the maximum concatenation burst size to values larger than 1522 bytes for DOCSIS 1.0 cable modems.

Note This change requires you to change any DOCSIS configuration files that specify a zero value for the maximum concatenation burst size. This limitation does not exist for DOCSIS 1.1 cable modems unless fragmentation has been disabled.

Performance

DOCSIS 1.0 cable modems lack the ability to explicitly request and provide scheduling parameters for advanced DOCSIS 1.1 scheduling mechanisms, such as unsolicited grants and real-time polling. DOCSIS 1.1 cable modems on the same upstream channel can benefit from the advanced scheduling mechanisms and a DOCSIS 1.1 CMTS can still adequately support voice traffic from DOCSIS 1.1 cable modems with DOCSIS 1.0 cable modems on the same upstream channel.

Provisioning

The format and content of the TFTP configuration file for a DOCSIS 1.1 cable modem are significantly different from the file for a DOCSIS 1.0 cable modem. A dual-mode configuration file editor is used to generate a DOCSIS 1.0 style configuration file for DOCSIS 1.0 cable modems and a DOCSIS 1.1 configuration file for DOCSIS 1.1 cable modems.

Registration

A DOCSIS 1.1 CMTS must handle the existing registration Type/Length/Value parameters from DOCSIS 1.0 cable modems as well as the new type TLVs from DOCSIS 1.1 cable modems. A DOCSIS 1.0 and DOCSIS 1.1 cable modem can successfully register with the same DOCSIS 1.1 CMTS.

A DOCSIS 1.1 cable modem can be configured to make an indirect reference to a service class that has been statically defined at the CMTS instead of explicitly asking for the service class parameters. When this registration request is received by a DOCSIS 1.1 CMTS, it encodes the actual parameters of the service class in the registration response and expects a DOCSIS 1.1-specific registration-acknowledge MAC message from the cable modem.

When a DOCSIS 1.0 cable modem registers with a DOCSIS 1.1 CMTS, the registration request explicitly requests all nondefault service-class parameters in the registration. The absence of an indirect service class reference eliminates the need for the DOCSIS 1.1 TLVs and eliminates the need to establish a local registration acknowledge wait state.

When a DOCSIS 1.1 CMTS receives a registration request from a DOCSIS 1.0 cable modem, it responds with the DOCSIS 1.0 style registration response and does not expect the cable modem to send the registration-acknowledge MAC message.

Information about DOCSIS 1.1

DOCSIS 1.1 is the first major revision of the initial DOCSIS 1.0 standard for cable networks. Although the initial standard provided quality data traffic over the coaxial cable network, the demands of real-time traffic such as voice and video required many changes to the DOCSIS specification.

The DOCSIS 1.1 specification provides the following feature enhancements over DOCSIS 1.0 networks:

Baseline Privacy Interface Plus

DOCSIS 1.0 introduced a Baseline Privacy Interface (BPI) to protect user data privacy across the shared-medium cable network and to prevent unauthorized access to DOCSIS-based data transport services across the cable network. BPI encrypts traffic across the RF interface between the cable modem and CMTS, and also includes authentication, authorization, and accounting (AAA) features.

BPI supports access control lists (ACLs), tunnels, filtering, protection against spoofing, and commands to configure source IP filtering on RF subnets to prevent subscribers from using source IP addresses that are not valid. DOCSIS 1.1 enhances these security features with BPI Plus (BPI+), which includes the following enhancements:

- X.509 Digital certificates provide secure user identification and authentication. The Cisco CMTS supports both self-signed manufacturer’s certificates and certificates that are chained to the DOCSIS Root CA certificate.
- Key encryption uses 168-bit Triple DES (3DES) encryption that is suitable for the most sensitive applications.
- 1024-bit public key with Pkcs#1 Version 2.0 encryption.
- Support for encrypted multicast broadcasts, so that only authorized service flows receive a particular multicast broadcast.
- Secure software download allows a service provider to upgrade a cable modem’s software remotely, without the risk of interception, interference, or alteration.

Concatenation

Concatenation allows a cable modem to make a single time-slice request for multiple upstream packets, sending all of the packets in a single large burst on the upstream. Concatenation can send multiple upstream packets as part of one larger MAC data frame, allowing the cable modem to make only one time-slot request for the entire concatenated MAC frame, reducing the delay in transmitting the packets on the upstream channel. This avoids wasting upstream bandwidth when sending a number of very small packets, such as TCP acknowledgement packets.

Dynamic MAC Messages

Dynamic Service MAC messages allow the cable modem to dynamically create service flows on demand. These messages are DOCSIS link layer equivalents of the higher layer messages that create, tear down, and modify a service flow.

The DOCSIS 1.1 dynamic services state machine supports the following messages:

- Dynamic Service Add (DSA)—This message is used to create a new service flow.
- Dynamic Service Change (DSC)—This message is used to change the attributes of an existing service flow.
- Dynamic Service Deletion (DSD)—This message is used to delete an existing service flow.

Note These messages are collectively known as DSX messages.

Enhanced Quality of Service

DOCSIS 1.1 provides enhanced quality of service (QoS) capabilities to give priority for real-time traffic such as voice and video:

- The DOCSIS 1.0 QoS model (a service ID (SID) associated with a QoS profile) has been replaced with a service flow and service class model that allows greater flexibility in assigning QoS parameters to different types of traffic and in responding to changing bandwidth conditions.
- Support for multiple service flows per cable modem allows a single cable modem to support a combination of data, voice, and video traffic.
- Greater granularity in QoS per cable modem in either direction, using unidirectional service flows.
- Upstream service flows can be assigned one of the following QoS scheduling types, depending on the type of traffic and application being used:
    - Best-effort—Data traffic sent on a non-guaranteed best-effort basis. This type of service flow is similar to the method used in DOCSIS 1.0 networks.
    - Real-time polling (rtPS)—Real-time service flows, such as video, that produce unicast, variable size packets at fixed intervals.
    - Non-real-time polling service (nrtPS)—Similar to the rtPS type, in that the cable modem is guaranteed regular opportunities to request data bursts of varying length, except that the CMTS can vary the time between its polling of the cable modem depending on the amount of traffic and congestion on the network.
    - Unsolicited grants (UGS)—Constant bit rate (CBR) or committed information rate (CIR) traffic, such as voice, that is characterized by fixed-size packets at fixed intervals, providing a guaranteed minimum data rate.

- Unsolicited grants with activity detection (USG-AD)—Combination of UGS and rtPS, to accommodate real-time traffic that might have periods of inactivity (such as voice using silence suppression). The service flow uses UGS fixed grants while active, but switches to rtPS polling during periods of inactivity to avoid wasting unused bandwidth.

Fragmentation

DOCSIS fragmentation allows the upstream MAC scheduler to slice large data requests to fit into the scheduling gaps between UGS (voice slots). This prevents large data packets from affecting real-time traffic, such as voice and video.

Fragmentation reduces the run-time jitter experienced by the UGS slots when large data grants preempt the UGS slots. Disabling fragmentation increases the run-time jitter, but also reduces the fragmentation reassembly overhead for fragmented MAC frames.

Note DOCSIS fragmentation should not be confused with the fragmentation of IP packets, which is done to fit the packets on network segments with smaller maximum transmission unit (MTU) size. DOCSIS Fragmentation is Layer 2 fragmentation that is primarily concerned with efficiently transmitting lower-priority packets without interfering with high-priority real-time traffic, such as voice calls. IP fragmentation is done at Layer 3 and is primarily intended to accommodate routers that use different maximum packet sizes.

Interoperability

DOCSIS 1.1 cable modems can coexist with DOCSIS 1.0 and 1.0+ cable modems in the same network. The Cisco CMTS provides the levels of service that are appropriate for each cable modem.

Payload Header Suppression

Payload header suppression (PHS) conserves link-layer bandwidth by suppressing repetitive or redundant packet headers on both upstream and downstream service flows. PHS is enabled or disabled per service flow, and each service flow can support a separate set of PHS rules that determine which parts of the header are suppressed. This ensures that PHS is done in the most efficient manner for each service flow and its particular type of application.

Downstream ToS Overwrite

Downstream ToS Overwrite is supported in DOCSIS 1.1. It can be used in IPv4 and IPv6 environment. You can use CLI command **cable service class** *class-index* **tos-overwrite** *and-mask or-mask* or the cable modem configuration file to configure downstream ToS overwrite.

DOCSIS 1.1 Quality of Service

The DOCSIS 1.1 QoS framework is based on the following objects:

- Service flow—A unidirectional sequence of packets on the DOCSIS link. Separate service flows are used for upstream and downstream traffic, and define the QoS parameters for that traffic.

- Service class—A collection of settings maintained by the CMTS that provide a specific QoS service tier to a cable modem that has been assigned a service flow associated with that service class.
- Packet classifier—A set of packet header fields used to classify packets onto a service flow to which the classifier belongs. The CMTS uses the packet classifiers to match the packet to the appropriate service flow.
- Payload header suppression (PHS) rule—A set of packet header fields that are suppressed by the sending entity before transmitting on the link, and are restored by the receiving entity after receiving a header-suppressed frame transmission. PHS increases the bandwidth efficiency by removing repeated packet headers before transmission.

See the following sections for more information on these components.

Service Flow

In DOCSIS 1.1, the basic unit of QoS is the service flow, which is a unidirectional sequence of packets transported across the RF interface between the cable modem and CMTS. A service flow defines a set of QoS parameters such as latency, jitter, and throughput assurances, and these parameters can be applied independently to the upstream and downstream traffic flows. This is a major difference from DOCSIS 1.0 networks, where the same QoS parameters were applied to both the downstream and upstream flows.

Note DOCSIS 1.0 networks used service IDs (SIDs) to identify the QoS parameter set for a particular flow. DOCSIS 1.1 networks use the service flow ID (SFID) to identify the service flows that have been assigned to a particular upstream or downstream. DOCSIS 1.1 networks still use the term SID, but it applies exclusively to upstream service flows.

Every cable modem establishes primary service flows for the upstream and downstream directions, with a separate SFID for the upstream and the downstream flows. The primary flows maintain connectivity between the cable modem and CMTS, allowing the CMTS to send MAC management messages at all times to the cable modem.

In addition, a DOCSIS 1.1 cable modem can establish multiple secondary service flows. The secondary service flows either can be permanently created (by configuring them in the DOCSIS configuration file that is downloaded to the cable modem), or the service flows can be created dynamically to meet the needs of the on-demand traffic, such as voice calls. Permanent service flows remain in effect, even if they are not being used, while dynamic service flows are deleted when they are no longer needed.

At any given time, a service flow might be in one of three states (provisioned, admitted, or active). Only active flows are allowed to pass traffic on the DOCSIS network. Every service flow is identified by an SFID, while upstream service flows in the admitted and active state have an extra Layer 2 SID associated with them. The SID is the identifier used by the MAC scheduler when specifying time-slot scheduling for different service flows.

Service Class

Each service flow is associated with a service class, which defines a particular class of service and its QoS characteristics, such as the maximum bandwidth for the service flow and the priority of its traffic. The service class attributes can be inherited from a preconfigured CMTS local service class (class-based flows), or they can be individually specified when a cable modem dynamically requests a service flow and the CMTS creates it.

The DOCSIS 1.1 service class also defines the MAC-layer scheduling type for the service flow. The schedule type defines the type of data burst requests that the cable modem can make, and how often it can make those requests. The following types of schedule types are supported:

- Best-effort (BE)—A cable modem competes with the other cable modems in making bandwidth requests and must wait for the CMTS to grant those requests before transmitting data. This type of service flow is similar to the method used in DOCSIS 1.0 networks.
- Real-time polling service (rtPS)—A cable modem is given a periodic time slot in which it can make bandwidth requests without competing with other cable modems. This allows real-time transmissions with data bursts of varying length.
- Non-real-time polling service (nrtPS)—A cable modem is given regular opportunities to make bandwidth requests for data bursts of varying size. This type of flow is similar to the rtPS type, in that the cable modem is guaranteed regular opportunities to request data bursts of varying length, except that the CMTS can vary the time between its polling of the cable modem, depending on the amount of traffic and congestion on the network.
- Unsolicited grant service (UGS)—A cable modem can transmit fixed data bursts at a guaranteed minimum data rate and with a guaranteed maximum level of jitter. This type of service flow is suitable for traffic that requires a Committed Information Rate (CIR), such as Voice-over-IP (VoIP) calls.
- Unsolicited grant service with activity detection (UGS-AD)—Similar to the UGS type, except that the CMTS monitors the traffic to detect when the cable modem is not using the service flow (such as voice calls when nobody is speaking). When the CMTS detects silence on the service flow, the CMTS temporarily switches the service flow to an rtPS type. When the cable modem begins using the flow again, the CMTS switches the flow back to the UGS type. This allows the CMTS to more efficiently support VoIP calls.

Each service flow is assigned a single service class, but the same service class can be assigned to multiple service flows. Also, a cable modem can be assigned multiple service flows, allowing it to have multiple traffic flows that use different service classes.

Packet Classifiers

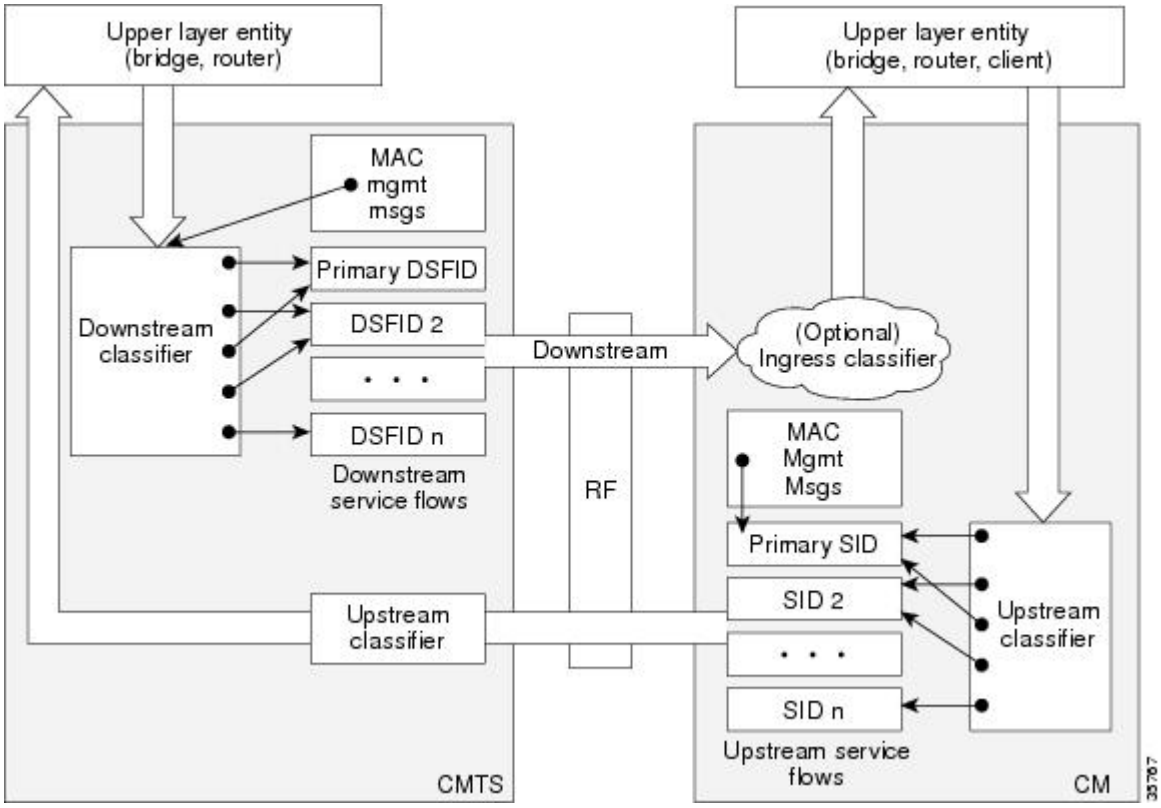
In DOCSIS 1.0 networks, a cable modem used only one set of QoS parameters for all of its traffic, so the CMTS simply had to route packets to and from the appropriate cable modems. In DOCSIS 1.1 networks, however, cable modems can be using multiple service flows, and each service flow can be given a different level of service. To quickly assign upstream and downstream packets to their proper service flows, the CMTS uses the concept of packet classifiers.

Each packet classifier specifies one or more packet header attributes, such as source MAC address, destination IP address, or protocol type. The classifier also specifies the service flow to be used when a packet matches this particular combination of headers. Separate classifiers are used for downstream and upstream service flows.

When the CMTS receives downstream and upstream packets, it compares each packet's headers to the contents of each packet classifier. When the CMTS matches the packet to a classifier, the CMTS then assigns the proper SFID to the packet and transmits the packet to or from the cable modem. This ensures that the packet is assigned its proper service flow, and thus its proper QoS parameters.

Figure below illustrates the mapping of packet classifiers.

Figure 32: Classification Within the MAC Layer



Packet Header Suppression Rules

Because many data and real-time applications may use fixed values in their packet header fields, DOCSIS 1.1 supports PHS to suppress the duplicate portions of the packet headers when a group of packets is transmitted during a session. Each service flow can support a separate set of PHS rules that determine which parts of the header are suppressed.

When PHS is being used, the transmitting CMTS suppresses the specified headers in all the packets for that service flow. The receiving CMTS then restores the missing headers before forwarding the packets on to their ultimate destination.

Proper use of PHS can increase the efficiency of packetized transmissions, especially for real-time data that is encapsulated by other protocols, such as VoIP traffic.

Quality of Service Comparison

This section summarizes the differences in QoS between DOCSIS 1.0, DOCSIS 1.0+, and DOCSIS 1.1 networks.

Note Cisco CMTS routers can transparently interoperate with cable modems running DOCSIS 1.0, DOCSIS 1.0+ extensions, or DOCSIS 1.1. If a cable modem indicates at system initialization that it is DOCSIS 1.1-capable, the Cisco CMTS router uses the DOCSIS 1.1 features. If the cable modem is not DOCSIS 1.1-capable, but does support the DOCSIS 1.0+ QoS extensions, the Cisco CMTS automatically supports the cable modem's requests for dynamic services. Otherwise, the cable modem is treated as a DOCSIS 1.0 device.

DOCSIS 1.0

DOCSIS1.0 uses a static QoS model that is based on a class of service (CoS) that is preprovisioned in the DOCSIS configuration file that is downloaded to the cable modem. The CoS is a bidirectional QoS profile that applies to both the upstream and downstream directions, and that has limited control, such as peak rate limits in either direction, and relative priority on the upstream.

DOCSIS 1.0 defines the concept of a service identifier (SID), which identifies the cable modems that are allowed to transmit on the network. In DOCSIS 1.0 networks, each cable modem is assigned only one SID for both the upstream and downstream directions, creating a one-to-one correspondence between a cable modem and its SID. All traffic originating from, or destined for, a cable modem is mapped to that particular SID.

Typically, a DOCSIS 1.0 cable modem has one CoS and treats all traffic the same, which means that data traffic on a cable modem can interfere with the quality of a voice call in progress. The CMTS, however, has a limited ability to prioritize downstream traffic based on IP precedent type-of-service (ToS) bits.

For example, voice calls using higher IP precedence bits receive a higher queueing priority (but without a guaranteed bandwidth or rate of service). A DOCSIS 1.0 cable modem could increase voice call quality by permanently reserving bandwidth for voice calls, but then that bandwidth would be wasted whenever a voice call is not in progress.

DOCSIS 1.0+

In response to the limitations of DOCSIS 1.0 networks in handling real-time traffic, such as voice calls, Cisco created the DOCSIS 1.0+ extensions to provide the more important QoS enhancements that were expected in DOCSIS 1.1. In particular, the DOCSIS 1.0+ enhancements provide basic Voice-over-IP (VoIP) service over the DOCSIS link.

Cisco's DOCSIS 1.0+ extensions include the following DOCSIS 1.1 features:

- Multiple SIDs per cable modem, creating separate service flows for voice and data traffic. This allows the CMTS and cable modem to give higher priority for voice traffic, preventing the data traffic from affecting the quality of the voice calls.
- Cable modem-initiated dynamic MAC messages—Dynamic Service Addition (DSA) and Dynamic Service Deletion (DSD). These messages allow dynamic SIDs to be created and deleted on demand, so that the bandwidth required for a voice call can be allocated at the time a call is placed and then freed up for other uses when the call is over.
- Unsolicited grant service (CBR-scheduling) on the upstream—This helps provide a higher-quality channel for upstream VoIP packets from an Integrated Telephony Cable Modem (ITCM) such as the Cisco uBR925 cable access router.
- Ability to provide separate downstream rates for any given cable modem, based on the IP-precedence value in the packet. This helps separate voice signaling and data traffic that goes to the same ITCM to address rate shaping purposes.

- Concatenation allows a cable modem to send several packets in one large burst, instead of having to make a separate grant request for each.

Caution All DOCSIS 1.0 extensions are available only when using a cable modem and CMTS that supports these extensions. The cable modem activates the use of the extensions by sending a dynamic MAC message. DOCSIS 1.0 cable modems continue to receive DOCSIS 1.0 treatment from the CMTS.

Interoperability with Different Versions of DOCSIS Networks

DOCSIS 1.1 cable modems have additional features and better performance than earlier DOCSIS 1.0 and 1.0+ models, but all three models can coexist in the same network. DOCSIS 1.0 and 1.0+ cable modems will not hamper the performance of a DOCSIS 1.1 CMTS, nor will they interfere with operation of DOCSIS 1.1 features.

Table below shows the interoperability of a DOCSIS 1.1 CMTS with different versions of cable modems.

Table 167: DOCSIS 1.1 Interoperability

For this configuration...	The result is...
DOCSIS 1.1 CMTS with DOCSIS 1.0 cable modems	DOCSIS 1.0 cable modems receive DOCSIS 1.0 features and capabilities. BPI is supported if available and enabled on the CMTS.
DOCSIS 1.1 CMTS with DOCSIS 1.0+ cable modems	DOCSIS 1.0+ cable modems receive basic DOCSIS 1.0 support. BPI is supported if available and enabled on the CMTS. In addition, DOCSIS 1.0+ cable modems also receive the following DOCSIS 1.1 features: • Multiple SIDs per cable modem • Dynamic service MAC messaging initiated by the cable modem • Unsolicited grant service (UGS, CBR-scheduling) on the upstream • Separate downstream rates for any given cable modem, based on the IP-precedence value • Concatenation
DOCSIS 1.1 CMTS with DOCSIS 1.1 cable modems	DOCSIS 1.1 cable modems receive all the DOCSIS 1.1 features listed in this document. BPI+ is supported if available and enabled on the CMTS.

Enhanced Rate Bandwidth Allocation (ERBA) Support for DOCSIS 1.0 Cable Modems

To define ERBA on the downstream for DOCSIS 1.0 cable modems, use the cable qos promax-ds-burst command in global configuration mode.

The ERBA feature is characterized by the following enhancements:

- Enables support for the DOCSIS1.1 Downstream Maximum Transmit Burst parameter on the Cisco CMTS by using the **cable ds-max-burst** configuration command.
- Allows DOCSIS1.0 modems to support the DOCSIS1.1 Downstream Maximum Transmit Burst parameter by mapping DOCSIS1.0 modems to overriding DOCSIS 1.1 QoS profile parameters on the Cisco CMTS.

ERBA allows DOCSIS1.0 modems to burst their temporary transmission rate up to the full line rate for short durations of time. This capability provides higher bandwidth for instantaneous bandwidth requests, such as those in Internet downloads, without having to make changes to existing service levels in the QoS Profile.

This feature allows you to set the DOCSIS 1.0 cable modems burst transmissions, with mapping to overriding DOCSIS 1.1 QoS profile parameters on the Cisco CMTS. DOCSIS 1.0 cable modems require DOCSIS 1.0 parameters when registering to a matching QoS profile. This feature enables maximum downstream line rates, and the ERBA setting applies to all cable modems that register to the corresponding QoS profile.

Note QoS definitions must previously exist on the Cisco CMTS headend to support this feature.

ERBA for DOCSIS 1.0 cable modems is supported with these new or enhanced commands or keywords:

- cable qos pro max-ds-burst burst-size
- show cable qos profile n [verbose]

DOCSIS 3.0 Downstream Peak Traffic Rate TLV Support for ERBA

The DOCSIS WFQ Scheduler allows each service flow to have one dedicated queue. When ERBA is enabled for the service flow, the peak rate is implemented as the queue shape rate within the scheduler, while the maximum sustained rate is set as the token bucket refill rate. When ERBA is turned off, the burst size and the peak rate value are not used.

The maximum traffic burst parameter is used to control a service flow burst duration, to burst up to the channel line rate or a configured peak rate, when it is within its maximum burst size allowance. On the Cisco cBR-8 Converged Broadband Router, the **cable ds-max-burst** command is used to control this behavior explicitly.

The *peak-rate* keyword is introduced to specify the peak rate an ERBA-enabled service flow can use. The peak rate value is applied to a specific service flow created after the configuration of the **cable ds-max-burst** command.

If the DOCSIS 3.0 TLV 25.27 is specified for a service flow, the *peak rate* value is set as the TLV value. However, if ERBA is not turned on for a service flow, the *peak rate* value is ignored.

During modem registration or Dynamic Service Addition (DSA) operation, the service class name TLV 25.4 is sent to create the static or dynamic downstream service flow that matches the service class template. These downstream service flows are created with a specific *peak rate*.

Some of the DOCSIS 1.x an DOCSIS 2.0 cable modems, which are not fully DOCSIS 1.x or DOCSIS 2.0 compliant, may fail to come online when the downstream peak rate TLV 25.27 is received from the CMTS during registration. To overcome this failure, you can configure the cable service attribute withhold-TLVs command to restrict sending of the peak traffic rate TLVs to DOCSIS1.x and DOCSIS 2.0 cable modems. For more information on how to suppress peak rate TLVs, see Suppressing Upstream and Downstream Peak Rate TLVs for pre DOCSIS 3.0 Cable Modems, on page 1036.

Note The ERBA feature is not applicable for high priority service flows and multicast service flows.

Table below summarizes the ERBA support for the Cisco cBR-8 router.

Table 168: Enhanced Rate Bandwidth Allocation Support for the Cisco cBR-8 Router

	Policer Rate	Policer Exceed Action	Policer Token Bucket Size	Queue Shape Rate
Traditional Service Flow	Maximum Sustained Traffic Rate (unused)	Transmit	A value computed internally by CMTS (unused)	Maximum Sustained Traffic Rate
ERBA-Enabled Service Flow	Maximum Sustained Traffic Rate	Drop	Maximum Traffic Burst TLV	Peak Traffic Rate

In Cisco cBR-8 routers, the dual token bucket-based shaper is used to support ERBA on the Cisco cBR-8 CCAP line card (the ERBA feature is always enabled on the Cisco cBR-8 CCAP line card). The dual token bucket shaper has two independent token buckets for each service flow. The maximum rate of one bucket is configured to MSR and the maximum tokens are set to maximum traffic burst. The other bucket is configured with the refilling rate of the *peak rate* and the maximum tokens are set to the default level of 4 milliseconds. Packets are shaped if any of the two buckets are exhausted.

Table below summarizes the ERBA dual token bucket configuration for the Cisco cBR-8 routers.

Table 169: ERBA Dual Token Bucket Configuration

	Token Bucket Rate (One)	Token Bucket Size (One)	Token Bucket Rate (Two)	Token Bucket Size (Two)
Traditional Service Flow	Maximum Sustained Traffic Rate	4ms * MSR	N/A	N/A
ERBA-enabled Service Flow	Maximum Sustained Traffic Rate	Maximum Traffic Burst or 4ms * MSR	Peak Rate	4ms * Peak Rate

Suppressing Upstream and Downstream Peak Rate TLVs for pre DOCSIS 3.0 Cable Modems

The DOCSIS 3.0 upstream (US) peak rate TLV 24.27 and downstream (DS) peak rate TLV 25.27 are enabled on the Cisco CMTS through the cable service class command or the CM configuration file. The DOCSIS 1.x and DOCSIS 2.0 CMs do not support these TLVs. Ideally, if a DOCSIS 1.x or DOCSIS 2.0 CM receives peak rate TLVs during registration, it should ignore these TLVs and proceed with the registration. However there are a few old non-compliant pre DOCSIS 3.0 CMs, which may fail to come online when peak-rate TLVs are received in the registration response from the Cisco CMTS. To overcome this, the Cisco CMTS enables suppression of the DOCSIS 3.0 peak rate TLVs for the pre-DOCSIS3.0 CMs.

To suppress the DOCSIS 3.0 US and DS peak rate TLVs, use the **cable service attribute withhold-TLVs command with the peak-rate** keyword in global configuration mode. When configured, this command restricts the Cisco CMTS from sending US and DS peak rate TLVs to the DOCSIS 1.x and DOCSIS 2.0 CMs. The decision to send the TLVs is based on the DOCSIS version of the CM received during registration. If the registration request is from a pre DOCSIS 3.0 CM, the peak rate TLVs are not sent in the registration response. However this command does not restrict sending of DOCSIS 3.0 peak-rate TLVs to DOCSIS 3.0 CMs.

Downstream Classification Enhancement with MAC Addresses

Downstream classifiers, specified in the cable modem configuration file, are used to map packets to service flows based on DOCSIS specifications. New combinations of downstream classifiers with a destination MAC address are supported. This enhancement enables service providers to better manage high priority service flows associated with a downstream classifier. For example, a single User Datagram Protocol (UDP) port can be shared by high priority and low priority traffic.

Downstream classification is automatically enabled on the Cisco CMTS router. The downstream classifier combinations that are supported on the router are listed below:

Without Combination

- IP (IPv4)
- IPv6
- TCP/UDP
- Destination MAC

With Combination

- IPv4 + TCP/UDP
- IPv6 + TCP/UDP
- Destination MAC + IPv4 (with the exception of a destination IP address)
- Destination MAC + IPv6 (with the exception of a destination IPv6 address)
- Destination MAC + TCP/UDP
- Destination MAC + IPv4 + TCP/UDP (with the exception of a destination IP address)
- Destination MAC + IPv6 + TCP/UDP (with the exception of a destination IPv6 address)

Benefits

DOCSIS 1.1 includes a rich set of features that provide advanced and flexible QoS capabilities for various types of traffic (voice, data, and video) over the cable network. It also provides enhanced security and authentication features.

Baseline Privacy Interface Plus Enhancement

The Plus (+) version of the Baseline Privacy Interface (BPI+) in DOCSIS 1.1 provides a set of extended services within the MAC sublayer that increase performance and system security. Digital certificates provide secure authentication for each cable modem, to prevent identity theft on the basis of MAC and IP addresses. Advanced encryption provides a secure channel between the cable modem and CMTS, and secure software download allows a service provider to upgrade the software on cable modems, without the threat of interception, interference, or alteration of the software code.

Dynamic Service Flows

The dynamic creation, modification, and deletion of service flows allows for on-demand reservation on Layer 2 bandwidth resources. The CMTS can now provide special QoS to the cable modem dynamically for the duration of a voice call or video session, as opposed to the static provisioning and reservation of resources at the time of cable modem registration. This provides a more efficient use of the available bandwidth.

Concatenation

The cable modem concatenates multiple upstream packets into one larger MAC data frame, allowing the cable modem to make only one time-slot request for the entire concatenated MAC frame, as opposed to requesting a time slot for each packet. This reduces the delay in transferring the packet burst upstream.

Enhanced QoS

Extensive scheduling parameters allow the CMTS and the cable modem to communicate QoS requirements and achieve more sophisticated QoS on a per service-flow level.

Different new time-slot scheduling disciplines help in providing guaranteed delay and jitter bound on shared upstream. Activity detection helps to conserve link bandwidth by not issuing time slots for an inactive service flow. The conserved bandwidth can then be reused for other best-effort data slots.

Packet classification helps the CMTS and cable modem to isolate different types of traffic into different DOCSIS service flows. Each flow could be receiving a different QoS service from CMTS.

Fragmentation

Fragmentation splits large data packets so that they fit into the smaller time slots inbetween UGS slots. This reduces the jitter experienced by voice packets when large data packets are transmitted on the shared upstream channel and preempt the UGS slots used for voice.

Multiple Subflows per SID

This feature allows the cable modem to have multiple calls on a single hardware queue. This approach scales much better than requiring a separate SID hardware queue on the cable modem for each voice call.

Payload Header Suppression

Payload Header Suppression (PHS) allows the CMTS and cable modem to suppress repetitive or redundant portions in packet headers before transmitting on the DOCSIS link. This conserves link bandwidth, especially with types of traffic such as voice, where the header size tends to be as large as the size of the actual packet.

Service Classes

The use of the service class provides the following benefits for a DOCSIS 1.1 network:

- It allows operators to move the burden of configuring service flows from the provisioning server to the CMTS. Operators provision the modems with the service class name; the implementation of the name is configured at the CMTS. This allows operators to modify the implementation of a given service to local circumstances without changing modem provisioning. For example, some scheduling parameters might need to be set differently for two different CMTSs to provide the same service. As another example, service profiles could be changed by time of day.
- It allows CMTS vendors to provide class-based-queuing if they choose, where service flows compete within their class and classes compete with each other for bandwidth.
- It allows higher-layer protocols to create a service flow by its service class name. For example, telephony signaling might direct the cable modem to instantiate any available provisioned service flow of class G.711.

Note The service class is optional. The flow scheduling specification may always be provided in full; a service flow may belong to no service class whatsoever. CMTS implementations *may* treat such unclassed flows differently from classed flows with equivalent parameters.

How to Configure the Cisco CMTS for DOCSIS 1.1 Operations

See the following sections for the configuration tasks for DOCSIS 1.1 operations. Each task in the list is identified as either required or optional.

Note This section describes only the configuration tasks that are specific for DOCSIS 1.1 operations. For complete configuration information, see the software configuration documents listed in the Additional References, on page 1060.

Configuring Baseline Privacy Interface

BPI+ encryption is by default enabled for 56-bit DES encryption on all cable interfaces. If BPI+ encryption has been previously disabled, or if you want to reconfigure BPI+ encryption on a cable interface on the CMTS, use the following procedure.

Note If you have disabled BPI+ encryption on a cable interface, and a cable modem attempts to register on that interface using BPI+ encryption, the CMTS will reject its registration request, displaying a %CBR-4-SERVICE_PERMANENTLY_UNAVAILABLE error message. The **show cable modem** command will also show that this cable modem has been rejected with a MAC status of reject(c).

Before you begin

BPI+ encryption is supported on all Cisco CMTS images that include “k1”, “k8”, or “k9” in its file name or BPI in the feature set description. All BPI images support 40-bit and 56-bit DES encryption.

By default, BPI+ encryption is enabled for 56-bit DES encryption. Also, when a cable modem is running DOCSIS 1.1 software, BPI+ encryption is enabled by default, unless the service provider has disabled it by setting the Privacy Enable field (TLV 29) in the DOCSIS configuration file to 0. Therefore, both the CMTS and cable modem are set to use BPI+ encryption when using the default configurations.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable` `Router#`	

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**interface cable***slot* /*subslot* /*port*
	Example:
	`Router(config)# interface cable 6/0/0`
	`Router(config-if)#`
		Enters interface configuration mode for the cable interface line card at this particular slot.

Step 4	**cable privacy**
	Example:
	`Router(config-if)# cable privacy`
	`Router(config-if)#`
		(Optional) Enables BPI+ 56-bit DES encryption on the cable interface (default).

Step 5	**cable privacyaccept-self-signed-certificate**
	Example:
	`Router(config-if)# cable privacy accept-self-signed-certificate`
	`Router(config-if)#`
		(Optional) Allows cable modems to register using self-signed manufacturer certificates, as opposed to the default of allowing only manufacturer's certificates that are chained to the DOCSIS root certificate.
		Caution Use the above command sparingly, as it bypasses DOCSIS BPI+ certificates. Otherwise, self-signed certificates provide workaround registration for cable modems that are not compliant with DOCSIS BPI+ certificates. This functionality is strictly intended for troubleshooting of a short duration or in the context of additional security measures.
		Note By default, the CMTS does not accept self-signed certificates. In the default configuration, if a cable modem attempts to register with self-signed certificates, the CMTS will refuse to allow the cable modem to register.

	Command or Action	Purpose

Step 6	**cable privacy authorize-multicast**
	Example:
	`Router(config-if)# cable privacy authorize-multicast`
	`Router(config-if)#`
		(Optional) Enables BPI+ encryption on the cable interface and uses AAA protocols to authorize all multicast stream (IGMP) join requests.
		Note If you use this command to authorize multicast streams, you must also use the **cable privacy authenticate-modem** command to enable AAA services on the cable interface.

Step 7	**cable privacy mandatory**
	Example:
	`Router(config-if)# cable privacy mandatory`
	`Router(config-if)#`
		(Optional) Requires baseline privacy be active for all CMs with BPI/BPI+ enabled in the DOCSIS configuration files, else the CMs are forced to go offline.
		If a CM does not have BPI enabled in its DOCSIS configuration file, it will be allowed to come online without BPI.

Step 8	**cable privacy oaep-support**
	Example:
	`Router(config-if)# cable privacy oaep-support`
	`Router(config-if)#`
		(Optional) Enables BPI+ encryption on the cable interface and enables Optimal Asymmetric Encryption Padding (OAEP). This option is enabled by default. Disabling this option could have a performance impact.

Step 9	**cable privacy kek** {**life-time** *seconds* }
	Example:
	`Router(config-if)# cable privacy kek life-time 302400`
	`Router(config-if)#`
		(Optional) Configures the life-time values for the key encryption keys (KEKs) for BPI+ operations on all cable interfaces.

Step 10	**cable privacy tek** {**life-time** *seconds*}
	Example:
	`Router(config-if)# cable privacy tek life-time 86400`
	`Router(config-if)#`
		(Optional) Configures the life-time values for the traffic encryption keys (TEKs) for BPI+ operations on all cable interfaces.

Step 11	**exit**
	Example:
	`Router(config-if)# exit`
	`Router(config)#`
		Exits interface configuration mode.
		Note Repeat steps Step 3, on page 1040 through Step 11, on page 1041 for each cable interface.

	Command or Action	Purpose

Step 12	**exit**
	Example:
	`Router(config)# exit`
	`Router#`
		Exits global configuration mode.

What to do next

You can also configure the following additional timers for BPI+ operations in the DOCSIS configuration file for each cable modem. As a general rule, you do not need to specify these timers in the DOCSIS configuration file unless you have a specific reason for changing them from their default values.

Table 170: Individual Cable Modem BPI+ Timer Values

Timer	Description
Authorize Wait Timeout	The amount of time a cable modem will wait for a response from a CMTS when negotiating a KEK for the first time.
Reauthorize Wait Timeout	The amount of time a cable modem will wait for a response from a CMTS when negotiating a new KEK because the Authorization Key (KEK) lifetime is about to expire.
Authorize Reject Wait Timeout	The amount of time a cable modem must wait before attempting to negotiate a new KEK if the CMTS rejects its first attempt to negotiate a KEK.
Operational Wait Timeout	The amount of time a cable modem will wait for a response from a CMTS when negotiating a TEK for the first time.
Rekey Wait Timeout	The amount of time a cable modem will wait for a response from a CMTS when negotiating a new TEK because the TEK lifetime is about to expire.

Downloading the DOCSIS Root Certificate to the CMTS

DOCSIS 1.1 allows cable modems to identify themselves using a manufacturer's chained X.509 digital certificate that is chained to the DOCSIS root certificate. The DOCSIS root certificate is already installed on the bootflash of the CMTS router. However, if you want to install another root certificate, for example, the Euro-DOCSIS certificate, download the certificate and save it on the bootflash as "euro-root-cert".

Tip For more information about the DOCSIS root certificate provided by Verisign, see the information at the following URL: http://www.verisign.com/products-services/index.html

Note You may load the DOCSIS root certificate and a EuroDOCSIS or PacketCable root certificate. Cisco recommends that the EuroDOCSIS PacketCable root certificates be copied into bootflash.

To download the DOCSIS root certificate to the Cisco CMTS, which is required if any cable modems on the network are using chained certificates, use the following procedure:

Procedure

Step 1 Download the DOCSIS root certificate from the DOCSIS certificate signer, Verisign. At the time of this document's printing, the DOCSIS root certificate is available for download at the following URL: http://www.verisign.com/products-services/index.html

Step 2 Verisign distributes the DOCSIS root certificate in a compressed ZIP archive file. Extract the DOCSIS root certificate from the archive and copy the certificate to a TFTP server that the CMTS can access.

Tip To avoid possible confusion with other certificates, keep the file's original filename of "CableLabs_DOCSIS.509" when saving it to the TFTP server.

Step 3 Log in to the Cisco CMTS using either a serial port connection or a Telnet connection. Enter the **enable** command and password to enter Privileged EXEC mode:

Example:

```
Router> enable

Password: <password>
Router#
```

Step 4 Use the **dir bootflash** command to verify that the bootflash has sufficient space for the DOCSIS root certificate (approximately 1,000 bytes of disk space):

Example:

```
Router# dir bootflash:

Directory of bootflash:/
    1  -rw-     3229188   Dec 30 2002 15:53:23
cbrsup-universalk9.2015-03-18_03.30_johuynh.SSA.bin
3407872 bytes total (250824 bytes free)
Router#
```

Tip If you delete files from the bootflash to make room for the DOCSIS root certificate, remember to use the **squeeze** command to reclaim the free space from the deleted files.

Step 5 Use the **copy tftp bootflash** command to copy the DOCSIS root certificate to the router's bootflash memory. (The file must be named "root-cert" on the bootflash for the CMTS to recognize it as the root certificate.)

Example:

```
Router# copy tftp bootflash:

Address or name of remote host []? tftp-server-ip-address

Source filename []? CableLabs_DOCSIS.509

Destination filename [CableLabs_DOCSIS.509]? root-cert

Loading CableLabs_DOCSIS.509 from tftp-server-ip-address (via FastEthernet0/0): !
[OK - 996/1024 bytes]
```

```
996 bytes copied in 4.104 secs (249 bytes/sec)
Router#
```

Tip You can also copy the root certificate to a PCMCIA Flash Disk (disk0 or disk1). However, because Flash Disks are not secure and easily removed from the router, we recommend that you keep the root certificate in the bootflash for both operational and security reasons.

Step 6 Verify that the DOCSIS root certificate has been successfully copied to the bootflash memory:

Example:

```
Router# dir bootflash:

Directory of bootflash:/
    1  -rw-     3229188   Dec 30 2002 15:53:23
cbrsup-universalk9.2015-03-18_03.30_johuynh.SSA.bin
    2  -rw-         996   Mar 06 2002 16:03:46  root-cert
3408876 bytes total (248696 zxbytes free)
Router#
```

Step 7 (Optional) After the first cable modem has registered using BPI+, you can use the **show crypto ca trustpoints** command to display the Root certificate that the CMTS has learned:

Note The **show crypto ca trustpoints** command does not display the root certificate until after at least one cable modem has registered with the CMTS using BPI+ encryption. Alternatively, you can use the unsupported command **test cable generate** in privileged EXEC mode to force the CMTS to register the root certificate.

Example:

```
Router# show crypto ca trustpoints
Root certificate
  Status: Available
  Certificate Serial Number: D54BB68FE934324F6B8FD0E41A65D867
  Key Usage: General Purpose
  Issuer:
    CN = DOCSIS Cable Modem Root Certificate Authority
     OU = Cable Modems
     O = Data Over Cable Service Interface Specifications
     C = US
  Subject Name:
    CN = "BPI Cable Modem Root Certificate Authority "
     OU = DOCSIS
     O = BPI
     C = US
  Validity Date:
    start date: 07:00:00 UTC Mar 27 2001
    end   date: 06:59:59 UTC Jan 1 2007
```

What to do next

Tip To display all certificates (Root, Manufacturers, CM) that the CMTS has learned, use the **show crypto ca certificates** command.

Adding a Manufacturer's Certificate as a Trusted Certificate

The DOCSIS specifications allow operators to control which manufacturer's and CM certificates are allowed on each CMTS by marking them as either trusted or untrusted. You can add a certificate to the list of trusted certificates on the Cisco CMTS using SNMP commands, as described in the following section:

Adding a Certificate as a Trusted Certificate Using SNMP Commands

You can also use an SNMP manager to create and add certificates to the CMTS list of trusted certificates by manipulating the tables and attributes in the DOCS-BPI-PLUS-MIB . To add a manufacturer's certificate, add an entry to the docsBpi2CmtsCACertTable table. Specify the following attributes for each entry:

- docsBpi2CmtsCACertStatus—Set to 4 to create the row entry.
- docsBpi2CmtsCACert—The hexadecimal data, as an X509Certificate value, for the actual X.509 certificate.
- docsBpi2CmtsCACertTrust—An Integer value from 1 to 4 specifying the certificate's trust status: 1=trusted, 2=untrusted, 3= chained, 4=root. Specify 1 for certificates that should be trusted and 3 for chained certificates that should be verified with the root certificate.

Similarly, to add a CM certificate to the list of trusted certificates, add an entry to the docsBpi2CmtsProvisionedCmCertTable table. Specify the following attributes for each entry:

- docsBpi2CmtsProvisionedCmCertStatus—Set to 4 to create the row entry.
- docsBpi2CmtsProvisionedCmCert—The hexadecimal data, as an X509Certificate value, for the actual X.509 certificate.
- docsBpi2CmtsProvisionedCmCertTrust—An Integer value from 1 to 2 specifying the certificate's trust status: 1=trusted, 2=untrusted. Specify 1 for CM certificates that should be trusted.

Tip Always set the CertStatus attributes before loading the actual certificate data, because otherwise the CMTS will assume the certificate is chained and will immediately attempt to verify it with the manufacturers and root certificates.

For example, to use the Unix command-line SNMP utility to add a manufacturer's certificate to the list of trusted certificates on the CMTS at IP address 192.168.100.134, enter the following command (be sure to substitute a valid index pointer for the table entry for the <*index*> value).

```
% setany -v2c 192.168.100.134 private docsBpi2CmtsCACertStatus.
<index>
 -i 4
 docsBpi2CmtsCACert.
<index>
 -o
 '<hex_data>' docsBpi2CmtsCACertTrust.
<index>
 -i 1
```

To do the same thing for a CM certificate, use the following command:

```
% setany -v2c 192.168.100.134 private docsBpi2CmtsProvisionedCmCertStatus.
<index>
 -i 4 docsBpi2CmtsProvisionedCmCert.
<index>
 -o
```

```
 '<hex_data>' docsBpi2CmtsProvisionedCmCertTrust.
<index>
 -i 1
```

Tip Most operating systems cannot accept input lines that are as long as needed to input the hexadecimal decimal string that specifies a certificate. For this reason, you should use a graphical SNMP manager to set these attributes. For a number of certificates, you can also use a script file, if more convenient.

Note If you are adding self-signed certificates, you must also use the **cable privacy accept-self-signed-certificate** command before the CMTS will accept the certificates.

Adding a Manufacturer's or CM Certificate to the Hotlist

The DOCSIS specifications allow operators to add a digital manufacturer's or CM certificate to a hotlist (also known as the certificate revocation list, or CRL) on the CMTS, to indicate that this particular certificate should no longer be accepted. This might be done when a user reports that their cable modem has been stolen, or when the service provider decides not to support a particular manufacturer's brand of cable modems.

Adding a Certificate to the Hotlist Using SNMP Commands

You can also use an SNMP manager to create and add certificates to the hotlist by manipulating the tables and attributes in the DOCS-BPI-PLUS-MIB . To add a manufacturer's certificate, add an entry to the docsBpi2CmtsCACertTable table. Specify the following attributes for each entry:

- docsBpi2CmtsCACertStatus—Set to 4 to create the row entry.
- docsBpi2CmtsCACert—The hexadecimal data, as an X509Certificate value, for the actual X.509 certificate.
- docsBpi2CmtsCACertTrust—An Integer value from 1 to 4 specifying the certificate's trust status: 1=trusted, 2=untrusted, 3= chained, 4=root. When adding a certificate to the hotlist, set this attribute to 2 for untrusted.

Similarly, to add a CM certificate to the hotlist, add an entry to the docsBpi2CmtsProvisionedCmCertTable table. Specify the following attributes for each entry:

- docsBpi2CmtsProvisionedCmCertStatus—Set to 4 to create the row entry.
- docsBpi2CmtsProvisionedCmCert—The hexadecimal data, as an X509Certificate value, for the actual X.509 certificate.
- docsBpi2CmtsProvisionedCmCertTrust—An Integer value from 1 to 2 specifying the certificate's trust status: 1=trusted, 2=untrusted. When adding a certificate to the hotlist, set this attribute to 2 for untrusted.

Tip Always set the CertStatus attributes before loading the actual certificate data, because otherwise the CMTS will assume the certificate is chained and will immediately attempt to verify it with the manufacturers and root certificates.

Note This procedure is identical to the one given for adding a certificate as a trusted certificate in the Adding a Certificate as a Trusted Certificate Using SNMP Commands, on page 1045, except that the docsBpi2CmtsProvisionedCmCertTrust attribute is set to 2 instead of 1.

For example, to use the Unix command-line SNMP utility to add a manufacturer's certificate to the hotlist on the CMTS at IP address 192.168.100.113, enter the following command (be sure to substitute a valid index pointer for the table entry for the <*index*> value).

```
% setany -v2c 192.168.100.113 private docsBpi2CmtsCACertStatus.
<index>
 -i 4
 docsBpi2CmtsCACert.
<index>
 -o
 '<hex_data>' docsBpi2CmtsCACertTrust.
<index>
 -i 2
```

To do the same thing for a CM certificate, use the following command:

```
% setany -v2c 192.168.100.113 private docsBpi2CmtsProvisionedCmCertStatus.
<index>
 -i 4
 docsBpi2CmtsProvisionedCmCert.
<index>
 -o
 '<hex_data>' docsBpi2CmtsProvisionedCmCertTrust.
<index>
 -i 2
```

Tip Most operating systems cannot accept input lines that are as long as needed to input the hexadecimal decimal string that specifies a certificate. For this reason, you should use a graphical SNMP manager to set these attributes. For a number of certificates, you can also use a script file, if more convenient.

Enabling Concatenation

To enable concatenation for one or more upstreams on a cable interface (which is the default configuration), use the following procedure:

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**interface cable***slot* / *port*
	Example:
	`Router(config)# interface cable 6/0`
	`Router(config-if)#`
		Enters interface configuration mode for the cable interface line card at this particular slot.

Step 4	**cable upstream** *n* **concatenation**
	Example:
	`Router(config-if)# cable upstream 0 concatenation`
	`Router(config-if)# cable upstream 1 concatenation`
	`Router(config-if)#`
		Enables concatenation for the specified upstream on the cable interface.
		Note Repeat this command for each upstream on the interface.

Step 5	**exit**
	Example:
	`Router(config-if)# exit`
	`Router(config)#`
		Exits interface configuration mode.

Step 6	**exit**
	Example:
	`Router(config)# exit`
	`Router#`
		Exits global configuration mode.

Enabling DOCSIS Fragmentation

To enable DOCSIS fragmentation for one or more upstreams on a cable interface (which is the default configuration), use the following procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable` **Example:**	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
	`Router#`	

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**interface cable***slot* /*port*
	Example:
	`Router(config)# interface cable 6/0`
	`Router(config-if)#`
		Enters interface configuration mode for the cable interface line card at this particular slot.

Step 4	**cable upstream***n* **fragmentation**
	Example:
	`Router(config-if)# cable upstream 2 fragmentation`
	`Router(config-if)# cable upstream 3 fragmentation`
	`Router(config-if)#`
		Enables fragmentation for the specified upstream on the cable interface.
		Note Repeat this command for each upstream on the interface.

Step 5	**cable upstream** *n* **unfrag-slot-jitter** [**limit***jitter* | **cac-enforce**]
	Example:
	`Router(config-if)# cable upstream 0 unfrag-slot-jitter limit 2000 cac-enforce`
	`Router(config-if)#`
		(Optional) Specifies the amount of jitter that can be tolerated on the upstream due to unfragmentable slots. The **limit** option specifies the allowable *jitter* limit in microseconds (0 to 4,294,967,295. The **cac-enforce** option configures the upstream so that it rejects service flows requesting jitter less than the fragmentable slot jitter.
		Note By default, *jitter* is set to a limit of 0 microseconds, and the **cac-enforce** option is enabled.

Step 6	**exit**
	Example:
	`Router(config-if)# exit`
	`Router(config)#`
		Exits interface configuration mode.

Step 7	**exit**
	Example:
	`Router(config)# exit`
	`Router#`
		Exits global configuration mode.

Example

The following example of the **show cable qos profile** command illustrates that the maximum downstream burst has been defined, and is a management-created QoS profile:

```
Router# show cable qos profile
ID  Prio Max       Guarantee Max        Max   TOS  TOS   Create  B     IP prec.
         upstream  upstream  downstream tx    mask value by      priv  rate
         bandwidth bandwidth bandwidth  burst                     enab  enab
1   0    0         0         0          0     0xFF 0x0   cmts(r) no    no
2   0    64000     0         1000000    0     0xFF 0x0   cmts(r) no    no
3   7    31200     31200     0          0     0xFF 0x0   cmts    yes   no
4   7    87200     87200     0          0     0xFF 0x0   cmts    yes   no
6   1    90000     0         90000      1522  0xFF 0x0   mgmt    yes   no
10  1    90000     0         90000      1522  0x1  0xA0  mgmt    no    no
50  0    0         0         96000      0     0xFF 0x0   mgmt    no    no
51  0    0         0         97000      0     0xFF 0x0   mgmt    no    no
```

The following example illustrates the maximum downstream burst size in sample QoS profile 10 with the **show cable qos profileverbose** command in privileged EXEC mode:

```
Router# show cable qos profile 10 verbose
Profile Index                           10
Name
Upstream Traffic Priority               1
Upstream Maximum Rate (bps)             90000
Upstream Guaranteed Rate (bps)          0
Unsolicited Grant Size (bytes)          0
Unsolicited Grant Interval (usecs)      0
Upstream Maximum Transmit Burst (bytes) 1522
Downstreamam Maximum Transmit Burst (bytes) 100000
IP Type of Service Overwrite Mask       0x1
IP Type of Service Overwrite Value      0xA0
Downstream Maximum Rate (bps)           90000
Created By                              mgmt
Baseline Privacy Enabled                no
```

Enabling DOCSIS 1.1 Downstream Maximum Transmit Burst on the Cisco cBR-8 Router

Perform the following steps to configure ERBA on the Cisco cBR-8 router. This procedure and the associated commands are subject to the guidelines and restrictions cited in this document.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	[**no**] **cable ds-max-burst burst-threshold** *threshold* **Example:** `Router(config)# cable ds-max-burst burst-threshold 2048`	Enables the support for DOCSIS 1.1 downstream max burst. To remove this configuration, use the **no** form of this command.
Step 4	**cable service class** *class-index* **peak-rate** *peak-rate* **Example:** `Router(config)# cable service class 1 peak-rate 1000`	Set the peak-rate value of a specific service class.

Step 5	**Ctrl^Z**
	Example:
	`Router(config)# Ctrl^Z`
	`Router#`
		Returns to privileged EXEC mode.

Example

When this feature is enabled, new service flows with burst size larger than the burst threshold are supported. However, the existing service flows are not affected.

When this feature is disabled, no new service flows are configured with the *Downstream Maximum Transmit Burst* parameter—the **cable ds-max-burst** command settings. However, the existing service flows are not affected.

Monitoring DOCSIS Operations

The following sections describe the commands that provide information about the DOCSIS network and its cable modems, the RF network and cable interfaces on the CMTS, and BPI+ operations.

Monitoring the DOCSIS Network

The **show cable modem** command is the primary command to display the current state of cable modems and the DOCSIS network. This command has many options that provide information on different aspects of DOCSIS operations.

Displaying the Status of Cable Modems

To display a list of known cable modems and their current status, use the **show cable modem** command.

You can also display a particular cable modem by specifying its MAC address or IP address with the **show cable modem** command. If you specify the MAC address or IP address for a CPE device, the command will display the information for the cable modem that is associated with that device.

Note If the CPE IP address is no longer associated with a cable modem, the **show cable modem** command might not display information about the cable modem. To display the IP address of the CPE device for the cable modem, use the **clear cable host ip-address** command to clear the IP address of the modem from the router database, and then enter the **ping docsis mac-address** command, which resolves the MAC address by sending the DOCSIS ping to the CM.

To display a list of cable modems sorted by their manufacturer, use the **vendor** option.

The MAC state field in each of these displays shows the current state of the cable modem:

Table 171: Descriptions for the MAC State Field

MAC State Value	Description
Registration and Provisioning Status Conditions	
init(r1)	The CM sent initial ranging.
init(r2)	The CM is ranging. The CMTS received initial ranging from the Cm and has sent RF power, timing offset, and frequency adjustments to the CM.
init(rc)	Ranging has completed.
init(d)	The DHCP request was received. This also indicates that the first IP broadcast packet has been received from the CM.
init(i)	The DHCP reply was received and the IP address has been assigned, but the CM has not yet replied with an IP packet.
init(o)	The CM has begun to download the option file (DOCSIS configuration file) using the Trivial File Transfer Protocol (TFTP), as specified in the DHCP response. If the CM remains in this state, it indicates that the download has failed.
init(t)	Time-of-day (TOD) exchange has started.
resetting	The CM is being reset and will shortly restart the registration process.
Non-error Status Conditions	

MAC State Value	Description
offline	The CM is considered offline (disconnected or powered down).
online	The CM has registered and is enabled to pass data on the network.
online(d)	The CM registered, but network access for the CM has been disabled through the DOCSIS configuration file.
online(pk)	The CM registered, BPI is enabled and KEK is assigned.
online(pt)	The CM registered, BPI is enabled and TEK is assigned. BPI encryption is now being performed.
expire(pk)	The Cm registered, BPI is enabled, KEK was assigned but has since expired.
expire(pt)	The Cm registered, BPI is enabled, TEK was assigned but has since expired.
Error Status Conditions	
reject(m)	The CM attempted to register but registration was refused due to a bad Message Integrity Check (MIC) value. This also could indicate that the shared secret in the DOCSIS configuration file does not match the value configured on the CMTS with the **cable shared-secret** command. It can also indicate that the **cable tftp-enforce** command has been used to require that a CM attempt a TFTP download of the DOCSIS configuration file before registering, but the CM did not do so.
reject(c)	The CM attempted to register, but registration was refused due to a a number of possible errors: • The CM attempted to register with a minimum guaranteed upstream bandwidth that would exceed the limits imposed by the **cable upstream admission-control** command. • The CM has been disabled because of a security violation. • A bad class of service (COS) value in the DOCSIS configuration file. • The CM attempted to create a new COS configuration but the CMTS is configured to not permit such changes.
reject(pk)	KEK key assignment is rejected, BPI encryption has not been established.
reject(pt)	TEK key assignment is rejected, BPI encryption has not been established.
reject(ts)	The CM attempted to register, but registration failed because the TFTP server timestamp in the CM registration request did not match the timestamp maintained by the CMTS. This might indicate that the CM attempted to register by replaying an old DOCSIS configuration file used during a prior registration attempt.
reject(ip)	The CM attempted to register, but registration failed because the IP address in the CM request did not match the IP address that the TFTP server recorded when it sent the DOCSIS configuration file to the CM. IP spoofing could be occurring.

MAC State Value	Description
reject(na)	The CM attempted to register, but registration failed because the CM did not send a Registration-Acknowledgement (REG-ACK) message in reply to the Registration-Response (REG-RSP) message sent by the CMTS. A Registration-NonAcknowledgement (REG-NACK) is assumed.

Displaying a Summary Report for the Cable Modems

The **show cable modem** command also can provide a summary report of the cable modems by using the **summary** and **total** options.

You can also use the **summary** and **total** options to display information for a single interface or a range of interfaces.

Displaying the Capabilities of the Cable Modems

To display the capabilities and current DOCSIS provisioning for cable modems, use the **mac** option.

To get a summary report of the cable modems and their capabilities, use the **mac** option with the **summary** and **total** options.

Displaying Detailed Information About a Particular Cable Modem

Several options for the show cable modem command display detailed information about a particular cable modem (as identified by its MAC address). The **verbose** option displays the most comprehensive output.

The **connectivity** and **maintenance** options also provide information that can be useful in troubleshooting problems with a particular cable modem.

Monitoring the RF Network and Cable Interfaces

You can use the **show interface cable** command to display information about the operation of the RF network and the cable interfaces on the CMTS.

Tip For a complete description of the **show cable interface** command and its options, see the "Cisco Cable Modem Termination System Commands" chapter in the *Cisco Broadband Cable Command Reference Guide* (see http://www.cisco.com/c/en/us/td/docs/cable/cbr/configuration/guide/b_cmts_quality_of_services/docsis_1_1.html#ref_1239231).

Displaying Information About Cloned Cable Modems

To display the list of cable modems detected as cloned, use the **privacy hotlist** option with the **show interface cable** command.

Denying RF Access For Cable Modems

To deny radio frequency (RF) access for cable modems during ranging, use the **cable privacy hotlist cm** *mac-address* command.

The following example shows how to block cloned cable modems using their own MAC address:

```
Router(config)# cable privacy hotlist cm 00C0.0102.0304
Router(config)#
```

When an operator identifies a modem's MAC address that should not be registered on a specific CMTS, the operator can add this MAC address to the CMTS using the above command. This command ensures that the modem will not be allowed to come online on any interface on that CMTS.

Displaying Information About the Mac Scheduler

To display information about the DOCSIS MAC layer scheduler that is operating on each cable interface, use the **mac-scheduler** option with the **show cable interface** command. You can display information for all of the upstreams on an interface, or you can display information for a single upstream on an interface.

Displaying Information About QoS Parameter Sets

To display information about the DOCSIS 1.1 QoS parameter sets that have been defined on a cable interface, use the **qos paramset** option with the **show cable interface** command.

You can also display detailed information for a particular parameter set by specifying the index number for its Class of Service along with the **verbose** option.

Displaying Information About Service Flows

To display the service flows and their QoS parameter sets that are configured on a cable interface, use the **service-flow** option with the **show interface cable** command.

To display the major QoS parameters for each service flow, add the **qos** option to this command.

To display the complete QoS parameters for a particular service flow, use the **qos** and **verbose** options. You can use these options separately or together.

Displaying Information About Service IDs

To display information about Service IDs (SIDs), which are assigned to only upstreams in DOCSIS 1.1 networks, use the **sid** option with the **show interface cable** command.

Add the **qos** option to display the major QoS parameters associated with each SID.

To display detailed information about a particular SID and its QoS parameters, use both the **qos** and **verbose** options.

Monitoring BPI+ Operations

See the following sections to monitor the state of BPI operations on the CMTS and its connected cable modems:

Displaying the Current BPI+ State of Cable Modems

To display the current BPI+ state of cable modems, use the **show cable modem** command. If used without any options, this command displays the status for cable modems on all interfaces. You can also specify a particular cable interface on the CMTS, or the IP address or MAC address for a specific cable modem:

```
Router# show cable modem
 [ip-address
```

```
| interface
| mac-address
```

The MAC State column in the output of the **show cable modem** command displays the current status of each cable modem. The following are the possible BPI-related values for this field:

Table 172: Possible show cable modem BPI+ States

State	Description
online	A cable modem has come online and, if configured to use BPI+, is negotiating its privacy parameters for the session. If the modem remains in this state for more than a couple of minutes, it is online but not using BPI+. Check that the cable modem is running DOCSIS-certified software and is using a DOCSIS configuration file that enables BPI+.
online(pk)	The cable modem is online and has negotiated a Key Encryption Key(KEK) with the CMTS. If BPI+ negotiation is successful, this state will be shortly followed by online(pt).
online(pt)	The cable modem is online and has negotiated a Traffic Encryption Key (TEK) with the CMTS. The BPI+ session has been established, and the cable modem is encrypting all user traffic with the CMTS using the specified privacy parameters.
reject(pk)	The cable modem failed to negotiate a KEK with the CMTS, typically because the cable modem failed authentication. Check that the cable modem is properly configured for BPI+ and is using valid digital certificates. If the CMTS requires BPI+ for registration, the cable modem will go offline and have to reregister. Check that the cable modem is properly registered in the CMTS provisioning system. **Note** If a cable modem fails BPI+ authentication, a message similar to the following appears in the CMTS log: `%CBR-5-UNAUTHSIDTIMEOUT: CMTS deleted BPI unauthorized Cable Modem 00c0.abcd.ef01`
reject(pt)	The cable modem failed to successfully negotiate a TEK with the CMTS. If the CMTS requires BPI+ for registration, the cable modem will have to reregister.

Displaying the BPI+ Timer Values on the CMTS

To display the values for the KEK and TEK lifetime timers on a particular cable interface, use the **show interface cable** *x/y* **privacy** [**kek** | **tek**] command.

Displaying the Certificate List on the CMTS

Use the **show crypt ca certificates** command to display the list of known certificates on the CMTS. For example:

```
Router# show crypto ca certificates

Certificate
  Status: Available
  Certificate Serial Number: 7DBF85DDDD8358546BB1C67A16B3D832
  Key Usage: General Purpose
  Subject Name
    Name: Cisco Systems
  Validity Date:
```

```
    start date: 00:00:00 UTC Sep 12 2001
    end   date: 23:59:59 UTC Sep 11 2021
Root certificate
  Status: Available
  Certificate Serial Number: 5853648728A44DC0335F0CDB33849C19
  Key Usage: General Purpose
    CN = DOCSIS Cable Modem Root Certificate Authority
     OU = Cable Modems
     O = Data Over Cable Service Interface Specifications
     C = US
  Validity Date:
    start date: 00:00:00 UTC Feb 1 2001
    end   date: 23:59:59 UTC Jan 31 2031
```

Configuration Examples for DOCSIS 1.1 Operations

This section lists the following sample configurations for DOCSIS 1.1 operations on the Cisco CMTS:

Example: DOCSIS 1.1 Configuration for Cisco cBR-8 Router (with BPI+)

```
version 12.2
service timestamps log datetime msec localtime
service password-encryption
!
hostname cBR-8
!
redundancy
 main-cpu
  auto-sync standard
logging queue-limit 100
no logging buffered
no logging rate-limit
enable password my-enable-password
!
ipc cache 5000
card 1/1 2cable-tccplus
card 2/0 1gigethernet-1
card 2/1 2cable-tccplus
card 3/0 1gigethernet-1
card 4/0 1oc12pos-1
card 8/0 5cable-mc520s
card 8/1 5cable-mc520s
cable flap-list insertion-time 60
cable flap-list power-adjust threshold 4
cable flap-list aging 86400
cable modem vendor 00.50.F1 TI
cable spectrum-group 2 band 11000000 16000000
cable spectrum-group 21 band 17000000 25000000
cable spectrum-group 32 shared
cable spectrum-group 32 band 5000000 42000000
cable modulation-profile 2 request 0 16 0 8 qpsk scrambler 152 no-diff 64 fixed uw16
cable modulation-profile 2 initial 5 34 0 48 qpsk scrambler 152 no-diff 128 fixed uw16
cable modulation-profile 2 station 5 34 0 48 qpsk scrambler 152 no-diff 128 fixed uw16
cable modulation-profile 2 short 6 75 6 8 16qam scrambler 152 no-diff 144 shortened uw8
cable modulation-profile 2 long 8 220 0 8 16qam scrambler 152 no-diff 160 shortened uw8
cable modulation-profile 21 request 0 16 0 22 qpsk scrambler 152 no-diff 32 fixed
cable modulation-profile 21 initial 5 34 0 48 qpsk scrambler 152 no-diff 64 fixed
cable modulation-profile 21 station 5 34 0 48 qpsk scrambler 152 no-diff 64 fixed
cable modulation-profile 21 short 3 76 12 22 qpsk scrambler 152 no-diff 64 shortened
```

```
cable modulation-profile 21 long 7 231 0 22 qpsk scrambler 152 no-diff 64 shortened
cable modulation-profile 22 request 0 16 0 22 qpsk scrambler 152 no-diff 32 fixed
cable modulation-profile 22 initial 5 34 0 48 qpsk scrambler 152 no-diff 64 fixed
cable modulation-profile 22 station 5 34 0 48 qpsk scrambler 152 no-diff 64 fixed
cable modulation-profile 22 short 4 76 7 22 16qam scrambler 152 no-diff 128 shortened
cable modulation-profile 22 long 7 231 0 22 16qam scrambler 152 no-diff 128 shortened
cable modulation-profile 23 request 0 16 0 22 qpsk scrambler 152 no-diff 32 fixed
cable modulation-profile 23 initial 5 34 0 48 qpsk scrambler 152 no-diff 64 fixed
cable modulation-profile 23 station 5 34 0 48 qpsk scrambler 152 no-diff 64 fixed
cable modulation-profile 23 short 4 76 7 22 16qam scrambler 152 no-diff 128 shortened
cable modulation-profile 23 long 7 231 0 22 16qam scrambler 152 no-diff 128 shortened
cable qos profile 5 max-downstream 10000
cable qos profile 5 max-upstream 1000
cable qos profile 5 priority 7
cable qos profile 5 tos-overwrite 0x3  0x0
cable qos profile 5 name cm_no_priority
cable qos profile 6 max-downstream 10000
cable qos profile 6 max-upstream 5000
cable qos profile 6 priority 7
cable qos profile 6 tos-overwrite 0x3  0x0
cable qos profile 6 name qos6
cable qos profile 7 max-downstream 128
cable qos profile 7 max-upstream 128
cable qos profile 7 priority 7
cable qos profile 8 max-downstream 10000
cable qos profile 8 max-upstream 1000
cable qos profile 8 priority 3
cable qos profile 8 tos-overwrite 0x3  0x0
cable qos profile 8 name qos8
no cable qos permission create
no cable qos permission update
cable qos permission modems
cable event syslog-server 10.10.10.131
ip subnet-zero
!
!
interface FastEthernet0/0/0
 ip address 10.10.32.21 255.255.0.0
 no cdp enable
!
interface GigabitEthernet2/0/0
 ip address 10.10.31.2 255.0.0.0
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 load-interval 30
 negotiation auto
 no cdp enable
!
interface GigabitEthernet3/0/0
 no ip address
 ip pim sparse-mode
 no ip route-cache cef
 load-interval 30
 shutdown
 negotiation auto
 no cdp enable
!
interface POS4/0/0
 no ip address
 crc 32
 no cdp enable
 pos ais-shut
!
```

```
!
interface Cable8/0/0
 ip address 10.10.10.28 255.255.255.0
 ip helper-address 1.10.10.133
 cable bundle 2 master
 cable downstream annex B
 cable downstream modulation 64qam
 cable downstream interleave-depth 32
 cable downstream frequency 669000000
 cable downstream channel-id 0
 no cable downstream rf-shutdown
 cable downstream rf-power 45
 cable upstream 0 connector 0
 cable upstream 0 spectrum-group 32
 cable upstream 0 power-level 0
 cable upstream 0 channel-width 1600000
 cable upstream 0 minislot-size 4
 cable upstream 0 range-backoff 3 6
 cable upstream 0 data-backoff 0 6
 cable upstream 0 modulation-profile 23
 no cable upstream 0 rate-limit
 no cable upstream 0 shutdown
 cable upstream 1 connector 1
 cable upstream 1 spectrum-group 32
 cable upstream 1 power-level 0
 cable upstream 1 channel-width 1600000
 cable upstream 1 minislot-size 4
 cable upstream 1 data-backoff 0 6
 cable upstream 1 modulation-profile 23
 no cable upstream 1 shutdown
 cable upstream 2 connector 2
 cable upstream 2 spectrum-group 32
 cable upstream 2 power-level 0
 cable upstream 2 channel-width 1600000
 cable upstream 2 minislot-size 4
 cable upstream 2 data-backoff 3 6
 cable upstream 2 modulation-profile 23
 no cable upstream 2 shutdown
 cable upstream 3 connector 3
 cable upstream 3 spectrum-group 32
 cable upstream 3 channel-width 1600000
 cable upstream 3 minislot-size 4
 cable upstream 3 modulation-profile 21
 no cable upstream 3 shutdown
 cable source-verify
 cable privacy kek life-time 300
 cable privacy tek life-time 180
 no keepalive
!
interface Cable8/0/1
 ip address 10.10.11.121
 cable bundle 2
 cable downstream annex B
 cable downstream modulation 64qam
 cable downstream interleave-depth 32
 cable downstream frequency 453000000
 cable downstream channel-id 0
 no cable downstream rf-shutdown
 cable upstream max-ports 6
 cable upstream 0 connector 4
 cable upstream 0 spectrum-group 2
 cable upstream 0 power-level 0
 cable upstream 0 channel-width 1600000
 cable upstream 0 minislot-size 4
```

```
 cable upstream 0 range-backoff 3 6
 cable upstream 0 data-backoff 0 6
 cable upstream 0 modulation-profile 23 21
 no cable upstream 0 rate-limit
 cable upstream 0 shutdown
 cable upstream 1 connector 5
 cable upstream 1 channel-width 1600000
 cable upstream 1 minislot-size 4
 cable upstream 1 modulation-profile 21
 cable upstream 1 shutdown
 cable upstream 2 connector 6
 cable upstream 2 channel-width 1600000
 cable upstream 2 minislot-size 4
 cable upstream 2 modulation-profile 21
 cable upstream 2 shutdown
 cable upstream 3 connector 7
 cable upstream 3 channel-width 1600000
 cable upstream 3 minislot-size 4
 cable upstream 3 modulation-profile 21
 cable upstream 3 shutdown
 cable upstream 4 channel-width 1600000
 cable upstream 4 minislot-size 4
 cable upstream 4 modulation-profile 21
 cable upstream 4 shutdown
 cable upstream 5 channel-width 1600000
 cable upstream 5 minislot-size 4
 cable upstream 5 modulation-profile 21
 cable upstream 5 shutdown
 cable source-verify
 cable privacy kek life-time 300
 cable privacy tek life-time 180
 no keepalive
!
!
ip classless
ip http server
no ip http secure-server
!
!
no cdp run
snmp-server community public RW
snmp-server community private RW
snmp-server enable traps cable
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 exec-timeout 0 0
 password my-telnet-password
 login
 length 0
!
end
```

Additional References

For additional information related to DOCSIS 1.1 operations, refer to the following references:

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

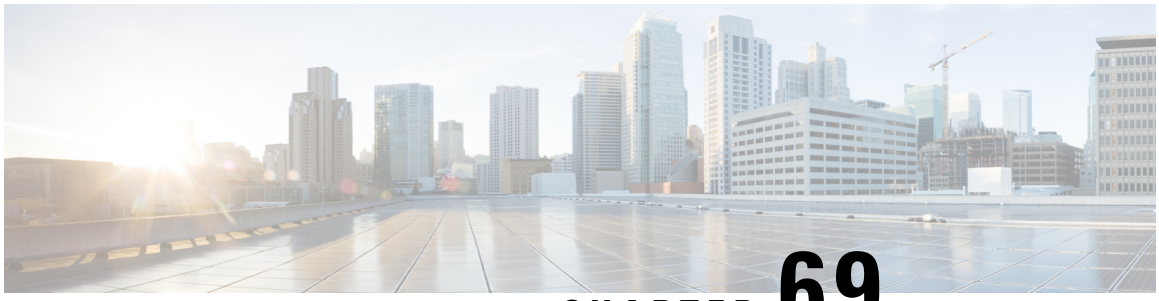
Feature Information for DOCSIS 1.1 for Cisco CMTS Routers

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 173: Feature Information for DOCSIS 1.1 for the Cisco CMTS Routers

Feature Name	Releases	Feature Information
DOCSIS 1.1 for the Cisco CMTS Routers	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on Cisco cBR Series Converged Broadband Router s.



CHAPTER 69

Default DOCSIS 1.0 ToS Overwrite

This document describes the Default DOCSIS 1.0 ToS Overwrite feature for the Cisco Cable Modem Termination System (CMTS). This feature eliminates the need to create multiple QoS profiles in order to perform type of service (ToS) overwrite by enabling a default ToS overwrite to be bound to all DOCSIS 1.0 Cable Modem (CM) created profiles.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1063
- Restrictions for Default DOCSIS 1.0 ToS Overwrite, on page 1064
- Information About Default DOCSIS 1.0 ToS Overwrite, on page 1064
- How to Configure Default DOCSIS 1.0 ToS Overwrite, on page 1065
- Additional References, on page 1067
- Feature Information for Default DOCSIS 1.0 ToS Overwrite, on page 1067

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 174: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for Default DOCSIS 1.0 ToS Overwrite

- The Default DOCSIS 1.0 ToS Overwrite feature is only applicable to CMs running DOCSIS version 1.0.
- Once the Default DOCSIS 1.0 ToS Overwrite feature is configured, all CMs will need to be reset in order for the effect to take place.
- Once the Default DOCSIS 1.0 ToS Overwrite feature is configured, all CMs will display the default values that were configured. After which, overwrite values can only be changed by editing the QoS profiles.

Information About Default DOCSIS 1.0 ToS Overwrite

To configure the Default DOCSIS 1.0 ToS Overwrite feature, you should understand the following topic:

Default DOCSIS 1.0 ToS Overwrite Overview

Currently, ToS overwrite requires the creation of static cable QoS profiles, which are assigned ToS fields and are then associated with 1.0 CMs. This implementation works well if only a few different service types are offered.

However, scalability issues arises when large numbers of service types are presented; each requiring a static QoS profile in order to perform ToS overwrite.

The Default DOCSIS 1.0 ToS Overwrite feature eliminates the need to create multiple QoS profiles in order to perform type-of-service (ToS) overwrite by automatically bounding all DOCSIS 1.0 Cable Modem (CM) created profiles to a default ToS overwrite.

DOCSIS

Created by CableLabs, Data Over Cable Service Interface Specification (DOCSIS) defines the interface standards and requirements for all cable modems associated with high-speed data distribution over a cable television system network.

The DOCSIS architecture consists of the following two components:

- Cable Modem (CM)
- Cable Modem Termination System (CMTS)

Each of these components are situated at different locations, often with the CM located on a customer site and the CMTS on the service provider site, and communication between the CM and CMTS is conducted over cable through DOCSIS.

Note Though there are several versions of DOCSIS available, the Default DOCSIS 1.0 ToS Overwrite feature is only applicable to CMs running DOCSIS 1.0.

Type-of-Service (ToS)

Tools such as type-of-service (ToS) bits identification make it possible to isolate network traffic by the type of application being used. ToS capabilities can be further expanded to isolate network traffic down to the specific brands, by the interface used, by the user type and individual user identification, or by the site address.

How to Configure Default DOCSIS 1.0 ToS Overwrite

The tasks in this section enables the use of the Default DOCSIS 1.0 ToS Overwrite feature.

Enabling Default DOCSIS 1.0 ToS Overwrite

All CMs with a DOCSIS 1.0 configuration file currently have their ToS overwrite default values are set to tos-and: 0xff and tos-or: 0x00. Since there were previously no mechanism in the DOCSIS 1.0 configuration file to specify the ToS overwrite, QoS profiles were created and assigned to the default ToS overwrites.

The following procedures enable the Default DOCSIS 1.0 ToS Overwrite feature, which will allow a default ToS overwrite to be bound to all CM created profiles.

Before you begin

There are no prerequisites for these procedures.

Note

- The Default DOCSIS 1.0 ToS Overwrite feature is only applicable to CMs running DOCSIS version 1.0.
- Once the Default DOCSIS 1.0 ToS Overwrite feature is configured, all CMs will need to be reset in order for the effect to take place.
- Once the Default DOCSIS 1.0 ToS Overwrite feature is configured, all CMs will display the default values that were configured. After which, overwrite values can only be changed by editing the QoS profiles.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configureterminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable default-tos-qos10 tos-overwrite** *tos-and tos-or* **Example:** `Router(config)# cable default-tos-qos10 tos-overwrite 0x1F 0xE0`	Configures the ToS overwrite default value for the CM. This default value will be bound to all future CM created profiles.
Step 4	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

What to do next

After configuring the ToS overwrite default value, reset the CM using the **clear cable modem delete** command to allow the new ToS overwrite default value to take effect.

Editing QoS Profiles

Once the Default DOCSIS 1.0 ToS Overwrite feature is configured, additional ToS overwrite values can be changed by editing the QoS profiles.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configureterminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable qos profile** {*groupnum* | *ip-precedence* | *guaranteed-upstream* | *max-burst* | *max-upstream* | *max-downstream* | *priority* | *tos-overwrite* | *value* **Example:** `Router(config)# cable qos profile 4 guaranteed-upstream 2`	Configures the QoS profile.
Step 4	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Additional References

The following sections provide references related to the Default DOCSIS 1.0 ToS Overwrite feature.

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, tools, and technical documentation. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Default DOCSIS 1.0 ToS Overwrite

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release,

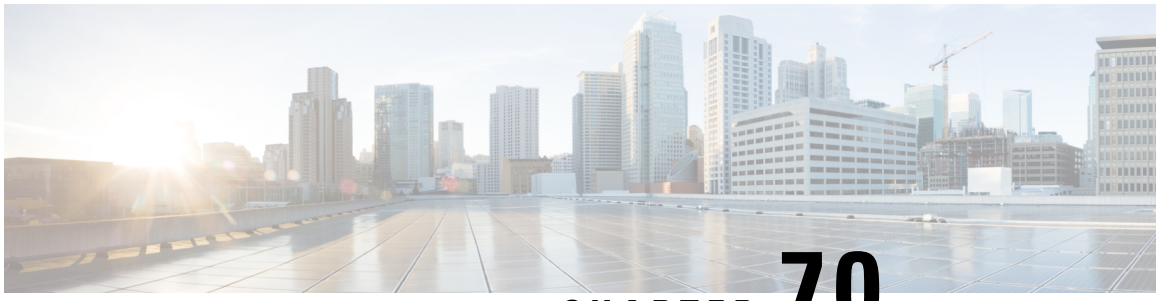
feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 175: Feature Information for Default DOCSIS 1.0 ToS Overwrite

Feature Name	Releases	Feature Information
Default DOCSIS 1.0 ToS overwrite	Cisco IOS-XE 16.5.1	This feature was integrated into Cisco <polaris release label> on the Cisco cBR Series Converged Broadband Router s.

-



CHAPTER 70

DOCSIS WFQ Scheduler on the Cisco CMTS Routers

The DOCSIS WFQ Scheduler is an output packet scheduler that provides output scheduling services on both WAN uplink interfaces and DOCSIS downstream interfaces.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1069
- Prerequisites for DOCSIS WFQ Scheduler, on page 1070
- Restrictions for DOCSIS WFQ Scheduler, on page 1070
- Information About DOCSIS WFQ Scheduler, on page 1070
- How to Configure DOCSIS WFQ Scheduler , on page 1076
- Additional References, on page 1077
- Feature Information for DOCSIS WFQ Scheduler, on page 1077

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 176: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for DOCSIS WFQ Scheduler

No specialequipment or softwareis needed to use the DOCSIS WFQ Scheduler feature.

Restrictions for DOCSIS WFQ Scheduler

- The DBS feature is only applicable to DOCSIS 3.0 downstream channel bonding.

Information About DOCSIS WFQ Scheduler

The DOCSIS WFQ scheduling engine is used to provide output packet scheduling services, including absolute priority queueing, weighted fair queueing, minimum rate guarantee, traffic shaping, and DOCSIS bonding group dynamic bandwidth sharing on the Cisco cBR-8 converged broadband router.

The DOCSIS WFQ Scheduler provides services on both WAN uplink interfaces and DOCSIS downstream interfaces. The scheduling parameters on WAN uplink interfaces are configured through the Modular QoS CLI (MQC). On cable downstream interfaces, queues are created for DOCSIS service flows with parameters configured by DOCSIS downstream QoS type, length, values (TLVs).

The default queue size for the DOCSIS service flows (with bandwidth greater than 150 Mbps) is based on the bandwidth on the cable downstream interfaces (see Table below). Additionally, the queue limit for all service flows can also be adjusted using the **cable queue-limit** command, buffer size in service class or downstream buffer control TLVs.

Note The default queue size change, and the **cable queue-limit** command do not affect the DOCSIS high priority queues.

Table below is an example of the queue size based on Annex B 256 QAM channels.

Table 177: Bandwidth, Queue Sizes, and Queue Limits

Channel	Bandwidth (Mbps)	Default Queue Size	Queue Size				
			1 ms	20 ms	30 ms	40 ms	200 ms
1	37.5	63	63	63	92	123	617
2	75	255	63	123	185	247	1235
3	112.5	255	63	185	277	370	1852
4	150	255	63	247	370	494	2470
5	187.5	319	63	308	463	617	3087
6	225	383	63	370	555	741	3705
7	262.5	447	63	432	648	864	4323
8	300	511	63	494	741	988	4940
12	450	767	63	741	1111	1482	7411
14	525	895	63	864	1296	1729	8646
16	600	1023	63	988	1482	1976	9881

The DOCSIS WFQ Scheduler also allows significant enhancement to the queue scaling limits.

The following sections explain the DOCSIS WFQ Scheduler features:

Queue Types

The DOCSIS WFQ Scheduler feature supports the following types of queues:

- Priority queues
- CIR queues
- Best Effort queues

Priority Queues

Priority queues are serviced with absolute priority over all the other queues. On DOCSIS downstream interfaces, the priority queues are configured by DOCSIS applications that request a priority service flow, for example, a packet cable voice service flow. On WAN uplink interfaces, the priority queues are configured by the MQC policy maps.

The following restrictions apply to priority queues:

- Only one priority queue is allowed per WAN uplink interface.
- Only one priority queue is allowed for low latency service flows created for each DOCSIS downstream interface.
- All low latency flows on a DOCSIS downstream are aggregated to the single priority queue.

CIR Queues

A CIR queue is guaranteed to be serviced with at least the Committed Information Rate (CIR). CIR queues are used to service DOCSIS service flows with non-zero minimum reserved rates. If the offered load to a CIR queue exceeds its CIR value, the excess traffic is serviced as best effort traffic.

Best Effort Queues

The Best Effort (BE) queues share the interface bandwidth not used by the priority queue and the CIR queues. The sharing is in proportion to each queue's excess ratio.

The following conditions apply to BE queues:

- On DOCSIS downstream interfaces, BE queues are created by DOCSIS service flows that do not request a minimum reserved rate.
- Each DOCSIS flow without a minimum reserved rate uses its own BE queue.

DOCSIS QoS Support

DOCSIS defines a set of quality of service (QoS) parameters, including traffic priority, maximum sustained traffic rate, minimum reserved traffic rate, maximum traffic burst, maximum downstream latency, and peak traffic rate.

The downstream service flows use the QoS parameters to specify the desired QoS. The downstream policer and scheduler provides services such as traffic shaping, bandwidth provisioning, traffic prioritization, and bandwidth guarantee.

The DOCSIS service flow parameters are mapped to the packet queue parameters and provided with appropriate QoS support for the packet queues to support the DOCSIS parameters

The following DOCSIS QoS parameters are supported:

- Traffic priority
- Maximum sustained traffic rate
- Minimum reserved traffic rate

Note The maximum traffic burst size and the peak traffic rate are supported as described in the http://www.cisco.com/c/en/us/td/docs/cable/cbr/configuration/guide/b_cmts_quality_of_services/docsis_wfq_scheduler.html#con_1085732.

Traffic Priority

The downstream channel bandwidth available to the best effort traffic, namely the channel bandwidth minus the amount consumed by the priority traffic and the CIR traffic, is allocated to the best effort service flows in proportion to their DOCSIS traffic priorities. For example, if there are three service flows sending packets at a particular moment over the same downstream channel, and their DOCSIS traffic priorities are 0, 1 and 3, respectively, their share of the channel bandwidth will be 1:2:4. To achieve this bandwidth allocation, each service flow is assigned a value known as its excess ratio which is derived from its DOCSIS priority. Table below shows the default mappings of DOCSIS priority to excess ratio.

Note When traffic priority for a flow is not explicitly specified, a default priority value of 0 is used as per the DOCSIS specification.

Table 178: DOCSIS Priority to Excess Ratio Mapping

DOCSIS Traffic Priority	Excess Ratio
0	4
1	8
2	12
3	16
4	20
5	24
6	28
7	32

Custom DOCSIS Priority to Excess Ratio Mappings

This option is introduced to configure custom priority to excess ratio mappings for downstream service flows that override the default mappings listed in the above Table.

Note The configured values are used only for new service flows that are created after the configuration has been applied. All the existing service flows maintain their previous excess ratio values.

The option to configure priority to excess ratio mappings is available on a per downstream forwarding interface basis and is applicable to legacy cable, wideband and modular cable, and integrated cable interfaces.

The cable downstream qos wfq weights command is used to configure the mappings.

Maximum Sustained Traffic Rate

The maximum sustained traffic rate (MSR) specifies the peak information rate of a service flow. The MSR of a service flow is mapped to the shape rate of the packet queue. When the maximum sustained traffic rate is not specified or set to zero, its traffic rate becomes limited only by the physical channel capacity set by DOCSIS specifications.

Minimum Reserved Traffic Rate

The minimum reserved traffic rate (MRR) specifies the minimum rate reserved for a service flow. The MRR of a service flow is mapped to the CIR of the packet queue, which ensures the minimum amount of bandwidth a queue gets under congestion. When the MRR is not specified, the CIR is set to zero as per DOCSIS specifications.

High Priority Traffic

High priority traffic flows are mapped to a Low Latency Queue (LLQ) on the data forwarding interface. The packets in LLQ are serviced with absolute priority over other queues on the same interface.

The following service flows require high priority service:

- Service flows with DOCSIS downstream latency TLV set to a value above zero. For example, PacketCable Multimedia Specification (PCMM) voice calls.
- PacketCable downstream service flows.
- Service flows with Unsolicited Grant Service (UGS) type—non-PacketCable voice calls—upstream flows.

Enhanced Rate Bandwidth Allocation

The DOCSIS WFQ Scheduler supports the Enhanced Rate Bandwidth Allocation (ERBA) feature for service flows. The ERBA feature allows cable modems (CMs) to burst their temporary transmission rates up to the full line rate for short durations of time. This capability provides higher bandwidth for instantaneous bandwidth requests without having to make changes to existing service levels in the QoS profile.

The DOCSIS WFQ Scheduler allows each service flow to have one dedicated queue. When ERBA is enabled for the service flow, the peak rate is implemented as the queue shape rate within the scheduler, while the maximum sustained rate is set as the token bucket refill rate. When ERBA is turned off, the burst size and the peak rate value are not used.

The maximum traffic burst parameter is used to control a service flow burst duration, to burst up to the channel line rate or a configured peak rate, when it is within its maximum burst size allowance. On the Cisco cBR-8 Converged Broadband Router, the **cable ds-max-burst** command is used to control this behavior explicitly.

Note The ERBA feature is not applicable for high priority service flows and multicast service flows.

Table below summarizes the ERBA support for the Cisco cBR-8 router.

Table 179: Enhanced Rate Bandwidth Allocation Support for the Cisco cBR-8 Router

	Policer Rate	Policer Exceed Action	Policer Token Bucket Size	Queue Shape Rate
Traditional Service Flow	Maximum Sustained Traffic Rate (unused)	Transmit	A value computed internally by CMTS (unused)	Maximum Sustained Traffic Rate
ERBA-Enabled Service Flow	Maximum Sustained Traffic Rate	Drop	Maximum Traffic Burst TLV	Peak Traffic Rate

For information about ERBA support on the Cisco CMTS routers, refer to Using Enhanced Bandwidth Rate Allocation (ERBA) Support for DOCSIS 1.0 Cable Modems at the following location: DOCSIS 1.1 for the Cisco CMTS Routers

Peak Traffic Rate

The *peak-rate* option of the **cable ds-max-burst** command allows you to specify the peak rate an ERBA-enabled service flow can use. The *peak-rate* value is a global value and is applied to all service flows created after the configuration of the **cable ds-max-burst** command. The default value of the *peak-rate* is zero.

If the DOCSIS 3.0 TLV 25.27 is specified for a service flow, the *peak-rate* value is set as the TLV value. However, if ERBA is not turned on for a service flow, the *peak-rate* value is ignored.

The *peak-rate* value can also be configured through cable service class command which forms part of the service class template. During modem registration or Dynamic Service Addition (DSA) operation, the service class name TLV 25.4 is sent to create the static or dynamic downstream service flow that matches the service class template. These downstream service flows are created with a specific *peak-rate* . If the *peak-rate* is not specified, then the value specified by the **cable ds-max-burst** command is used.

If a service flow has both service class and TLV 25.27 defined *peak-rate* , then the *peak-rate* value specified in the TLV is used.

Some of the DOCSIS 1.x and DOCSIS 2.0 cable modems, which are not fully DOCSIS 1.x or DOCSIS 2.0 compliant, may fail to come online when they receive TLV 25.27 from the Cisco CMTS during registration. In order to overcome this you can configure the **cable service attribute withhold-TLVs command with the peak-rate** keyword to restrict sending of this TLV to non-DOCSIS 3.0 cable modems.

DOCSIS 3.0 Downstream Bonding Support with Bonding Group Dynamic Bandwidth Sharing

DOCSIS 3.0 introduces the concept of downstream channel bonding. Each Bonding Group (BG) is made up of a collection of downstream channels, which can be used by one or more bonding groups. Each downstream channel can also serve as a primary channel in a MAC domain and carry non-bonded traffic, while being part of a BG.

Prior to DOCSIS 3.0 standards, the downstream service flows were associated with a single downstream interface, which in turn corresponded to a physical downstream on an RF channel. In DOCSIS 3.0, the downstream service flows are associated with the downstream bonding groups. These bonding groups can use multiple downstream RF channels.

DBS is the dynamic allocation of bandwidth for wideband (WB) and integrated cable (IC) interfaces sharing the same downstream channel. Due to the channel sharing nature of the bonding groups, the bandwidth

available to bonding groups or non-bonded channels is not fixed. The bandwidth depends on the configuration and the traffic load on the WB or IC.

Note Bonding groups are implemented as WB interfaces and non-bonded channels as IC interfaces.

In the DBS mode, the bandwidth of the shared RF channels is dynamically allocated among the WB and IC interfaces. The DBS enables efficient use of the underlying RF channel bandwidth even in the presence of high burst traffic. The DBS is configured at the WB or IC interface level. By default, bandwidth for a WB or IC channel is statically allocated (non-DBS).

For information about DBS support on the Cisco CMTS routers, refer to the Dynamic Bandwidth Sharing on the Cisco CMTS Router feature.

How to Configure DOCSIS WFQ Scheduler

You cannot configure the DOCSIS WFQ Scheduler feature as it is automatically loaded. The parameters that the schedule uses include the interface bandwidth and queue parameters.

This section describes the following required and optional procedures:

Mapping DOCSIS Priority to Excess Ratio

This section describes how to map DOCSIS priorities to custom excess ratios for downstream service flows. These custom mappings will override the default mappings.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface wideband-cable** *slot*/*subslot*/*port* :*wideband-channel* or **interface integrated-cable** *slot*/*subslot*/*port* :*rf-channel* **Example:** `Router(config)# interface wideband-cable 2/0/0:0` or `Router(config)# interface integrated-cable 1/0/0:0`	Enters interface configuration mode for the indicated cable downstream interface.

	Command or Action	Purpose
Step 4	**cable downstream qos wfq weigthts {weight1...weight8}** **Example:** `Router(config-if)# cable downstream qos wfq weights 10 20 30 40 50 60 70 80`	Configures the custom excess ratios for 8 priorities: **Note** The custom values are used only for new service flows and not existing ones.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying the Downstream Queues Information

To verify the downstream queue information for a modem, use the **show cable modem** [*mac-address* |*ip-address*]**service-flow** command.

To check queue stats of all queues on an Integrated-Cable or Wideband-Cable interface, use the **show cable dp queue** *interface* command.

Additional References

The following sections provide references related to the DOCSIS WFQ Scheduler feature.

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

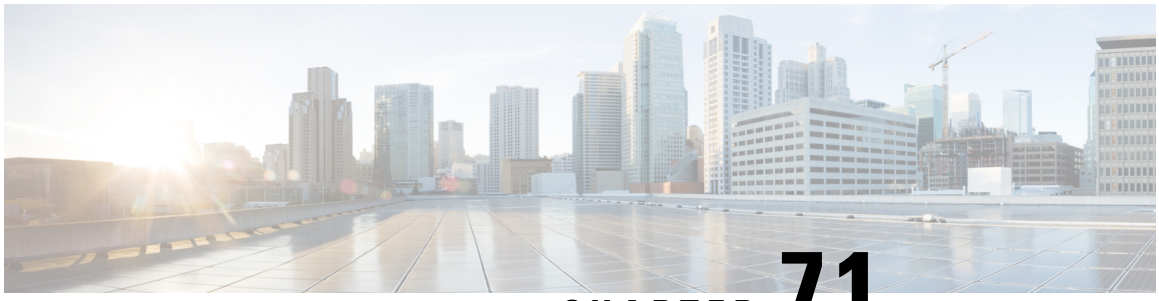
Feature Information for DOCSIS WFQ Scheduler

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 180: Feature Information for DOCSIS WFQ Scheduler

Feature Name	Releases	Feature Information
DOCSIS WFQ scheduler	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 71

Fairness Across DOCSIS Interfaces

The Fairness Across DOCSIS Interfaces feature introduces an adaptive mechanism to effectively distribute reservable bandwidth for committed information rate (CIR) flows and fair bandwidth for best-effort (BE) service flows across adjacent bonding groups (BGs).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1079
- Prerequisites for Fairness Across DOCSIS Interfaces , on page 1080
- Restrictions for Fairness Across DOCSIS Interfaces, on page 1080
- Information About Fairness Across DOCSIS Interfaces, on page 1081
- How to Configure Fairness Across DOCSIS Interfaces, on page 1082
- Verifying the Fairness Across DOCSIS Interfaces, on page 1086
- Configuration Examples for Fairness Across DOCSIS Interfaces, on page 1088
- Additional References, on page 1091
- Feature Information for Fairness Across DOCSIS Interfaces , on page 1091

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 181: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Fairness Across DOCSIS Interfaces

Note The term 'Bonding Group (BG)' is used in this document to refer to all the integrated-cable (IC) and wideband-cable (WC) interfaces in the context of Fairness Across DOCSIS Interfaces feature context. The IC interfaces are considered as a single-channel BG.

Restrictions for Fairness Across DOCSIS Interfaces

- The CIR flows cannot reserve all the RF bandwidth. The CIR flows can only reserve 90 percent[5] of the RF bandwidth that is not statically reserved by the "bandwidth-percent", in addition to the legacy CIR bandwidth.
- It is recommended that the CIR reservation be cleared before disabling Fairness Across DOCSIS Interfaces feature to ensure that the CIR reservation is not more than the static reservable bandwidth specified by the "bandwidth-percent" in legacy configuration. This is to prevent CIR over-subscription after disabling Fairness Across DOCSIS Interfaces feature.

[5] The reservable bandwidth for CIR flows consists of static and dynamic portions. By default, the static portion of bandwidth is assigned from the legacy configuration. The dynamic portion of bandwidth comes from the headroom left on each RF channel for BE traffic.

- The effect of Fairness Across DOCSIS Interfaces feature depends on topology and flow distribution. In certain cases, Fairness Across DOCSIS Interfaces feature may not achieve BE fairness or maximum CIR utilization.
- Fairness Across DOCSIS Interfaces feature applies only to dynamic bandwidth sharing (DBS) enabled IC and WB interfaces.

Information About Fairness Across DOCSIS Interfaces

The Fairness Across DOCSIS Interfaces feature is an enhancement over the DOCSIS WFQ scheduler. It enables downstream CIR service flows to be admitted on the interfaces over the thresholds defined in the legacy configuration (that is, “bandwidth-percent” or “max-reserved-bandwidth”). For example, the feature enables large CIR flows (like multicast service flows) to be admitted when the current parameters cannot guarantee enough bandwidth. However, its success rate depends on the allocation and reservation of the bandwidth for cable interfaces within common RF channels.

This feature also ensures fair bandwidth for downstream BE service flows across cable interfaces with common RF channels. The per-flow bandwidth of all active service flows on the adjacent BGs are balanced periodically. The weights (DOCSIS traffic priority (traffic priority + 1)) of all the BGs are equal for downstream BE service flows. The bandwidth, available for BE traffic, can be used to admit additional CIR flows.

Note For information about DOCSIS traffic priority, see DOCSIS WFQ Scheduler on the Cisco CMTS Routers guide.

On-demand CIR Acquisition

When multiple bonding groups sharing the RF-channel bandwidth and the current bonding group's guaranteed bandwidth is insufficient, this feature can "borrow" neighbor bonding group's non-reserved guaranteed bandwidth for current bonding group's CIR.

This feature is only used by multicast service flow.

Fairness Across Bonding Groups

Fairness Across DOCSIS Interfaces feature use the weight value of the aggregated active flow count, that is EIR demand, to periodically re-balance the reservable bandwidth. So that the service flows with the same weight in different bonding groups will have roughly the same throughput.

OFDM Channels

OFDM Channel

DOCSIS 3.1 introduces modes for higher throughput and higher spectral efficiency while still allowing backward compatibility to DOCSIS 3.0. OFDM Channel support includes 1 OFDM channel per port with channel bandwidth from 24 MHz to 192 MHz wide. In Cisco IOS-XE 16.5.1, a bonding group can consist of SC-QAMs and OFDM channels. An OFDM channel can have multiple profiles configured, and each profile

may have different rate. The OFDM Channel rate can vary constantly depending on the profiles being used. For more information on OFDM channels, see *OFDM Channel Configuration Guide*.

OFDM Channel Rate

An OFDM channel can have multiple profiles configured, and each profile can have different rates. For example, with a 96MHz OFDM channel that is configured with profile A (Control Profile) with modulation 1024-QAM, profile B with modulation 2048-QAM, and profile C with modulation 4096-QAM, the profile rates of profile A, B, and C are 616Mbps, 680Mbps, and 736Mbps respectively.

In Cisco IOS-XE 16.5.1, if an OFDM channel has both Control Profile (profile A) and Data Profiles (profile B, C, and so on) configured, the lowest Data Profile rate is used for Fairness Across DOCSIS Interface calculation. Otherwise, the Control Profile rate is used.

Interface Bandwidth

A Wideband-Cable (WB) interface can consist of both SC-QAMs and OFDM channels. If it contains OFDM channels, the highest profile rates are used to calculate the interface bandwidth.

For example, with a 96MHz OFDM channel that is configured with profile A having modulation 1024-QAM, profile B with modulation 2048-QAM, and profile C with modulation 4096-QAM, the profile rates of profile A, B, and C are 616Mbps, 680Mbps, and 736Mbps respectively. Here, 736Mbps is used to calculate the interface bandwidth.

How to Configure Fairness Across DOCSIS Interfaces

This section describes the following tasks that are required to implement Fairness Across DOCSIS Interfaces feature:

Configuring Fairness Across DOCSIS Interfaces

This section describes how to enable Fairness Across DOCSIS Interfaces feature on the cable interfaces. The configuration is applied to all WB or IC interfaces on the router.

Restriction We recommend that you clear the CIR reservation before disabling the Fairness Across DOCSIS Interfaces feature to ensure that CIR reservation is not more than the static reservable bandwidth specified by the “bandwidth-percent” in the legacy configuration.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable acfe enable** **Example:** `Router(config)# cable acfe enable`	Enables Fairness Across DOCSIS Interfaces feature on the cable interfaces.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Maximum Excess Information Rate Ratio

This section describes how to configure the maximum Excess Information Rate (EIR) ratio between the BE bandwidth among adjacent BGs.

The EIR ratio is used to maintain the maximum EIR bandwidth difference between BGs. It helps to prevent BGs (which has only a few active BE service flows) from getting very low or zero EIR bandwidth. Otherwise, these BGs will not be able to admit CIR flows as they get only very low EIR bandwidth.

For example, there are two BGs sharing the same RF channel, with BG1 having 1000 active BE service flows and BG2 having none. If “max-eir-ratio” is not used, BG1 gets all the bandwidth leaving no bandwidth for BG2. When a voice CIR tries for bandwidth at BG2, it will get rejected. If “max-eir-ratio” is set at 10, BG2 gets about 10 percent of the QAM that is sufficient to admit the voice CIR. The ‘max-eir-ratio’ is a trade-off between perfect fairness and CIR utilization. It means, compromising 'flow fairness' to prevent some BGs from getting all the bandwidth leaving the other BGs with none.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable acfe max-eir-ratio** *eir-ratio* **Example:**	Configures the maximum EIR ratio between the BE bandwidth among adjacent BGs.

	Command or Action	Purpose
	`Router(config)# cable acfe max-eir-ratio 20`	
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Constant Excess Information Rate Demand

This section describes how to configure the constant excess information rate (EIR) demand for a bonding group (BG). EIR demand is a unitless value that is used to determine relative bandwidth ratio between BGs.

An active EIR flow with DOCSIS priority-0 is given 1000 units of demand in ACFE module. Therefore a BG with constant-eir-demand set to 1 will get no more than 1/1000 of the bandwidth of a single service flow.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable acfe constant-eir-demand** *value* **Example:** `Router(config)# cable acfe constant-eir-demand 20`	Configures the constant EIR demand as 20 for a BG.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Maximum Bonus Bandwidth

This section describes how to configure the maximum usable bonus bandwidth for a BG.

Bonus bandwidth is the additional bandwidth provided by the Fairness Across DOCSIS Interfaces feature to each BG for CIR reservation. In the default maximum bonus bandwidth configuration, a single BG can reserve

all the underlying RF bandwidth. When the maximum bonus is set, the AC module will not admit CIR flows above that setting even if the scheduler has guaranteed more bandwidth. This will effectively prevent BGs from being starved for CIR flows.

Note The **cable acfe max-bonus-bandwidth** command configuration is applicable only for the new incoming CIR flows. It will not terminate the existing CIR flows that exceeds the **max-bonus-bandwidth** .

Restriction If the maximum bonus bandwidth is less than the current CIR reservation on an interface, no new CIR flows are admitted until the CIR reservation drops below the maximum bonus bandwidth configuration.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** {**wideband-cable** | **interface-cable** }*slot*/*subslot* /*port* :*interface-num* **Example:** `Router(config)# interface wideband-cable 1/0/0:0`	Specifies the interface to be configured. **Note** The valid values for the arguments depend on CMTS router and cable interface line card. See the hardware documentation for your router chassis and cable interface line card for supported values.
Step 4	**cable acfe max-bonus-bandwidth** *bonus-bandwidth* **Example:** `Router(config-if)# cable acfe max-bonus-bandwidth 1000000`	Configures the maximum usable bonus bandwidth for a BG.
Step 5	**end** **Example:** `Router(config)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying the Fairness Across DOCSIS Interfaces

To monitor the Fairness Across DOCSIS Interfaces feature, use the following procedures:

Verifying Reservable Bandwidth

To display the reserved and reservable bandwidth for a particular interface, use the **show interface {wideband-cable | modular-cable | integrated-cable}** command as shown in the example:

```
Router# show interfaces wideband-cable 1/0/0:1 downstream
Total downstream bandwidth 1875 Kbps
Total downstream reserved/reservable bandwidth 20000/1500 Kbps
Total downstream guaranteed/non-guaranteed bonus bandwidth 20760/9741 Kbps
Router#
```

The “reservable bandwidth” is a part of the guaranteed bandwidth from the legacy configuration. When the Fairness Across DOCSIS Interfaces feature is disabled, values of both the “guaranteed bonus bandwidth” and “non-guaranteed bonus bandwidth” is zero. When the feature is enabled, the “reservable bandwidth” and “guaranteed bonus bandwidth” represents the maximum CIR that can be reserved on the interface. Unicast CIR flows exceeding this limit are rejected. The additional “non-guaranteed bonus bandwidth” allows the multicast CIR flows to pass the AC module. However, the service flow may not be created successful because the bandwidth comes from the shared pool.

To display the reserved and reservable bandwidth for a particular interface, use the **show cable admission-control interface** command as shown in the example:

```
Router#show cable admission-control interface wideband-Cable 1/0/0:0

Interface Wi1/0/0:0
BGID: 28673

Resource - Downstream Bandwidth
-------------------------------
App-type    Name       Reservation/bps    Exclusive
1                      0                  Not configured
2                      0                  Not configured
3                      0                  Not configured
4                      0                  Not configured
5                      0                  Not configured
6                      0                  Not configured
7                      0                  Not configured
8                      20000000           Not configured
Max Reserved BW = 1500000 bps
Total Current Reservation = 20000000 bps
Guaranteed Bonus BW = 20760000 bps
Non-guaranteed Bonus BW = 9741000 bps
Subset BGs: In1/0/0:8 In1/0/0:9 In1/0/0:10 In1/0/0:11 In1/0/0:12
Superset BGs: N/A
Overlapping BGs: Wi1/0/0:8 Wi1/0/0:9 Wi1/0/0:10
Router#
```

Effective with Cisco IOS-XE Release 3.18.0SP, Capacity BW is also displayed. It is a summation of the channel capacity of the RF channels in this interface, and the capacity of OFDM channels is calculated considering the lowest profile rate.

```
Router#show cable admission-control interface wideband-Cable 2/0/0:1

Interface Wi2/0/0:1
BGID: 8194

Resource - Downstream Bandwidth
-------------------------------
App-type    Name       Reservation/bps    Maximum        Rejected
1                      4000               90%            0
2                      0                  N/A            0
3                      0                  90%            0
4                      0                  N/A            0
5                      0                  N/A            0
6                      0                  90%            0
7                      0                  N/A            0
8                      0                  87%            0
Max Reserved BW = 11424000 bps
Total Current Reservation = 4000 bps
Guaranteed Bonus BW = 884352000 bps
Non-guaranteed Bonus BW = 225904000 bps
Capacity BW = 1428000000 bps
Subset BGs: In2/0/0:0 In2/0/0:1 In2/0/0:2 In2/0/0:3 In2/0/0:4 In2/0/0:5 In2/0/0:6 In2/0/0:7
 In2/0/0:158 Wi2/0/0:0
Superset BGs: N/A
```

Verifying Global Fairness Across DOCSIS Interfaces Status and Statistics

To display the global status and statistics of the Fairness Across DOCSIS Interfaces feature, use the **show cable acfe summary** command as shown in the example:

```
Router# show cable acfe summary
ACFE state: Enabled
EIR Rebalance period (secs): 5
EIR Rebalance invocations:   254
CIR Acquire rate/limit:      100/100
CIR Acquire invocations:     0
CIR Acquire throttled:       0
CIR Oversubscriptions:       0
Maximal EIR ratio:           10
Constant EIR demand:         2
```

Verifying Per-Controller Fairness Across DOCSIS Interfaces Status and Statistics

To display the status and statistics for each controller interface, use the **show cable acfe controller** command as shown in the following example:

```
Router# show cable acfe controller integrated-Cable 1/0/0
EIR Rebalance invoked: 450963
Adaptive CIR granted: 20
Adaptive CIR rejected: 1
Total clusters: 9
RF FlexBW
8 36376
9 36376
10 32625
......
```

The BG clusters span across multiple channels and are used as a means to share the underlying RF channel bandwidth dynamically.

Use the **show controllers integrated-Cable acfe cluster** command to show Per-controller statistics and clusters and checking the bandwidth information as follows:

```
Router# show controllers integrated-Cable 1/0/0 acfe cluster 0
Integrated-Cable 1/0/0 status:
Topology changed: No

========Cluster 0========
Number of RF: 2
RF FlexBW  WB  ExcessBW Quanta
0  35625   -   35438    35438
           0   187   187
1  35250   0  35250   35250

Number of BG: 2
Intf Demand CIR Max  CstrMin Alloc NBonus Ratio
WB0  1000  0  70875  35250  35437  35438  14855190400
IC0  1000  0  35625  0      35438  187    14855609600
```

Verifying Per-Interface Fairness Across DOCSIS Interfaces Status and Statistics

To display the status and statistics for each interface, use the **show cable acfe interface** command as shown in the following example:

```
Router# show cable acfe interface wideband-cable 1/0/0:1
EIR Demand (raw/scale):   0/1
Per-Flow EIR BW (kbps):   19125
Guar Bonus BW (kbps):     19125
Non-guar Bonus BW (kbps): 38250
Reserved Bonus BW (kbps): 0
!
```

Configuration Examples for Fairness Across DOCSIS Interfaces

This section lists the following sample configurations for the Fairness Across DOCSIS Interfaces feature on a Cisco CMTS router:

Example: Fairness Across DOCSIS Interfaces

The following sample configuration shows Fairness Across DOCSIS Interfaces feature enabled on the router:

```
Current configuration : 39682 bytes
!
! Last configuration change at 04:30:02 UTC Wed Jan 19 2
! NVRAM config last updated at 04:23:17 UTC Wed Jan 19 2
!
version 12.2
```

```
!
cable clock dti
cable acfe enable
!
.
.
.
```

Example: Maximum EIR Demand Ratio

The following sample configuration shows maximum EIR demand ratio configured on the router:

```
Building configuration...
Current configuration : 54253 bytes
!
version 12.2
!
cable clock dti
cable acfe enable
cable acfe max-eir-ratio 20
!
```

The effect of the **cable acfe max-eir-ratio** command is demonstrated using a simple BG cluster.

```
!
interface integrated-Cable1/0/0:0
cable bundle 1
 cable rf-bandwidth-percent 10
!
interface Wideband-Cable9/0/0:0
 cable bundle 1
  cable rf-channels channel-list 0
  bandwidth-percent 1
  end
!
```

On this RF channel, 20 percent of the bandwidth is reserved by the 'bandwidth-percent' allowing Fairness Across DOCSIS Interfaces feature to use 27 Mbps, that is: (100 - 20) * 90 * 37.5). If the 'max-eir-ratio' is above 100 and the WB interface has 99 active BE flows and the IC interface has only 1 BE flow, then IC interface gets only 270 kbps, that is 1/(1+99)*27 of the bonus bandwidth. The BE traffic enjoys perfect fairness here. However, it is not possible to admit a unicast CIR flow beyond 270 kbps on the IC interface, as it would exceed the bonus bandwidth. If the 'max-eir-ratio' is set to 10, then the IC interface is treated to have 99/10 flows on it, resulting in a higher bonus bandwidth allocation. The 'max-eir-ratio' is a trade-off between perfect fairness and CIR utilization.

Example: Constant EIR Demand

The following sample configuration shows constant EIR demand on the router:

```
Building configuration...
Current configuration : 54253 bytes
!
version 12.2
!
cable clock dti
cable acfe enable
cable acfe max-eir-ratio 20
```

```
cable acfe constant-eir-demand 2
!


!
interface integrated-Cable1/0/0:0
cable bundle 1
 cable rf-bandwidth-percent 10
 cable acfe constant-eir-demand 2
!
interface Wideband-Cable9/0/0:0
 cable bundle 1
  cable rf-channels channel-list 0
  bandwidth-percent 1
  cable acfe constant-eir-demand 2
  end
!
```

Example: Maximum Bonus Bandwidth

The following sample configuration shows the maximum bonus bandwidth enabled on the router:

```
Building configuration...
Current configuration : 274 bytes
!
interface Wideband-Cable1/0/0:0
cable bundle 1
cable rf-channel 0 bandwidth-percent 10
cable acfe max-bonus-bandwidth 10000
end
!
```

In this per-interface configuration, even if the Fairness Across DOCSIS Interfaces feature guarantees more than 10 Mbps for a WB interface, the AC module will not pass more than 10 Mbps bandwidth above the legacy reservable bandwidth.

```
!
.
.
.
```

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Fairness Across DOCSIS Interfaces

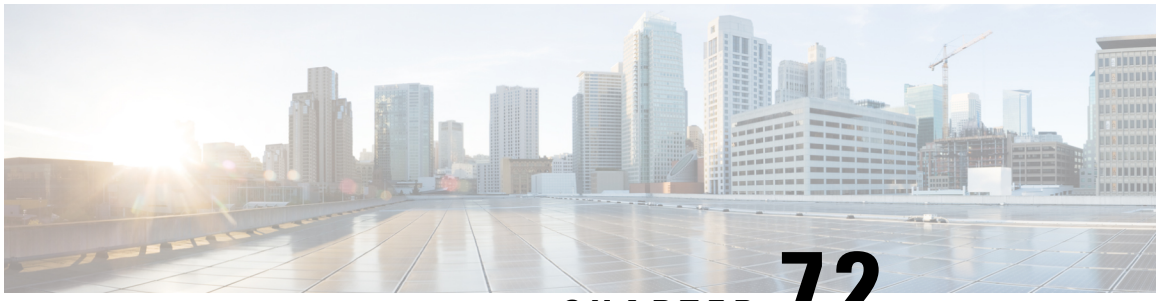
Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 182: Feature Information for Downstream Interface Configuration

Feature Name	Releases	Feature Information
Fairness Across DOCSIS Interfaces	Cisco IOS-XE 3.15.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router .
Fairness Across DOCSIS Interfaces	Cisco IOS-XE 3.17.0S	On-demand CIR acquisition and Fairness across bonding groups were supported. The following commands were modified: • **show cable acfe summary** • **show cable acfe interface**

Feature Name	Releases	Feature Information
Hierarchical QoS	Cisco IOS-XE 3.18.0SP	This feature was introduced on the Cisco cBR Series Converged Broadband Router.



CHAPTER 72

Service Group Admission Control

This document describes the Service Group Admission Control feature.

- Finding Feature Information, on page 1093
- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1093
- Restrictions for Service Group Admission Control, on page 1094
- Information About Service Group Admission Control, on page 1094
- How to Configure, Monitor, and Troubleshoot Service Group Admission Control, on page 1096
- Configuration Examples for SGAC, on page 1102
- Additional References, on page 1104
- Feature Information for Service Group Admission Control, on page 1105

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 183: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for Service Group Admission Control

- To configure SGAC, the Fairness Across DOCSIS Interfaces feature must be enabled.
- SGAC supports downstream only.

Information About Service Group Admission Control

Overview

Service Group Admission Control (SGAC) is a mechanism that gracefully manages service group based admission requests when one or more resources are not available to process and support the incoming service request. Lack of such a mechanism not only causes the new request to fail with unexpected behavior but could potentially cause the flows that are in progress to have quality related problems. SGAC monitors such resources constantly, and accepts or denies requests based on resource availability.

SGAC enables you to provide a reasonable guarantee about the Quality of Service (QoS) to subscribers at the time of call admission, and to enable graceful degradation of services when resource consumption approaches critical levels. SGAC reduces the impact of unpredictable traffic demands in circumstances that would otherwise produce degraded QoS for subscribers.

Note SGAC begins graceful degradation of service when either a critical threshold is crossed, or when bandwidth is nearly consumed on the Cisco CMTS, depending on the resource being monitored.

SGAC enables you to configure thresholds for each resource on the Cisco CMTS. These thresholds are expressed in a percentage of maximum allowable resource utilization. Alarm traps may be sent each time a threshold is crossed for a given resource.

For downstream (DS) channels, you can configure the bandwidth allocation with thresholds for each fiber node.

SGAC and Downstream Bandwidth Utilization

SGAC allows you to control the bandwidth usage for various DOCSIS traffic types or application types. The application types are defined by the user using a CLI to categorize the service flow.

Categorization of Service Flows

The SGAC feature allows you to allocate the bandwidth based on the application types. Flow categorization allows you to partition bandwidth in up to eight application types or buckets. The composition of a bucket is defined by the command-line interface (CLI), as is the definition of rules to categorize service flows into one of these eight application buckets. Various attributes of the service flow may be used to define the rules.

For flows created by PacketCable, the following attributes may be used:

- The priority of the Packetcable gate associated with the flow (high or normal)

For flows created by PacketCable MultiMedia (PCMM), the following attributes may be used:

- Priority of the gate (0 to 7)
- Application type (0 to 65535)

All flows use the following attribute type:

- Service class name

Before a service flow is admitted, it is passed through the categorization routine. Various attributes of the service flow are compared with the user-configured rules. Based on the match, the service flow is labeled with application type, from 1 to 8. The bandwidth allocation is then performed per application type.

Before a service flow is admitted, it is categorized based on its attributes. The flow attributes are compared against CLI-configured rules, one bucket at a time. If a match is found for any one of the rules, the service flow is labeled for that bucket, and no further check is performed.

Bucket 1 rules are scanned first and bucket 8 rules are scanned last. If two different rules match two different buckets for the same service flow, the flow gets categorized under the first match. If no match is found, the flow is categorized as Best Effort (BE) and the bucket with best effort rule is labelled to the flow. By default, the BE bucket is bucket 8.

Thresholds for Downstream Bandwidth

SGAC monitors downstream bandwidth consumption using the configured maximum reserved bandwidth. It rejects service flows with a non-zero minimal rate that would make the total reserved bandwidth exceed the configured threshold.

Flexible Bandwidth Allocation

To address the issue of restricted bandwidth allocation for different application types, admission control can be applied for both normal priority and emergency voice flows. This is done by extending the threshold and assigning a group of application types in a fiber node. Each downstream service flow continues to be categorized for a single application type. However, the one-to-one mapping between an application type and a threshold no longer exists.

Each configured threshold and its associated group of application types can thus be treated as a constraint. A service flow categorized to a certain application type must pass all the constraints associated with that application type.

Overview of Bonding Group Admission Control

DOCSIS 3.0 introduced bonded channels or bonding groups that allow a single cable modem to send data over multiple RF channels achieving higher throughput. These bonding groups are defined for both upstream and downstream channels. Bonding groups are created by combining multiple RF channels. A single RF channel may also be shared by multiple bonding groups.

Note Effective from Cisco IOS-XE 3.18.0SP Release, as per DOCSIS 3.1, if bonding group contains an OFDM channel, the bonding group's total bandwidth that can be reserved (its capacity), is calculated using the least efficient OFDM profile it can use.

Bonding group SGAC functionality allows to define the maximum reserved bandwidth for an application-type as a fraction of the available bandwidth. This fraction of the bandwidth is defined as a percentage value of the total bandwidth that can be reserved.

How to Configure, Monitor, and Troubleshoot Service Group Admission Control

Configuration procedures are optional because the default configurations are enabled by default. This section presents a sequence of procedures for non-default configurations, monitoring and debugging procedures for both the default or non-default operations of SGAC.

Defining Rules for Service Flow Categorization

This procedure describes how to configure service flow categorization rules on the Cisco CMTS. This flexible procedure changes default global service flow rules with variations of the **cable application type include** command.

Any one or several of these steps or commands may be used, in nearly any combination, to set or re-configure SGAC on the Cisco CMTS.

Note Application rules for SGAC are global configurations, and downstream bandwidth resources use the same sets of service flow rules.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable application-type** *n* **include packetcable** { **normal** | **priority** } **Example:** `Router(config)# cable application-type 5 include packetcable priority`	For PacketCable, this command variation maps PacketCable service flow attributes to the specified bucket. PacketCable service flows are associated with PacketCable gates. The gate can be normal or high-priority.

Step 4	**cable application-type** *n* **include pcmm** {**priority** *gate-priority* / **app-id** *gate-app-id* }
	Example:
	`Router(config)# cable application-type 2 include pcmm priority 7`
	`Router(config)# cable application-type 2 include pcmm app-id 152`
		For PCMM, this command variation maps PCMM service flow priority or application to the specified bucket. The PCMM gates are characterized by a priority level and by an application identifier.

Step 5	**cable application-type** *n* **include service-class** *service-class-name* **Example:** `Router(config)# cable application-type 1 include service-class stream1`	For service class parameters, this command variation applies a service class name to the service flows, and applies corresponding QoS parameters. DOCSIS 1.1 introduced the concept of service classes. A service class is identified by a service class name. A service class name is a string that the Cisco CMTS associates with a QoS parameter set One of the objectives of using a service class is to allow the high level protocols to create service flows with the desired QoS parameter set. Using a service class is a convenient way to bind the application with the service flows. The rules provide a mechanism to implement such binding. Note the following factors when using the command in this step:

	Command or Action	Purpose
		• Service classes are separately configured using the **cable service class** command to define the service flow. • A named service class may be classified into any application type. • Up to ten service class names may be configured per application types. Attempting to configure more than ten service classes prints an error message. • Use the **no cable traffic-type** command to remove the configuration of a service class before adding a new class.
Step 6	**cable application-type** *n* **include BE** **Example:** `Router# cable application-type 3 include BE`	For Best Effort service flows, this command variation elaborates on Step 3, and changes the default bucket of 8 for Best Effort service flows with non-zero Committed Information Rate (CIR). These BE service flows are often created during cable modem registration.
Step 7	**Ctrl-Z** **Example:** `Router(config)# Ctrl^Z`	Returns to Privileged EXEC mode.

Example

The following example maps high-priority PacketCable service flows into application bucket 5.

```
Router(config)# cable application-type 5 include packetcable priority
```

The following example maps normal PacketCable service flows into application bucket 1.

```
Router(config)# cable application-type 1 include packetcable normal
```

The following example maps the specified bucket number with PCMM service flow with a priority of 7, then maps an application identifier of 152 for the same bucket number:

```
Router(config)# cable application-type 2 include pcmm priority 7
Router(config)# cable application-type 2 include pcmm app-id 152
```

The following example maps the Best Effort CIR flows to bucket 3:

```
Router(config)# cable application-type 3 include BE
```

Naming Application Buckets

This procedure enables you to assign alpha-numeric names to six of the eight application buckets that SGAC supports. The default bucket identifiers range from 1 to 8.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configureterminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable application-type** *n***name** *bucket-name* **Example:** `Router(config)# cable application-type 7 name besteffort`	Assigns an alpha-numeric name for the specified bucket. **Note** This bucket name appears in supporting **show** and **debug** commands along with the default bucket number.
Step 4	Ctrl-Z **Example:** `Router(config)# Ctrl^Z`	Returns to Privileged EXEC mode.

Preempting High-Priority Emergency 911 Calls

You may configure SGAC rules and thresholds so that the high-priority voice (911) traffic receives an exclusive share of bandwidth. Because the average call volume for Emergency 911 traffic may not be very high, the fraction of bandwidth reserved for Emergency 911 calls may be small. In the case of regional emergency, the call volume of Emergency 911 calls may surge. In this case, it may be necessary to preempt some of the normal voice traffic to make room for surging Emergency 911 calls.

The Cisco CMTS software preempts one or more normal-priority voice flows to make room for the high-priority voice flows. SGAC provides the command-line interface (CLI) to enable or disable this preemption ability.

SGAC preemption logic follows the following steps:

1. When the first pass of admission control fails to admit a high priority PacketCable flow, it checks if it is possible to admit the flow in another bucket configured for normal PacketCable calls (applicable only if the PacketCable normal and high-priority rules are configured for different buckets). If the bandwidth is available, the call is admitted in the normal priority bucket.
2. If there is no room in normal priority bucket, it preempts a normal priority PacketCable flow and admits the high priority flow in the bucket where the low priority flow was preempted.

3. If there is no normal priority flow that it can preempt, it rejects the admission for high-priority flow. This usually happens when both normal and high-priority buckets are filled with 911 flows.

This preemption is effective only for PacketCable high-priority flows.

When a downstream low-priority service flow is chosen for preemption, the corresponding service flow for the same voice call in the opposite direction gets preempted as well.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **cable admission-control preempt priority-voice** **Example:** `Router(config)# no cable admission-control preempt priority-voice`	Changes the default Emergency 911 call preemption functions on the Cisco CMTS, supporting throughput and bandwidth requirements for Emergency 911 calls above all other buckets on the Cisco CMTS. The **no** form of this command disables this preemption, and returns the bucket that supports Emergency 911 calls to default configuration and normal function on the Cisco CMTS.

Step 4	**Ctrl-Z**
	Example:
	`Router(config)# Ctrl^Z`
	`Router#`
		Returns to Privileged EXEC mode.

Calculating Bandwidth Utilization

The SGAC feature maintains a counter for every US and DS channel, and this counter stores the current bandwidth reservation. Whenever a service request is made to create a new service flow, SGAC estimates the bandwidth needed for the new flow, and adds it to the counter. The estimated bandwidth is computed as follows:

- For DS service flows, the required bandwidth is the minimum reservation rate, as specified in the DOCSIS service flow QoS parameters.

In each of the above calculations, SGAC does not account for the PHY overhead. DOCSIS overhead is counted only in the UGS and UGS-AD flows. To estimate the fraction of bandwidth available, the calculation must account for the PHY and DOCSIS overhead, and also the overhead incurred to schedule DOCSIS maintenance

messages. SGAC applies a correction factor of 80% to the raw data rate to calculate the total available bandwidth.

Note For the DS and US flow in bonded channels, the maximum reserved bandwidth is the bandwidth defined for the SGAC threshold values. This value is indicated in kbps.

Enabling SGAC Check

A fiber node configured on the CMTS represents one or more matching physical fiber nodes in the HFC plant. The CMTS uses the fiber node configuration to identify the DOCSIS downstream service group (DS-SG) and DOCSIS upstream Service Group (US-SG) of the physical fiber nodes in the plant. The Service Group information is compared with MAC Domain channel configuration to automatically calculate the MAC Domain downstream and upstream service groups (MD-DS-SGs and MD-US-SGs respectively) within the MAC Domains.

Under each Fiber node, use the following procedure to enable SGAC check for an application type and any service flow of the specified application type, which is admitted to a service group.

Before you begin

Fairness Across DOCSIS Interfaces feature should always be enabled and the bandwidth percentage configured on each bonding group should be kept minimal to allow flexible adjustment of reservable bandwidth.

Restrictions

SGAC is supported only on the downstream.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configureterminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable fiber-node** *id* **Example:** `Router(config)# cable fiber-node 1`	Enters cable fiber-node configuration mode to configure a fiber node.
Step 4	**admission-control application-type** *n* **ds-bandwidth** *pct* **Example:**	Enables SGAC checking for the specified application-type. Use the **no** form of this command to disable SGAC checking.

	Command or Action	Purpose
	`Router(config-fiber-node)# admission-control application-type 1 ds-bandwidth 1`	
Step 5	Ctrl-Z **Example:** `Router(config-if)# Ctrl^Z`	Returns to Privileged EXEC mode.

What to do next

Use the **show cable admission-control fiber-node** *n* command to verify admission-control configuration.

Configuration Examples for SGAC

This section describes solutions-level examples of the SGAC feature on the Cisco CMTS. This section illustrates the functioning of SGAC in default or non-default operational configurations.

Example: SGAC Configuration Commands

In this section of configuration examples, the following SGAC parameters are set on the Cisco CMTS:

- All the packetcable flows are mapped into bucket 1.
- The BE service flows are mapped into bucket 8.

The following configuration commands enable these settings:

- To map the packetcable voice flows, use:

```
cable application-type 1 include packetcable normal
cable application-type 1 include packetcable priority
cable application-type 1 name PktCable
```

- To map the BE flows into bucket 8, use:

```
cable application-type 8 name HSD
cable application-type 8 include best-effort
```

- Given the above configurations, you may also control bandwidth allocation to a PCMM streaming video application. The streaming video application is identified by the PCMM application ID 35. The following commands implement this configuration:

```
cable application-type 2 name PCMM-Vid
cable application-type 2 include pcmm app-id 35
```

- These configurations may be verified on the Cisco CMTS using the following **show** commands:

```
Router# show cable application-type
For bucket 1, Name PktCable
     Packetcable normal priority gates
     Packetcable high priority gates
For bucket 2, Name PCMM-Vid
```

```
     PCMM gate app-id = 30
For bucket 3, Name Gaming
     PCMM gate app-id = 40
For bucket 4, Name
For bucket 5, Name
For bucket 6, Name
For bucket 7, Name
For bucket 8, Name HSD
     Best-effort (CIR) flows
```

```
Router# show cable admission-control fiber-node 1
App-type    Name       Exclusive
1                      N/A
2                      N/A
3           Normal     10%
4                      N/A
5                      N/A
6                      N/A
7           Emergency  N/A
8                      N/A
```

```
Router#show cable admission-control interface integrated-Cable 8/0/0:0

Interface In8/0/0:0
RFID 24576

Resource - Downstream Bandwidth
-------------------------------
App-type    Name       Reservation/bps    Exclusive      Rejected
1                      0                  N/A            0
2                      0                  N/A            0
3           Normal     0                  10%            0
4                      0                  N/A            0
5                      0                  N/A            0
6                      0                  N/A            0
7           Emergency  0                  N/A            0
8                      0                  N/A            0
Max Reserved BW = 300000 bps
Total Current Reservation = 0 bps
Guaranteed Bonus BW = 21055000 bps
Non-guaranteed Bonus BW = 7744000 bps
Superset BGs: Wi8/0/0:0 Wi8/0/0:4 Wi8/0/0:6
```

```
Router#show cable admission-control interface wideband-Cable 8/0/0:0

Interface Wi8/0/0:0
BGID: 24577

Resource - Downstream Bandwidth
-------------------------------
App-type    Name       Reservation/bps    Exclusive      Rejected
1                      0                  N/A            0
2                      0                  N/A            0
3           Normal     0                  10%            0
4                      0                  N/A            0
5                      0                  N/A            0
6                      0                  N/A            0
7           Emergency  0                  N/A            0
8                      0                  N/A            0
Max Reserved BW = 600000 bps
Total Current Reservation = 0 bps
Guaranteed Bonus BW = 21055000 bps
```

```
Non-guaranteed Bonus BW = 36844000 bps
Subset BGs: In8/0/0:0 In8/0/0:1
Superset BGs: Wi8/0/0:4 Wi8/0/0:6
Overlapping BGs: N/A
```

These above configuration examples might be omitted or changed, but the remaining examples in this section presume the above configurations.

Example: SGAC for Downstream Traffic

This example presumes that you have configured the rules according to the commands illustrated at the start of this section.

- All the voice flows in bucket 1.
- All the CIR data flows are categorized in bucket 8.

The below example illustrates a sample configuration for SGAC with downstream traffic. In this example, if voice traffic exceeds 30% bandwidth consumption, additional voice flows are denied.

- 30% downstream throughput is reserved exclusively for voice traffic.

The following command implements this configuration:

```
Router(config-fiber-node)#admission-control application-type 1 ds-bandwidth 30
```

The below example illustrates how flexible bandwidth allocation is configured. In this example, normal voice traffic (application-type 1) is associated with two thresholds. Normal voice traffic alone can use up to 40% of the service group's capacity, while normal and emergency voice traffic combined can use up to 50% of the service group’s capacity. This means that emergency voice traffic can have at least 10% of the service group's capacity, even if normal voice traffic has used up its share of 40%:

```
Router(config-fiber-node)#admission-control application-type 1 ds-bandwidth 40
Router(config-fiber-node)#admission-control application-type 1-2 ds-bandwidth 50
```

where,

- 1 is normal voice application type
- 2 is emergency voice application type

Additional References

The following topics provide references related to SGAC for the Cisco CMTS.

Related Documents

Related Topic	Document Title
Cisco CMTS Cable Commands	Cisco CMTS Cable Command Reference

Standards

Standard	Title
CableLabs™ DOCSIS 1.1 specifications	http://www.cablelabs.com/cablemodem/
CableLabs™ PacketCable specifications	http://www.cablelabs.com/packetcable/
CableLabs™ PacketCable MultiMedia specifications	http://www.cablelabs.com/packetcable/specifications/multimedia.html

MIBs

MIB	MIBs Link
MIBs	To locate and download MIBs for selected platforms, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

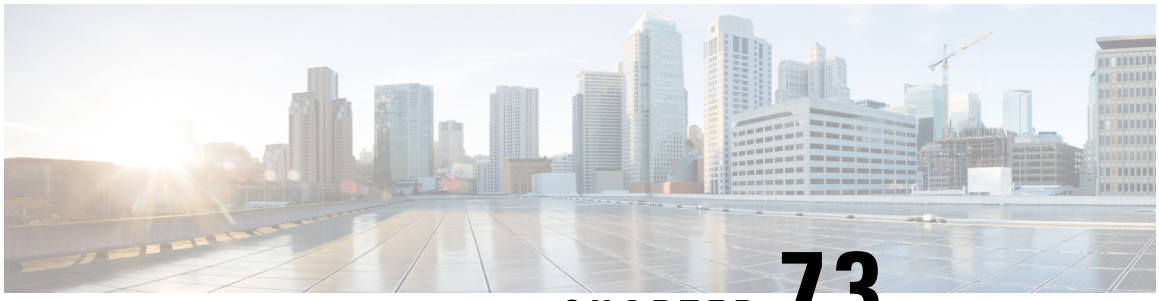
Feature Information for Service Group Admission Control

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 184: Feature Information for Service Group Admission Control

Feature Name	Releases	Feature Information
Service group admission control	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on Cisco cBR Series Converged Broadband Router s.



CHAPTER 73

Subscriber Traffic Management

This document describes the Subscriber Traffic Management (STM) feature Version 1.3. STM feature supports all DOCSIS-compliant cable modems.

The STM feature allows a service provider to configure a maximum bandwidth threshold over a fixed period for a specific service class (or quality of service [QoS] profile)). The subscribers who exceed this configured threshold can then be identified and allocated reduced QoS. STM works as a low-CPU alternative to Network-Based Application Recognition (NBAR) and access control lists (ACLs). However, using STM does not mean that NBAR and ACLs have to be turned off; STM can be applied along with NBAR and ACLs. STM also works in conjunction with the Cisco Broadband Troubleshooter to support additional network management and troubleshooting functions in the Cisco CMTS.

Important In this document, the phrase QoS profile is synonymously used to indicate a service class for a DOCSIS 1.1 cable modem. However, QoS profile applies only to DOCSIS 1.0 operations. In instances where QoS profile is mentioned to indicate DOCSIS1.1 operations, the QoS profile should be treated as a service class.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1108
- Restrictions for Subscriber Traffic Management on the Cisco CMTS Routers, on page 1108
- Information About Subscriber Traffic Management on the Cisco CMTS Routers, on page 1109
- How to Configure the Subscriber Traffic Management Feature on the Cisco CMTS Routers, on page 1114
- Monitoring the Subscriber Traffic Management Feature on the Cisco CMTS Routers, on page 1125
- Configuration Examples for Subscriber Traffic Management on the Cisco CMTS Routers, on page 1128
- Additional References, on page 1131
- Feature Information for Subscriber Traffic Management, on page 1133

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 185: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for Subscriber Traffic Management on the Cisco CMTS Routers

Note In this document, the phrase QoS profile is synonymously used to indicate a service class for a DOCSIS 1.1 cable modem. However, QoS profile applies only to DOCSIS 1.0 operations. In instances where QoS profile is mentioned to indicate DOCSIS 1.1 operations, the QoS profile should be treated as a service class.

The STM feature has the following restrictions and limitations:

- In STM version 1.1, the sampling rate range (duration) is calculated using the monitoring duration rather than the constant range (10 to 30 minutes) used in STM 1.0.
  - If the monitoring duration is more than a day (1440 minutes), the duration sample rate is calculated as (duration / 100).
  - If the monitoring duration is less than a day, the sample rate range is from 10 to 30 minutes.
  - If you are using STM 1.0 with a duration of two days and a sample rate of 20 minutes, and you try to restore that configuration in STM 1.1, the command fails because now the valid range is from 28 to 86 minutes.
- For DOCSIS1.0, the registered QoS profile specified by an enforce-rule must match exactly a QoS profile that exists on the Cisco CMTS. To manage a cable modem that is using a modem-created QoS profile, you must first create that same exact QoS profile on the Cisco CMTS. All parameters in the QoS profile must match before the cable modem can be managed by the enforce-rule.
- The Cisco cBR series routers support a certain maximum of 40 enforce-rules. If you have created the maximum number of enforce-rules and want to create another rule, you must first delete one of the existing rules.
- Changing the configuration of an enforce-rule automatically resets all byte counters for the subscribers who are mapped to that enforce-rule.
- When specifying a QoS profile to be enforced when users violate their registered QoS profiles, both the originally provisioned QoS profile and the enforced QoS profile must be created on the Cisco CMTS.
- The Subscriber Traffic Management feature calculates duration based on the time set on the router, not uptime. Therefore, if you use the **clock set** command to change the time on the router, you might affect the STM monitoring behavior.
- The maximum cycle for subscriber traffic management is 31 days. If you choose a cycle of 31 days, the minimum sample rate that you can set is (31 days/100) minutes.

Information About Subscriber Traffic Management on the Cisco CMTS Routers

This section contains the following:

Feature Overview

The STM feature allows service providers to configure a maximum bandwidth threshold over a fixed period, for a specific service class (or QoS profile). The subscribers who exceed this configured threshold can then be identified and allocated a reduced QoS. This feature supplements current techniques such as NBAR and ACLs, to ensure that a minority of users do not consume a majority of a cable network's bandwidth.

Current subscriber controls, such as NBAR and ACLs, examine all packets coming into the CMTS. These techniques can curb a large volume of problem traffic, but they are not as effective in dealing with the latest generation of peer-to-peer file-sharing applications that can place heavy demands on a network's available bandwidth.

The STM feature allows service providers to focus on a minority of potential problem users without impacting network performance or other users who are abiding by their service agreements.

The STM feature supports two types of monitoring:

- Legacy Monitoring—Legacy monitoring allows you to set up a single monitoring duration without the ability to choose the time of day when that monitoring is performed. The configured monitoring parameters remain constant throughout the day.
- Peak-Offpeak Monitoring—Peak-Offpeak monitoring allows you to specify up to two high-traffic periods in a day for monitoring, in addition to the ability to continue monitoring during the remaining (or off-peak) periods. By combining the peak time option with weekend monitoring, you can identify and limit the bandwidth usage of certain subscribers for up to two peak network usage periods during weekdays, and during a different set of peak usage periods on weekends.

When a cable modem goes offline and remains offline for 24 hours, the Cisco CMTS router deletes its service flow IDs from its internal databases, and also deletes the modem's traffic counters. This can allow some users to exceed their bandwidth limits, go offline, and come back online with new counters. The Subscriber Traffic Management feature helps to thwart these types of theft-of-service attacks by implementing a penalty period for cable modems that violate their service level agreements (SLAs). Even if a cable modem goes offline, its counters are still reset, and the CMTS continues to enforce the penalty period.

Feature List

The Subscriber Traffic Management feature has the following operational features:

- Subscriber Traffic Management 1.1 (STM 1.1) supports cable modems that have registered for DOCSIS 1.1 operations (using the service class/service flow ID [SFID] model).
- Up to 40 enforce-rules can be created on each router.
- Separate enforce-rules can be used for downstream traffic and for upstream traffic. However, the limit on the total number of enforce-rules that can be configured includes the upstream and downstream rules combined.
- Each enforce-rule uses a subscriber's registered QoS profile to identify which users should be monitored for excessive traffic for DOCSIS1.0 cable modems. The registered QoS profile must exist on the Cisco CMTS. If you want to manage cable modems that are using QoS profiles that were created by the cable modem, you must first manually create a QoS profile with the exact same QoS parameters on the Cisco CMTS, and then allow the cable modem to come online using the manually created profile.
- Each enforce-rule specifies the maximum number of kilobytes a user can transmit during a specified window.
- Subscribers who exceed the maximum bandwidth that is specified by their enforce-rule can be automatically switched to a separate enforced QoS profile that limits their network use for a customizable penalty period. The enforced QoS profile can change the guaranteed bandwidth, priority, or any other aspect of the traffic that the service provider considers an acceptable response to subscribers who violate their service agreements.
- Subscribers are automatically switched back to their registered QoS profile at the end of their penalty period. A technician at the service provider's network operations center (NOC) can also switch them back before the penalty period expires.

Note To manually switch back, delete the cable modem and allow it to register again.

- This feature also supports a **no-persistence** option, so that the enforced QoS profile does not remain in effect when a cable modem reboots. This option is particularly useful when the feature is initially implemented, so that the service providers can identify problem subscribers and applications, without

creating a major impact on the entire user base. When repeat offenders are found, they can then be switched to an enforce-rule that does keep the enforced QoS profile in effect even when the cable modem reboots.

- Service providers can display a list of all subscribers' current usage statistics. Service providers can also display a list of just those subscribers who are overconsuming bandwidth.
- The penalty period persists across reboots of the cable modem, so subscribers cannot avoid the enforced QoS profile by resetting their modems and reregistering on the cable network. This allows service providers to set an appropriate penalty for those users that consistently exceed the maximum bandwidth they have been allocated. Service providers also can specify a time of day when CMs that are identified for penalty can be released from the penalty period.
- If a user that is using excessive bandwidth decides to upgrade to a higher level of service, the service provider can reconfigure the provisioning system to assign a new QoS profile to the cable modem. The user can then reboot the cable modem and come online using the new level of service.
- Service providers can change subscriber service classes for a particular modem using the **cable modem service-class-name** command.
- Different subscriber monitoring parameters can be configured for weekends, including peak and offpeak monitoring windows. You can also establish the same monitoring windows for every day of the week, or turn off monitoring altogether on the weekends as desired.

Sliding Window for Monitoring Service Flows

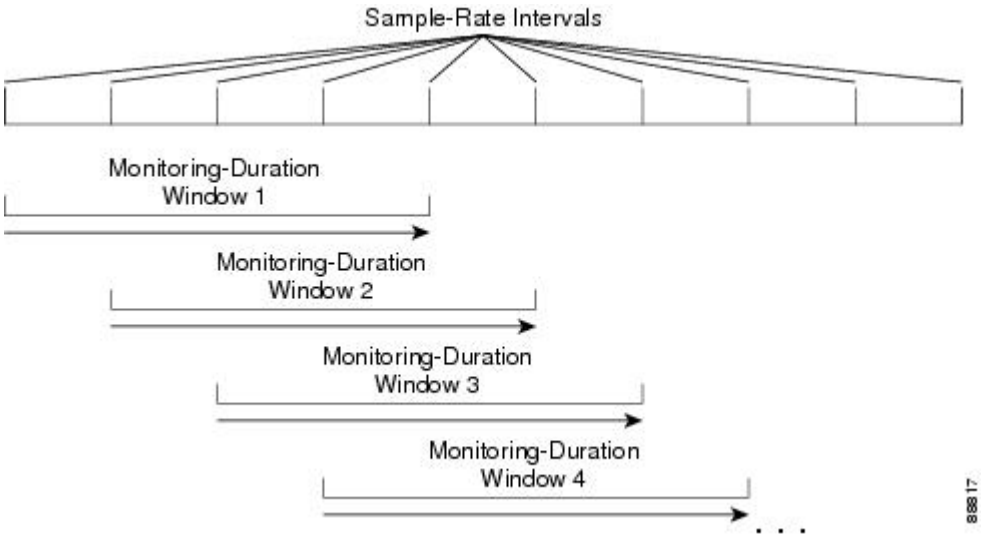
When an enforce-rule is activated, the CMTS periodically checks the bandwidth being used by subscribers to determine whether any subscribers are consuming more bandwidth than that specified by their registered QoS profiles. The CMTS keeps track of the subscribers using a sliding window that begins at each sample-rate interval and continues for the monitoring-duration period.

Each sample-rate interval begins a new sliding window period for which the CMTS keeps track of the total bytes transmitted. At the end of each sliding window period, the CMTS examines the byte counters to determine if any subscriber is currently overconsuming bandwidth on the network.

For example, with the default sample-rate interval of 15 minutes and the default monitoring-duration window of 360 minutes (6 hours), the CMTS samples the bandwidth usage every 15 minutes and determines the total bytes transmitted at the end of each 360-minute window. Therefore, every 15 minutes, the CMTS determines each subscriber's usage statistics for the preceding 6-hour period.

Figure below illustrates how this process works, with a new sliding window beginning at the beginning of each sample-rate interval period.

Figure 33: Monitoring-Duration Windows



Weekend Monitoring

With standard legacy and peak-offpeak monitoring configuration, monitoring continues to occur on the weekends.

STM version 1.2 supports configuration of different monitoring conditions on weekends. Weekend monitoring options support the same parameters that are available in the existing monitoring options, but use a separate set of commands to configure alternate monitoring on weekend days. This includes configuration of peak and offpeak weekend monitoring windows.

In addition, the CLI supports the ability to turn off any monitoring on the weekend, or to use the same monitoring conditions for every day of the week.

SNMP Trap Notifications

Simple Network Management Protocol (SNMP) trap notification can be sent whenever a subscriber violates the enforce-rule. This trap is defined in the CISCO-CABLE-QOS-MONITOR-MIB and is enabled using the **snmp-server enable traps cable** command.

Each SNMP trap notification contains the following information:

- MAC address of the subscriber's cable modem
- Name of the enforce-rule being applied to this subscriber
- Total bytes sent by the subscriber during the monitoring-duration window
- Time at which the subscriber's penalty period expires

The CISCO-CABLE-QOS-MONITOR-MIB also contains the following tables that provide information about the Subscriber Traffic Management configuration and about subscribers who violate their enforce-rules:

- ccqmCmtsEnforceRuleTable—Contains the attributes of the enforce-rules that are currently configured on the Cisco CMTS.
- ccqmEnfRuleViolateTable—Provides a snapshot list of the subscribers who violated their enforce-rules over the sliding monitoring-duration window.

The following objects are used for enforce rules:

- ccqmCmtsEnfRulePenaltyEndTime
- ccqmCmtsEnfRuleWkndOff
- ccqmCmtsEnfRuleWkndMonDuration
- ccqmCmtsEnfRuleWkndAvgRate
- ccqmCmtsEnfRuleWkndSampleRate
- ccqmCmtsEnfRuleWkndFirstPeakTime
- ccqmCmtsEnfRuleWkndFirstDuration
- ccqmCmtsEnfRuleWkndFirstAvgRate
- ccqmCmtsEnfRuleWkndSecondPeakTime
- ccqmCmtsEnfRuleWkndSecondDuration
- ccqmCmtsEnfRuleWkndSecondAvgRate
- ccqmCmtsEnfRuleWkndOffPeakDuration
- ccqmCmtsEnfRuleWkndOffPeakAvgRate
- ccqmCmtsEnfRuleWkndAutoEnforce
- ccqmCmtsEnfRuleFirstPeakTimeMin
- ccqmCmtsEnfRuleSecondPeakTimeMin
- ccqmCmtsEnfRuleWkndFirstPeakTimeMin
- ccqmCmtsEnfRuleWkndSecondPeakTimeMin
- ccqmCmtsEnfRulePenaltyEndTimeMin
- ccqmCmtsEnfRuleWkPenaltyPeriod
- ccqmCmtsEnfRuleWkndPenaltyPeriod
- ccqmCmtsEnfRuleRelTimeMonitorOn

The following objects are used for enforce rule violations:

- ccqmEnfRuleViolateID
- ccqmEnfRuleViolateMacAddr
- ccqmEnfRuleViolateRuleName
- ccqmEnfRuleViolateByteCount
- ccqmEnfRuleViolateLastDetectTime
- ccqmEnfRuleViolatePenaltyExpTime
- ccqmEnfRuleViolateAvgRate

Cable Modem Interaction with the Subscriber Traffic Management Feature

The Subscriber Traffic Management feature ensures that users cannot bypass the QoS restrictions by rebooting their cable modems or performing other configuration changes. The service provider, however, continues to be able to change the modems' profiles and other configuration parameters as desired.

When the Subscriber Traffic Management feature is enabled, the following behavior is in effect:

- The primary service flow counters for downstream and upstream traffic are preserved when the cable modem reboots. The service provider, however, can reset the counters by changing the QoS profile for the cable modem using the **cable modem qos profile** command and resetting the cable modem.

- Secondary service flow counters are reset whenever the cable modem reboots. This happens regardless of the enforce-rule configuration.
- The cable modem retains its current primary downstream and upstream service flows when it reboots. If the cable modem is in an enforced QoS profile penalty period when it reboots, it continues using the enforced QoS profile after the reboot. Service providers can manually change the profile by assigning a new QoS profile using the **cable modem qos profile** command.

Note Changing the QoS profile for a cable modem using the **cable modem qos profile** command, also changes the enforce-rule for the cable modem when it reboots. When the cable modem comes back online, it begins operating under the enforce-rule whose registered QoS profile (see the **qos-profile registered** command) matches the new QoS profile the modem is using.

- Service providers can also change the enforce-rule configuration. The following happens when the provider changes the enforce-rule configuration:
  - If the enforce-rule is disabled (using the **no enabled** command), all cable modems using that rule's registered QoS profile are no longer managed by the Subscriber Traffic Management feature. Configuring no enabled, deactivates the enforce-rule and moves all the modems in penalty to its registered QoS.
  - If the registered QoS profile for the rule is changed (using the **qos-profile registered** command), the cable modems that are using the previous registered QoS profile are no longer managed by the Subscriber Traffic Management feature. Instead, any cable modems that use the new registered QoS profile begin being managed by this rule.
  - If the enforced QoS profile for the rule is changed (using the **qos-profile enforced** command), any cable modems using this rule that are currently in the penalty period continue using the previously configured enforced QoS profile. Any cable modems that enter the penalty period after this configuration change, however, use the new enforced QoS profile.
- Service providers also have the option of making an enforce-rule nonpersistent, so that the enforced QoS profile does not remain in force when a cable modem reboots. Instead, when the cable modem reboots and reregisters with the Cisco CMTS, the CMTS assigns it the QoS profile that is specified in its DOCSIS configuration file.

How to Configure the Subscriber Traffic Management Feature on the Cisco CMTS Routers

This section contains the following:

Creating and Configuring an Enforce-Rule

Every service class name that needs to be monitored will be linked with an enforce-rule. An enforce-rule defines the monitoring duration, the sample rate, the penalty period, and the registered service class name that the enforce-rule is linked to and the enforced service class name.

Use the procedure given below to create and configure an enforce-rule. An enforce-rule does not become active until the **enabled** command is given.

Before you begin

- The registered and enforced service (QoS) profiles must be created on the CMTS before creating an enforce-rule that uses those profiles. If you want to manage a cable modem that currently uses a modem-created QoS profile, you must first manually create a new QoS profile on the CMTS with the same QoS parameters as the modem-created profile. Then allow the modem to come online using the manually created profile before beginning this procedure.
  - To display quality of service (QoS) profiles for a Cisco CMTS, use the show cable qos profile command in privileged EXEC mode.
  - To configure a QoS profile, use the cable qos profile command in global configuration mode. To set a particular value to its default, or to delete the profile when no specific parameters have been set, use the no form of this command.
- For monitoring of DOCSIS 1.1 cable modems:
  - Only DOCSIS 1.1 modems that register with a service class name are monitored.
  - To ensure that the DOCSIS 1.1 service flow counters remain across a reboot of the CM, configure the **cable primary-sflow-qos11 keep all** global configuration command.
- Only primary upstream and downstream service flows are supported.

Restriction

- When configuring peak-offpeak monitoring, you can define a maximum of two peak durations within a day, and also monitoring of the remaining hours, if you configure the offpeak duration. The monitoring duration and threshold for first peak, second peak, and offpeak, can be different. However, the monitoring duration for any peak or offpeak configuration cannot be more than a day.
- The parameters defined by the named service class should always be a compatible subset of the registered set of parameters for the CM. Only certain options can be changed using a CMTS router service class, such as the **max-rate**, **priority**, or **tos-overwrite** options. The **max-burst** option in both the enforced and registered CMTS router service classes must strictly match the value for **max-burst** in the registered DOCSIS configuration file. If the service class value does not match, either the cable modem registration will fail with a reject-c state, or the enforced class will fail.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters the global configuration mode.

	Command or Action	Purpose
Step 3	**cable qos enforce-rule** *name* **Example:** `Router(config)# cable qos enforce-rule test`	Creates an enforce-rule with the specified *name* and enters the enforce-rule configuration mode. **Note** Each enforce-rule can be created by giving it a name.
Step 4	**monitoring-basics** {**legacy** | **peak-offpeak**} {**docsis10** | **docsis11**} **Example:** `Router(enforce-rule)# monitoring-basics peak-offpeak docsis11`	Defines the kind of monitoring desired and the type of modems to be monitored. The default is legacy and DOCSIS 1.0.
Step 5	Perform one of the following: • If you specified DOCSIS 1.0 cable modems in Step 4, on page 1116, use the following commands: **a.** **qos-profile registered** *profile-id* **b.** **qos-profile enforced** *profile-id* [**no-persistence**] • If you specified DOCSIS 1.1 cable modems in Step 4, on page 1116, use the **service-class** {**enforced** | **registered**} *name* command. **Example:** `Router(enforce-rule)# service-class enforced test`	• For DOCSIS 1.0 cable modems: **a.** Specifies the registered quality of service (QoS) profile that should be used for this enforce-rule. **Note** If you want to manage a cable modem that currently uses a modem-created QoS profile, you must first manually create a new QoS profile on the CMTS with the same QoS parameters as the modem-created profile. Then allow the modem to come online using the manually created profile before using this command. **b.** Specifies the quality of service (QoS) profile that should be enforced when users violate their registered QoS profiles for DOCSIS 1.0 cable modems. • For DOCSIS 1.1 (and later) cable modems, identifies a particular service class with the specified *name* for cable modem monitoring in an enforce-rule.
Step 6	**duration** *minutes* **avg-rate** *rate* **sample-interval** *minutes*[**penalty** *minutes*] {**downstream** | **upstream**} [**enforce**] **Example:**	Specifies the time period and sample rate used for monitoring subscribers when legacy monitoring is configured (Step 4, on page 1116).

	Command or Action	Purpose
	`Router(enforce-rule)# duration 10 avg-rate 500 sample-interval 10 penalty 120 downstream enforce`	

Step 7	**peak-time1** {*hour* | *hour:minutes*} **duration** *minutes* **avg-rate** *rate* [**peak-time2** {*hour* | *hour:minutes*} **duration** *minutes* **avg-rate** *rate*][**duration** *offpeak-minutes* **avg-rate** *offpeak-rate*] **sample-interval** *minutes*[**penalty** *minutes*] {**downstream** | **upstream**}[**enforce**]
	Example:
	`Router(enforce-rule)# peak-time1 6 duration 180 avg-rate 2 peak-time2 18 duration 180 avg-rate 2 duration 120 avg-rate 3 sample-interval 10 upstream enforce`
	`Router(enforce-rule)# peak-time1 6:30 duration 180 avg-rate 2 peak-time2 18:40 duration 180 avg-rate 2 duration 120 avg-rate 3 sample-interval 10 penalty 120 upstream enforce`
		Specifies peak monitoring periods when peak-offpeak monitoring is configured (Step 4, on page 1116).

Step 8	**penalty-period** *minutes* [**time-of-day** {*hour*|*hour*:*minutes*}] [**monitoring-on**] **Example:** `Router(enforce-rule)# penalty-period 10`	(Optional) Specifies the period for which an enforced QoS profile should be in effect for subscribers who violate their registered QoS profiles.
Step 9	**enabled** **Example:** `Router(enforce-rule)# enabled`	(Optional) Activates the enforce-rule and begins subscriber traffic management.
Step 10	**end** **Example:** `Router(enforce-rule)# end`	Exits enforce-rule configuration mode and returns to privileged EXEC mode.

Examples

This section provides command-line interface (CLI) examples, including the help feature for some of the enforce-rule commands.

Example: Legacy Monitoring Configuration

The following example shows a sample configuration of an enforce-rule for legacy monitoring:

```
Router(config)# cable qos enforce-rule test
Router(enforce-rule)# monitoring-basics ?
  legacy        Enable legacy (same average rate for all day)  monitoring
  peak-offpeak  Enable peak-offpeak monitoring
Router(enforce-rule)# monitoring-basics legacy ?
  docsis10  Enforce-rule will map to docsis 1.0 modems
  docsis11  Enforce-rule will map to docsis 1.1 modems
Router(enforce-rule)# monitoring-basics legacy docsis11
Router(enforce-rule)# service-class ?
  enforced    Enforced service class
  registered  Registered service class
Router(enforce-rule)# service-class registered ?
  WORD  Registered service class name
Router(enforce-rule)# service-class registered BEUS
Router(enforce-rule)# service-class enforced test
Router(enforce-rule)# duration ?
  <10-10080>  Duration in minutes
Router(enforce-rule)# duration 10 ?
  avg-rate  Average rate for the duration in kbits/sec
Router(enforce-rule)# duration 10 avg-rate ?
  <1-4294967>  average rate in kbits/sec
Router(enforce-rule)# duration 10 avg-rate 2 ?
  sample-interval  Rate of sampling in Minutes
Router(enforce-rule)# duration 10 avg-rate 2 sample-interval ?
  <1-30>  Sampling rate in Minutes
Router(enforce-rule)# duration 10 avg-rate 2 sample-interval 10 ?
  downstream  downstream
  upstream    upstream
Router(enforce-rule)# duration 10 avg-rate 2 sample-interval 10 upstream ?
  enforce  enforce the qos-profile automatically
  <cr>
Router(enforce-rule)# duration 10 avg-rate 2 sample-interval 10 upstream enf
Router(enforce-rule)# $ avg-rate 2 sample-interval 10 upstream enforce
Router(enforce-rule)# enabled
Router(enforce-rule)# end
```

Example: Peak-offpeak Monitoring Configuration

The following example shows a sample configuration of an enforce-rule for peak-offpeak monitoring:

```
Router(config)# cable qos enforce-rule test
Router(enforce-rule)# monitoring-basics peak-offpeak
Router(enforce-rule)# monitoring-basics peak-offpeak docsis10
Router(enforce-rule)# qos-profile ?
  enforced    Enforced qos profile
  registered  QoS profile index
Router(enforce-rule)# qos-profile registered ?
  <1-255>  Registered QoS profile index
Router(enforce-rule)# qos-profile registered 5
Router(enforce-rule)# qos-profile enforced 4
Router(enforce-rule)# peak-time1 6 ?
  duration  First peak duration
Router(enforce-rule)# peak-time1 6 duration ?
  <60-1440>  Duration in minutes
Router(enforce-rule)# peak-time1 6 duration 180 ?
  avg-rate  First peak average rate in kbits/sec
Router(enforce-rule)# peak-time1 6 duration 180 avg-rate ?
  <1-4294967>  Average rate in kbits/sec
Router(enforce-rule)# peak-time1 6 duration 180 avg-rate 2 ?
  duration         Off-peak duration
```

```
  peak-time2       Second peak time
  sample-interval  Rate of sampling in minutes


Router(enforce-rule)# peak-time1 6 duration 180 avg-rate 2 peak-time2 ?
  <10-1440>  Start of second peak time
Router(enforce-rule)# peak-time1 6 duration 180 avg-rate 2 peak-time2 18 ?
  duration  Second peak duration
Router(enforce-rule)# $6 duration 180 avg-rate 2 peak-time2 18 duration ?
  <10-1440>  Duration in minutes
Router(enforce-rule)# $6 duration 180 avg-rate 2 peak-time2 18 duration 240 ?
  avg-rate  Second peak average rate in kbits/sec
Router(enforce-rule)# $ 180 avg-rate 2 peak-time2 18 duration 240 avg-rate ?
  <1-4294967>  Average rate in kbits/sec
Router(enforce-rule)# $ 180 avg-rate 2 peak-time2 18 duration 240 avg-rate 3 ?
  duration         Off-peak duration
  sample-interval  Rate of sampling in minutes
Router(enforce-rule)# $ 180 avg-rate 2 peak-time2 18 duration 240 avg-rate 3 d
Router(enforce-rule)# $-time2 18 duration 240 avg-rate 3 duration 120 ?
  avg-rate  Off-peak average rate in kbits/sec
Router(enforce-rule)# $duration 240 avg-rate 3 duration 120 avg-rate 1 ?

  sample-interval  Rate of sampling in minutes
Router(enforce-rule)# $40 avg-rate 3 duration 120 avg-rate 1 sample-interval ?
  <1-30>  Sampling rate in Minutes
Router(enforce-rule)# $e 3 duration 120 avg-rate 1 sample-interval 10 ?

  downstream  downstream
  upstream    upstream
Router(enforce-rule)# $e 3 duration 120 avg-rate 1 sample-interval 10 upstream ?
  enforce  enforce the qos-profile automatically
  <cr>
Router(enforce-rule)# $on 120 avg-rate 1 sample-interval 10 upstream enforce
Router(enforce-rule)# enabled
Router(enforce-rule)# end
```

Configuring Weekend Monitoring

This section describes the tasks required to configure weekend monitoring for STM on a Cisco CMTS router.

Prerequisites

You must first configure the weekday monitoring parameters for an enforce-rule before configuring weekend monitoring. See the Creating and Configuring an Enforce-Rule, on page 1114.

Restrictions

- Up to 40 total enforce-rules across both upstream and downstream configurations are supported.
- When using SNMP for weekend monitoring, only SNMP GET and GETMANY operations are supported.

Configuring Different Legacy Monitoring Conditions for Weekends

Use the following procedure if you want to establish different legacy monitoring conditions for subscribers for either upstream or downstream traffic on weekend days.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable qos enforce-rule** *name* **Example:** `Router(config)# cable qos enforce-rule test`	Accesses the enforce-rule with the specified *name* and enters enforce-rule configuration mode.
Step 4	**weekend duration** *minutes* **avg-rate** *rate* **sample-interval** *minutes* {**downstream** | **upstream**} [**penalty** *minutes*] [**enforce**] **Example:** `Router(enforce-rule)# weekend duration 15 avg-rate 500 sample-interval 10 penalty 120 downstream enforce`	Specifies the time period and sample rate used for monitoring subscribers on weekends.
Step 5	**end** **Example:** `Router(enforce-rule)# end`	Exits enforce-rule configuration mode and returns to privileged EXEC mode.

Configuring Different Peak-Offpeak Monitoring Conditions for Weekends

Use the following procedure if you want to establish different peak and offpeak monitoring conditions for subscribers for either upstream or downstream traffic on weekend days.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**cable qos enforce-rule** *name* **Example:** `Router(config)# cable qos enforce-rule test`	Accesses the enforce-rule with the specified *name* and enters enforce-rule configuration mode.

Step 4	**weekend peak-time1** {*hour* | *hour*:*minutes*} **duration** *minutes* **avg-rate** *rate* [**peak-time2** *hour* **duration** *minutes* **avg-rate** *rate*] [**duration** *offpeak-minutes* **avg-rate** *offpeak-rate*] **sample-interval** *minutes*[**penalty** *minutes*] {**downstream**| **upstream**}[**enforce**]
	Example:
	`Router(enforce-rule)# weekend peak-time1 9 duration 180 avg-rate 2 peak-time2 16 duration 180 avg-rate 2 duration 120 avg-rate 3 sample-interval 10 upstream enforce`
	Example:
	`Router(enforce-rule)# weekend peak-time1 9:30 duration 180 avg-rate 2 peak-time2 16:58 duration 180 avg-rate 2 duration 120 avg-rate 3 sample-interval 10 penalty 120 upstream enforce`
		Specifies peak and offpeak monitoring times on weekends.

Step 5	**end** **Example:** `Router(enforce-rule)# end`	Exits enforce-rule configuration mode and returns to privileged EXEC mode.

Disabling Weekend Monitoring

Use the following procedure to turn off the weekend monitoring configuration and monitor on weekdays only.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**cable qos enforce-rule** *name* **Example:** `Router(config)# cable qos enforce-rule test`	Accesses the enforce-rule with the specified *name* and enters enforce-rule configuration mode.
Step 4	**weekend off** **Example:** `Router(enforce-rule)# weekend off`	Disables monitoring on weekends.
Step 5	**end** **Example:** `Router(enforce-rule)# end`	Exits enforce-rule configuration mode and returns to privileged EXEC mode.

Removing Weekend Monitoring Conditions and Use the Same Monitoring Criteria Every Day

Use the following procedure to remove the specified weekend monitoring conditions and use the same monitoring criteria all week (including weekends).

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable qos enforce-rule** *name* **Example:** `Router(config)# cable qos enforce-rule test`	Accesses the enforce-rule with the specified *name* and enters enforce-rule configuration mode.
Step 4	**no weekend** **Example:** `Router(enforce-rule)# no weekend`	Performs monitoring on the weekends using the same parameters for weekdays and weekends.

	Command or Action	Purpose
Step 5	**end** **Example:** `Router(enforce-rule)# end`	Exits enforce-rule configuration mode and returns to privileged EXEC mode.

Disabling an Enforce-Rule

Use the following procedure to disable an enforce-rule. The enforce-rule remains in the CMTS configuration file, but any subscriber traffic management that uses this enforce-rule ends.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable qos enforce-rule***name* **Example:** `Router(config)# cable qos enforce-rule test`	Creates an enforce-rule with the specified *name* and enters enforce-rule configuration mode.
Step 4	**no enabled** **Example:** `Router(enforce-rule)# no enabled`	Disables the enforce-rule and ends subscriber traffic management for users with the rule's registered QoS profile. It moves all modems in penalty to its registered QoS.
Step 5	**end** **Example:** `Router(enforce-rule)# end`	Exits enforce-rule configuration mode and returns to privileged EXEC mode.

Removing an Enforce-Rule

Use the following procedure to delete an enforce-rule and remove it from the CMTS configuration file. Any subscriber traffic management that uses this rule also ends.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no cable qos enforce-rule***name* **Example:** `Router(config)# no cable qos enforce-rule ef-rule`	Deletes the enforce-rule with the specified *name* . This enforce-rule and its configuration are removed from the CMTS configuration, and any subscriber traffic management that uses this rule ends.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Changing a Cable Modem Service Class

Use the following procedure to change a QoS service class for a particular DOCSIS 1.1 cable modem.

Restriction

- The command is supported only on DOCSIS 1.1 CM primary service flows.
- You can specify the **cable modem service-class-name** command only after the CM has been online for at least 200 seconds.
- The parameters defined by the named service class should always be a compatible subset of the registered set of parameters for the CM. Only certain options can be changed using a CMTS router service class, such as the **max-rate**, **priority**, or **tos-overwrite** options. The **max-burst** option in both the enforced and registered CMTS router service classes must strictly match the value for **max-burst** in the registered DOCSIS configuration file. If the service class value does not match, then CM registration will fail with a reject-c state, or the enforced class will fail.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	**cable modem** {*ip-address* | *mac-address*} **service-class-name** *name* **Example:** `Router# cable modem aaaa.bbbb.cccc service-class-name test`	Changes a QoS service class for a particular cable modem.
Step 3	**end** **Example:** `Router# end`	Exits privileged EXEC mode.

Monitoring the Subscriber Traffic Management Feature on the Cisco CMTS Routers

This section describes the following tasks that can be used to monitor the Subscriber Traffic Management feature:

Displaying the Currently Defined Enforce-Rules

To display all enforce-rules that are currently defined on the Cisco CMTS router, or the definitions for a particular enforce-rule, use the **show cable qos enforce-rule** command in privileged EXEC mode.

For offpeak monitoring, use the **show cable qos enforce-rule** command to display the monitoring duration and average-rate values applicable for that time of day. If no monitoring is taking place, 0 is displayed.

The following example shows sample output from the **show cable qos enforce-rule** command for all configured enforce-rules:

```
Router# show cable qos enforce-rule
            Name               Dur  Dir byte-cnt Auto rate  penalty Reg Enf Ena Persist
                              (min)     (kbytes) enf  (min) (min)   QoS QoS
 residential                   10   us  5        act  1     10080   5   10  Yes Yes
 ef-q11d                       30   ds  150      act  1     20      11  99  Yes Yes
 ef-q11u                       30   us  60       act  1     20      11  99  Yes Yes
 ef-q21                        720  us  60       act  1     10      21  81  Yes Yes
 ef-q21d                       300  ds  150      act  1     10      21  81  Yes Yes
 ef-q22                        720  us  60       act  1     10      22  82  Yes Yes
 ef-q22d                       300  ds  150      act  1     10      22  82  Yes No
 ef-q23                        720  us  60       act  1     10      23  83  Yes Yes
 ef-q23d                       300  ds  150      act  1     10      23  83  Yes Yes
 ef-q24                        720  us  60       act  1     10      24  84  Yes Yes
 ef-q24d                       300  ds  150      act  1     10      24  84  Yes Yes
 ef-q25                        720  us  60       act  1     10      25  85  Yes Yes
 ef-q25d                       300  ds  150      act  1     10      25  85  Yes Yes
 ef-q26                        720  us  60       act  1     10      26  86  Yes Yes
 ef-q26d                       300  ds  150      act  1     10      26  86  Yes Yes
 ef-q27                        720  us  60       act  1     10      27  87  Yes Yes
 ef-q27d                       300  ds  150      act  1     10      27  87  Yes Yes
 ef-q28                        720  us  60       act  1     10      28  88  Yes Yes
```

```
 ef-q28d                      300  ds  150      act  1     10      28  88  Yes No
 ef-q5d                       300  ds  150      act  1     10      5   99  Yes Yes
 ef-q5u                       720  us  600      act  1     10      5   99  Yes Yes
```

The following example shows sample output from the **show cable qos enforce-rule** command for a particular enforce-rule named “test”:

```
Router# show cable qos enforce-rule test
     Name        Type Dur  Dir Avg-rate Auto rate     Reg        Enf      En Per
                      (min)     kbits/s  enf  (min)
test            p-off 120   us 1        act  10     255         4         Y  Y
```

The following example shows the sample output from the **show cable qos enforce-rule verbose** command for an enforce-rule named “test”:

```
Router# show cable qos enforce-rule test verbose
Name                          : test
Version                       : docsis11
Monitoring Type               : peak-offpeak
Registered                    : REG-DS
Enforced                      : ENF-DS
Monitoring Duration           : 70 (in minutes)
Sample-rate                   : 10 (in minutes)
Average-rate                  : 3 kbits/sec
Direction                     : downstream
Auto Enforce                  : Yes
Current Penalty Duration      : 10 (in minutes)
Default Penalty Duration      : 10 (in minutes)
Penalty End-time              : 23:0 (time of day)
Rule Enabled                  : Yes
Persistence                   : Yes
Weekend                       : No
Penalty Off                   : No
Monitor Weekend               : Yes
Monitoring after RelTime      : Off
First Peak Time               : 10:0
Duration                      : 60 (in minutes)
First Average-rate            : 1 kbits/sec
Second Peak Time              : 19:0
Duration                      : 65 (in minutes)
Second Average-rate           : 2 kbits/sec
Offpeak Duration              : 70 (in minutes)
Offpeak Average-rate          : 3 kbits/sec
Auto Enforce                  : Yes
Sample Rate                   : 10
Penalty-Period for week-days  : 0
Weekend First Peak Time       : 11:0
Weekend Duration              : 75 (in minutes)
Weekend First Average-rate    : 4 kbits/sec
Weekend Second Peak Time      : 20:0
Weekend Duration              : 80 (in minutes)
Weekend Second Average-rate   : 5 kbits/sec
Weekend Offpeak Duration      : 85 (in minutes)
Weekend Offpeak Average-rate  : 6 kbits/sec
Weekend Auto Enforce          : Yes
Weekend Sample Rate           : 12
Penalty-Period for week-ends  : 0
router#sh clock
*17:30:50.259 UTC Mon Apr 19 2010
```

The following example shows sample output from the **show cable qos enforce-rule verbose** command for a particular enforce-rule named “test” that has specified peak-offpeak weekend monitoring options:

```
Router# show cable qos enforce-rule test verbose
Name : test
Version : docsis10
Monitoring Type : peak-offpeak
Registered : 255
Enforced : 4
Monitoring Duration : 120 (in minutes)
Sample-rate : 10 (in minutes)
Average-rate : 1 kbits/sec
Direction : upstream
Penalty Time : 10080 (in minutes)
Penalty End-time : 23 (time of day in hrs)
Rule Enabled : Yes
Persistence : Yes
Week-end : Yes
First Peak Time : 6
Duration : 180 (in minutes)
First Average-rate : 2 kbits/sec
Second Peak Time : 18
Duration : 240 (in minutes)
Second Averate-rate : 3 kbits/sec
Offpeak Duration : 120 (in minutes)
Offpeak Average-rate : 1 kbits/sec
Auto-enforce : active
Weekend First Peak Time : 8
Weekend First Duration : 120 (in minutes)
Weekend First Average-rate : 2 kbits/sec
Weekend Second Peak Time : 18
Weekend Second Duration : 180 (in minutes)
Weekend Second Average-rate : 5 kbits/sec
Weekend Offpeak Duration : 240 (in minutes)
Weekend Offpeak Average-rate : 4 kbits/sec
Weekend Auto-enforce : active
```

Displaying the Current Subscriber Usage

To display the usage for all subscribers on a cable interface, use the show cable subscriber-usage command in privileged EXEC mode without any options.

To display the usage for just those subscribers who are violating their registered quality of service (QoS) profiles, use the show cable subscriber-usage over-consume form of the command.

The following example shows sample output from the **show cable subscriber-usage** command for all users on the specified cable interface:

```
Router# show cable subscriber-usage cable 6/0/1
Sfid Mac Address   Enforce-rule Total-Kbyte    Last-detect    Last-penalty   Pen
                      Name        Count         time           time        Flag
3    0007.0e03.110d efrule-q5   121944817  Jan1  03:44:08  Jan1  03:54:08  Act
4    0007.0e03.110d efrule-q5d  1879076068 Jan1  03:35:05  Jan1  03:45:06  Act
5    0007.0e03.1431 efrule-q5   120052387  Jan1  03:44:18  Jan1  03:54:18  Act
6    0007.0e03.1431 efrule-q5d  1838493626 Jan1  03:34:55  Jan1  03:44:55  Act
7    0007.0e03.1445 efrule-q5   120919427  Jan1  03:44:08  Jan1  03:54:08  Act
8    0007.0e03.1445 efrule-q5d  1865955172 Jan1  03:35:06  Jan1  03:45:06  Act
9    0007.0e03.1225 efrule-q5   120200155  Jan1  03:44:18  Jan1  03:54:18  Act
10   0007.0e03.1225 efrule-q5d  1839681070 Jan1  03:34:55  Jan1  03:44:55   -
11   0007.0e03.0cb1 efrule-q5   122941643  Jan1  03:43:58  Jan1  03:53:58  Act
12   0007.0e03.0cb1 efrule-q5d  1889107176 Jan1  03:35:06  Jan1  03:45:06  Act
13   0007.0e03.1435 efrule-q5   119504795  Jan1  03:44:18  Jan1  03:54:18  Act
14   0007.0e03.1435 efrule-q5d  1835164034 Jan1  03:34:55  Jan1  03:44:55   -
```

By default, the display is sorted by the service flow ID (SFID). To sort the display by the subscriber byte count, with the largest byte counts listed first, use the **sort-byte-count** option. The following example shows sample output for **show cable subscriber-usage sort-byte-count** form of the command:

Note The **sort-byte-count** option was replaced by the **sort-avg-rate** option.

```
Router# show cable subscriber-usage
 sort-byte-count

Sfid Mac Address   Enforce-rule Total-Kbyte  Last-detect    Last-penalty   Pen
                       Name        Count       time             time        Flag
7    0007.0e03.2cad test1       65157114   Feb24 11:36:34  Mar3  11:36:34  Act
9    0007.0e03.2c45 test1       16381014                                   -
5    0007.0e03.2c25 test1       13440960                                   -
```

Configuration Examples for Subscriber Traffic Management on the Cisco CMTS Routers

This section lists sample configurations for the Subscriber Traffic Management feature on a CMTS router:

Example: DOCSIS Configuration File and STM Service Classes

The following example shows a sample DOCSIS configuration file along with sample registered and enforced QoS service classes that you could define on a Cisco CMTS router to perform subscriber traffic management.

DOCSIS Configuration File Options

This is an example of a very basic set of options that you can configure for a cable modem in your DOCSIS configuration file that supports a successful configuration of new QoS service class options on the Cisco CMTS router.

Note There are certain QoS parameters that cannot be changed from the registered QoS parameter set and a new service class. For example, the **max-burst** value must match the originally registered in the DOCSIS configuration file, and the registered and enforced QoS service classes on the Cisco CMTS router. If the **max-burst** value differs from the registered CMTS service class and the DOCSIS configuration file, the CM might go into reject-c state, or the enforced class could fail.

The following example shows the configuration of two service classes named “BE-STM-US-1” and “BE-STM-DS-1” in a DOCSIS configuration file to define a basic set of upstream and downstream parameters:

```
03 (Net Access Control) = Yes
17 (Baseline Privacy Block)
S01 (Authorize Wait Timeout) = 10
18 (Maximum Number of CPE) = 10
24 (Upstream Service Flow Block)
S01 (Flow Reference) = 1
```

```
S04 (Service Class Name) = BE-STM-US-1
S06 (QoS Parameter Set Type) = 7
25 (Downstream Service Flow Block)
S01 (Flow Reference) = 2
S04 (Service Class Name) = BE-STM-DS-1
S06 (QoS Parameter Set Type) = 7
29 (Privacy Enable) = Yes
```

The following example shows sample **cable service class** commands on the Cisco CMTS router for configuration of subscriber traffic management that correspond to the service class names in the DOCSIS configuration file of "BE-STM-US-1" and "BE-STM-DS-1." These service classes correspond to the registered service classes configured by the **service-class registered** command for the QoS enforce-rules shown later in this example:

```
cable service class 2 name BE-STM-US-1
cable service class 2 upstream
cable service class 2 max-rate 2000000
cable service class 2 max-burst 3044
cable service class 2 max-concat-burst 8000
cable service class 3 name BE-STM-DS-1
cable service class 3 downstream
cable service class 3 max-rate 30000000
cable service class 3 max-concat-burst 8000
```

For the cable modem to achieve maximum US thoroughput, provide a large value to the max-concat-burst keyword in the cable service class command.

The following example shows sample **cable service class** commands on the Cisco CMTS router that configure new QoS parameters for identified subscribers to limit bandwidth using the **max-rate** parameter. These service classes correspond to the enforced service classes configured by the **service-class enforced** command for the QoS enforce rules shown later in this example:

```
cable service class 102 name BEUS-1
cable service class 102 upstream
cable service class 102 max-rate 48888
cable service class 102 max-burst 3044
cable service class 102 max-concat-burst 8000
cable service class 103 name BEDS-1
cable service class 103 downstream
cable service class 103 max-rate 988888
cable service class 103 max-concat-burst 8000
```

The following example shows configuration of the corresponding enforce-rules for upstream and downstream monitoring, which identifies the registered and enforced service classes:

```
cable qos enforce-rule US-1
  monitoring-basics legacy docsis11
  penalty-period 10
  service-class registered BE-STM-US-1
  service-class enforced BEUS-1
  duration 10 avg-rate 1 sample-interval 10 up enf
  enabled
!
cable qos enforce-rule DS-1
  monitoring-basics legacy docsis11
  penalty-period 10
  service-class registered BE-STM-DS-1
  service-class enforced BEDS-1
  duration 10 avg-rate 1 sample-interval 10 do enf
  enabled
```

Example: Downstream Configuration

The following example shows a typical enforce-rule configuration for traffic in the downstream direction:

```
!
cable qos enforce-rule downstream-rule
  monitoring-basics legacy docsis11
  penalty-period 10
  service-class registered class5
  service-class enforced class99
  duration 30 avg-rate 1 sample-interval 10 downstream enforce
  enabled
```

Example: Upstream Configuration

The following example shows a typical enforce-rule configuration for traffic in the upstream direction:

```
!
cable qos enforce-rule upstream-rule
  monitoring-basics legacy docsis11
  penalty-period 10
  service-class registered class5
  service-class enforced class99
  duration 30 avg-rate 1 sample-interval 10 upstream enforce
  enabled
```

Example: Downstream and Upstream Configuration

The following example shows a typical enforce-rule configuration for traffic in both the downstream and upstream directions. Two separate rules are created, using the identical configuration, except for the keywords **upstream** and **downstream** in the **duration** command.

Note The enforce rules for the upstream and downstream directions can use either an identical configuration, or they can use their own individual configurations.

```
!
cable qos enforce-rule upstream-rule
  monitoring-basics legacy docsis11
  penalty-period 10
  service-class registered class5
  service-class enforced class99
  duration 30 avg-rate 5 sample-interval 10 upstream enforce
  enabled
cable qos enforce-rule downstream-rule
  monitoring-basics legacy docsis11
  penalty-period 10
  service-class registered class5
  service-class enforced class99
  duration 30 avg-rate 5 sample-interval 10 downstream enforce
  enabled
```

The following example shows an enforce-rule configuration for traffic in upstream direction. A unique penalty duration is configured for upstream, with monitoring turned on after the penalty release time.

Note For upstream direction, a unique penalty duration (120 minutes) is configured, which takes precedence over the duration configured using the penalty-period command (60 minutes). A fresh monitoring starts after the penalty release time (23:00), when all the traffic counters are reset to 0.

```
!
cable qos enforce-rule upstream_rule
  monitoring-basics peak-offpeak docsis10
  penalty-period 60 time-of-day 23:00 monitoring-on
  qos-profile registered 6
  qos-profile enforced 100
  peak-time1 10:30 duration 120 avg-rate 10 peak-time2 22:10 duration 60 avg-rate 10
sample-interval 10 penalty 120 upstream enforce
enabled
```

Example: Weekend Monitoring Configuration

The following example shows a sample configuration of peak-offpeak weekend monitoring for DOCSIS 1.0 cable modems:

```
cable qos enforce-rule monitoring
    monitoring-basics peak-offpeak docsis10
    penalty-period 60
    qos-profile registered 6
    qos-profile enforced 100
    peak-time1 10 duration 120 avg-rate 10 peak-time2 23 duration 60 avg-rate 10
sample-interval 10 upstream enforce
    weekend peak-time1 8 duration 60 avg-rate 100 peak-time2 20 duration 60 avg-rate 10000
 duration 90 avg-rate 20000 sample-interval 20 downstream enforce
    enabled
```

Additional References

For additional information related to the Subscriber Traffic Management feature, refer to the following references:

Related Documents

Related Topic	Document Title
Cable commands	Cisco IOS CMTS Cable Command Reference

Standards

Standards[6]	Title
SP-RFIv1.1-I09-020830	*Data-over-Cable Service Interface Specifications Radio Frequency Interface Specification, version 1.1* (http://www.cablemodem.com)

Standards[6]	Title
draft-ietf-ipcdn-docs-rfmibv2-06	*Radio Frequency (RF) Interface Management Information Base for DOCSIS 2.0 Compliant RF Interfaces*

[6] Not all supported standards are listed.

MIBs

MIBs[7]	MIBs Link
• CISCO-CABLE-QOS-MONITOR-MIB • DOSCIS-QOS-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

[7] Not all supported MIBs are listed.

RFCs

RFCs[8]	Title
RFC 2233	DOCSIS OSSI Objects Support
RFC 2665	DOCSIS Ethernet MIB Objects Support
RFC 2669	Cable Device MIB

[8] Not all supported RFCs are listed.

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Subscriber Traffic Management

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

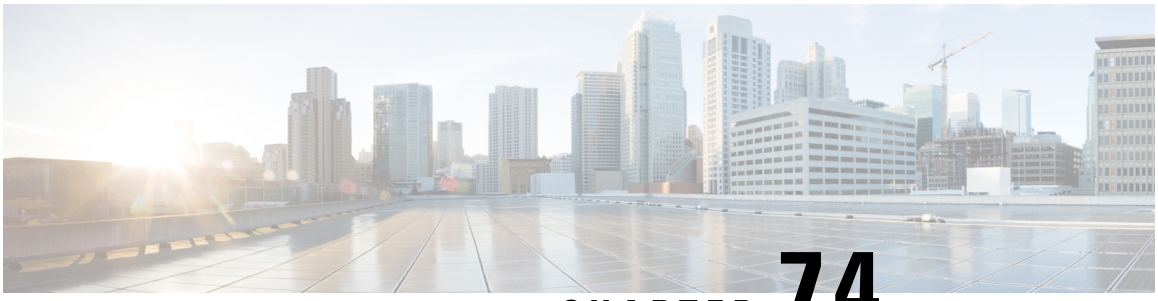
Table 186: Feature Information for Subscriber Traffic Management

Feature Name	Releases	Feature Information
Subscriber Traffic Management	Cisco IOS-XE Release 3.17.0S	This feature was introduced on the Cisco cBR Series Converged Broadband Router s.

PART XI

Security and Cable Monitoring Configuration

- Dynamic Shared Secret, on page 1137
- Lawful Intercept Architecture, on page 1163
- Cable Monitoring Feature for Cisco cBR Series Routers , on page 1177
- Source-Based Rate Limit, on page 1185
- Cable Duplicate MAC Address Reject, on page 1203
- Cable ARP Filtering, on page 1215
- Subscriber Management Packet Filtering Extension for DOCSIS 2.0, on page 1231
- MAC Filtering, on page 1239
- IPv6 Segment Routing on Cisco cBR, on page 1245



CHAPTER 74

Dynamic Shared Secret

This document describes the Dynamic Shared Secret feature, which enables service providers to provide higher levels of security for their Data-over-Cable Service Interface Specifications (DOCSIS) cable networks. This feature uses randomized, single-use shared secrets to verify the DOCSIS configuration files that are downloaded to each cable modem.

The Dynamic Shared Secret feature automatically creates a unique DOCSIS shared secret on a per-modem basis, creating a one-time use DOCSIS configuration file that is valid only for the current session. This ensures that a DOCSIS configuration file that has been downloaded for one cable modem can never be used by any other modem, nor can the same modem reuse this configuration file at a later time.

This patented feature is designed to guarantee that all registered modems use only the quality of service (QoS) parameters that have been specified by the DOCSIS provisioning system for a particular modem at the time of its registration. This feature is an accepted DOCSIS standard.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1138
- Prerequisites for Dynamic Shared Secret, on page 1138
- Restrictions for Dynamic Shared Secret, on page 1139
- Information About Dynamic Shared Secret, on page 1142
- How to Configure the Dynamic Shared Secret Feature, on page 1148
- How to Monitor the Dynamic Shared Secret Feature, on page 1154
- Troubleshooting Cable Modems with Dynamic Shared Secret, on page 1158
- Configuration Examples for Dynamic Shared Secret, on page 1158
- Additional References, on page 1160
- Feature Information for Dynamic Shared Secret, on page 1161

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 187: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Dynamic Shared Secret

The configuration of Dynamic Shared Secret feature is supported on the Cisco CMTS routers.

Following is a list of other important prerequisites for the Dynamic Shared Secret feature:

- The Cisco CMTS must be running Cisco IOS-XE 3.15.0S or later.
- The Dynamic Shared Secret feature supports an external provisioning server.
- A cable modem must be able to register with the Cisco CMTS before enabling the Dynamic Shared Secret feature.

- For full security, DOCSIS configuration files should have filenames that are at least 5 or more characters in length.
- For best performance during the provisioning of cable modems, we recommend using Cisco Network Registrar Release 3.5 or later.

Note When the Dynamic Shared Secret feature is enabled using its default configuration, a cable modem diagnostic webpage shows a scrambled name for its DOCSIS configuration file. This filename changes randomly each time that the cable modem registers with the CMTS. To change the default behavior, use the **nocrypt** option with the **cable dynamic-secret** command.

Restrictions for Dynamic Shared Secret

General Restrictions for Dynamic Shared Secret

- Shared-secret and secondary-shared-secret cannot be configured with Dynamic Shared Secret feature.
- If you configure the Dynamic Shared Secret feature on a primary cable interface, you should also configure the feature on all of the corresponding subordinate cable interfaces.
- The Dynamic Shared Secret feature ensures that each cable modem registering with the CMTS can use only the DOCSIS configuration file that is specified by the service provider's authorized Dynamic Host Configuration Protocol (DHCP) and TFTP servers, using the DOCSIS-specified procedures.
- The Dynamic Shared Secret feature does not affect cable modems that are already online and provisioned. If a cable modem is online, you must reset it, so that it reregisters, before it complies with the Dynamic Shared Secret feature.
- The DMIC lock mode uses the following behavior during a switchover event in HCCP N+1 Redundancy. All cable modems which were previously in lock mode are taken offline during a switchover event, and the prior state of locked modems is lost. If previously locked modems remain non-compliant, they will return to LOCK mode after three failed registration attempts. If the modems have become DOCSIS compliant, they will return online in the normal fashion. Refer to the SNMP Support, on page 1145 for additional information about DMIC lock mode.
- If a Broadband Access Center for Cable (BACC) provisioning server is being used, the Device Provisioning Engine (DPE) TFTP server verifies that the IP address of the TFTP client matches the expected DOCSIS cable modem IP Address. If a match is not found, the request is dropped. This functionality is incompatible with the CMTS DMIC feature. Use the no tftp verify-ip command on all BACC DPE servers to disable the verification of the requestor IP address on dynamic configuration TFTP requests. Refer to the Cisco Broadband Access Centre DPE CLI Reference in the http://www.cisco.com/c/en/us/td/docs/net_mgmt/broadband_access_center_for_cable/4-0/command/reference/DPECLIRef40.html for additional information.

Cable Modem Restrictions for Dynamic Shared Secret

DHCP Restriction for Incognito Server and Thomson Cable Modems

The Dynamic Host Configuration Protocol (DHCP) passes configuration information to DHCP hosts on a TCP/IP network. Configuration parameters and other control information are stored in the options field of the DHCP message.

When using DMIC with the Incognito DHCP server, the Incognito server must be re-configured so that the following two options are *not* sent in the DHCP message:

- *option 66* —This option is used to identify a TFTP server when the sname field in the DHCP header has been used for DHCP options. Option 66 is a variable-length field in the Options field of a DHCP message described as "an option used to identify a TFTP server when the 'sname' field in the DHCP header has been used for DHCP options" as per RFC 2132.
- *sname field* —The sname field is a 64-octet field in the header of a DHCP message described as "optional server host name, null terminated string," as per RFC2131. A DHCP server inserts this option if the returned parameters exceed the usual space allotted for options. If this option is present, the client interprets the specified additional fields after it concludes interpretation of the standard option fields.

Note It is not compliant with DOCSIS to include both of these options in the DHCP message.

The problematic packet capture below is a DHCP offer in which both sname and option 66 are set (in this respective sequence):

```
0000  00 30 19 47 8f 00 00 d0 b7 aa 95 50 08 00 45 00
0010  01 4a 8f 50 00 00 80 11 46 30 ac 10 02 01 ac 10
0020  0a 01 00 43 00 43 01 36 0c 75 02 01 06 00 b0 a0
0030  25 01 00 00 00 00 00 00 00 00 ac 10 0a 53 00 00
0040  00 00 ac 10 0a 01 00 10 95 25 a0 b0 00 00 00 00
0050  00 00 00 00 00 00 5b 31 37 32 2e 31 36 2e 32 2e
(sname option immediately above)
0060  31 5d 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0090  00 00 00 00 00 00 64 65 66 61 75 6c 74 2e 63 66
00a0  67 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00b0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00c0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00d0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00e0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00f0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0110  00 00 00 00 00 00 63 82 53 63 35 01 02 36 04 ac
0120  10 02 01 33 04 00 06 94 0d 01 04 ff ff ff 00 02
0130  04 ff ff b9 b0 03 08 ac 10 02 fe ac 10 0a 01 04
0140  04 ac 10 02 01 07 04 ac 10 02 01 42 0a 31 37 32
(option 66 immediately above)
0150  2e 31 36 2e 32 2e 31 ff
```

When using DMIC with Incognito DHCP servers and Thomson cable modems, you must prevent both options from being sent in the DHCP offer. Use one of the following workaround methods to achieve this:

- Change the Incognito DHCP server so that it does not include the sname option as described above.

- Change the cable modem code so that sname is not prioritized above option 66, as in the problematic packet capture shown in the example above.
- Migrate to a compliant DHCP and TFTP server such as CNR. This also offers significantly higher performance.

Refer to these resources for additional DOCSIS DHCP information, or optional DHCP MAC exclusion:

- *DHCP Options and BOOTP Vendor Extensions, RFC 2132*

http://www.ietf.org/rfc/rfc2132.txt

- *Filtering Cable DHCP Lease Queries on Cisco CMTS Routers*

http://www.cisco.com/en/US/docs/cable/cmts/feature/cblsrcvy.html

DOCSIS Compliance

- Cable modems are assumed to be DOCSIS-compliant. If a cable modem is not fully DOCSIS-compliant, it could trigger a CMTS Message Integrity Check (MIC) failure during registration in rare circumstances. Under normal operations, however, it can be assumed that cable modems that fail the CMTS MIC check from the Dynamic Shared Secret feature are either not DOCSIS-compliant, or they might have been hacked by the end user to circumvent DOCSIS security features.

Some of the cable modems with the following OUIs have been identified as having problems with the Dynamic Shared Secret feature, depending on the hardware and software revisions:

-
  - 00.01.03
  - 00.E0.6F
  - 00.02.B2

These particular cable modems can remain stuck in the init(o) MAC state and cannot come online until the Dynamic Shared Secret feature is disabled. If this problem occurs, Cisco recommends upgrading the cable modem's software to a fully compliant software revision.

Alternatively, these cable modems may be excluded from the *dynamic* secret function using the following command in global configuration mode:

cable dynamic-secret exclude

Excluding cable modems means that if a violator chooses to modify their cable modem to use one of the excluded OUIs, then the system is no longer protected. Refer to the #unique_1580.

Tip To help providers to identify non-DOCSIS compliant modems in their network, the Dynamic Shared Secret feature supports a "mark-only" option. When operating in the mark-only mode, cable modems might be able to successfully obtain higher classes of service than are provisioned, but these cable modems will be marked as miscreant in the **show cable modem** displays (with **!online**, for example). Such cable modems also display with the **show cable modem rogue** command. Service providers may decide whether those cable modems must be upgraded to DOCSIS-compliant software, or whether the end users have hacked the cable modems for a theft-of-service attack.

The following example illustrates output from a Cisco CMTS that is configured with the **cable dynamic-secret mark** command with miscreant cable modems installed. These cable modems may briefly show up as "reject(m)" for up to three registration cycles before achieving the **!online** status.

```
Router# show cable modem rogue
                                    Spoof TFTP
MAC Address    Vendor     Interface  Count Dnld Dynamic Secret
000f.0000.0133 00.0F.00   C4/0/U1    3     Yes  905B740F906B48870B3A9C5E441CDC67
000f.0000.0130 00.0F.00   C4/0/U1    3     Yes  051AEA93062A984F55B7AAC979D10901
000f.0000.0132 00.0F.00   C4/0/U2    3     Yes  FEDC1A6DA5C92B17B23AFD2BBFBAD9E1
vxr#scm | inc 000f
000f.0000.0133 4.174.4.101     C4/0/U1  !online       1     -7.00 2816    0   N
000f.0000.0130 4.174.4.89      C4/0/U1  !online       2     -6.50 2819    0   N
000f.0000.0132 4.174.4.90      C4/0/U2  !online       18    -7.00 2819    0   N
```

TFTP Restrictions

- Cable modems can become stuck in the TFTP transfer state (this is indicated as init(o) by the **show cable modem** command) in the following situation:
  - The Dynamic Shared Secret feature is enabled on the cable interface, using the **cable dynamic-secret** command. This feature applies if the cable modem is a miscreant cable modem, or if the cable modem is a DOCSIS 1.0 cable modem running early DOCSIS 1.0 firmware that has not yet been updated. This feature also applies if the TFTP server is unable to provide the cable modem's TFTP configuration file to the Cisco CMTS. This is the case, for example, when using BACC and not configuring the system to permit a TFTP request from a non-matching source IP address. The **debug cable dynamic-secret** command also shows this failure.
  - A large number of cable modems are registering at the same time. Some or all of those cable modems could also be downloading the DOCSIS configuration file using multiple TFTP transfers that use multiple TFTP ports on the Cisco CMTS router, and the TFTP server is unable to keep up with the rate of TFTP requests generated by the system. Some TFTP servers may be limited to the number of concurrent TFTP get requests initiated by the same source IP address per unit time, or simply unable to handle the rate of new modem registrations before cable dynamic-secret is configured. The **debug cable dynamic-secret** command shows failure to receive some files in this situation.

This situation of stuck cable modems can result in the TFTP server running out of available ports, resulting in the cable modems failing the TFTP download stage. To prevent this situation from happening, temporarily disable the Dynamic Shared Secret feature on the cable interface or reduce the size of the DOCSIS configuration file.

Information About Dynamic Shared Secret

The DOCSIS specifications require that cable modems download, from an authorized TFTP server, a DOCSIS configuration file that specifies the quality of service (QoS) and other parameters for the network session. Theft-of-service attempts frequently attempt to intercept, modify, or substitute the authorized DOCSIS configuration file, or to download the file from a local TFTP server.

To prevent theft-of-service attempts, the DOCSIS specification allows service providers to use a shared secret password to calculate the CMTS Message Integrity Check (MIC) field that is attached to all DOCSIS configuration files. The CMTS MIC is an MD5 digest that is calculated over the DOCSIS Type/Length/Value (TLV) fields that are specified in the configuration file, and if a shared secret is being used, it is used in the MD5 calculation as well.

The cable modem must include its calculation of the CMTS MIC in its registration request, along with the contents of the DOCSIS configuration file. If a user modifies any of the fields in the DOCSIS configuration file, or uses a different shared secret value, the CMTS cannot verify the CMTS MIC when the cable modem

registers. The CMTS does not allow the cable modem to register, and marks it as being in the "reject(m)" state to indicate a CMTS MIC failure.

Users, however, have used various techniques to circumvent these security checks, so that they can obtain configuration files that provide premium services, and then to use those files to provide themselves with higher classes of services. Service providers have responded by changing the shared secret, implementing DOCSIS time stamps, and using modem-specific configuration files, but this has meant creating DOCSIS configuration files for every cable modem on the network. Plus, these responses would have to be repeated whenever a shared secret has been discovered.

The Dynamic Shared Secret feature prevents these types of attacks by implementing a dynamically generated shared secret that is unique for each cable modem on the network. In addition, the dynamic shared secrets are valid only for the current session and cannot be reused, which removes the threat of "replay attacks," as well as the reuse of modified and substituted DOCSIS configuration files.

Modes of Operation

The Dynamic Shared Secret feature can operate in three different modes, depending on what action should be taken for cable modems that fail the CMTS MIC verification check:

- Marking Mode—When using the **mark** option, the CMTS allows cable modems to come online even if they fail the CMTS MIC validity check. However, the CMTS also prints a warning message on the console and marks the cable modem in the **show cable modem** command with an exclamation point (!), so that this situation can be investigated.
- Locking Mode—When the **lock** option is used, the CMTS assigns a restrictive QoS configuration to CMs that fail the MIC validity check twice in a row. You can specify a particular QoS profile to be used for locked cable modems, or the CMTS defaults to special QoS profile that limits the downstream and upstream service flows to a maximum rate of 10 kbps.

If a customer resets their CM, the CM will reregister but still uses the restricted QoS profile. A locked CM continues with the restricted QoS profile until it goes offline and remains offline for at least 24 hours, at which point it is allowed to reregister with a valid DOCSIS configuration file. A system operator can manually clear the lock on a CM by using the **clear cable modem lock** command.

This option frustrates users who are repeatedly registering with the CMTS in an attempt to guess the shared secret, or to determine the details of the Dynamic Shared Secret security system.

- Reject Mode—In the reject mode, the CMTS refuses to allow CMs to come online if they fail the CMTS MIC validity check. These cable modems are identified in the **show cable modem** displays with a MAC state of "reject(m)" (bad MIC value). After a short timeout period, the CM attempts to reregister with the CMTS. The CM must register with a valid DOCSIS configuration file before being allowed to come online. When it does come online, the CMTS also prints a warning message on the console and marks the cable modem in the **show cable modem** command with an exclamation point (!), so that this situation can be investigated.

Note To account for possible network problems, such as loss of packets and congestion, the Cisco CMTS will allow a cable modem to attempt to register twice before marking it as having failed the Dynamic Shared Secret authentication checks.

Operation of the Dynamic Shared Secret

The Dynamic Shared Secret feature automatically creates a unique DOCSIS shared secret on a per-modem basis, creating a one-time use DOCSIS configuration file that is valid only for the current session. This ensures that a DOCSIS configuration file that has been downloaded for one cable modem can never be used by any other modem, nor can the same modem reuse this configuration file at a later time.

This patent pending feature is designed to guarantee that all registered modems are using only the QOS parameters that have been specified by the DOCSIS provisioning system for that particular modem at the time of its registration.

When a DOCSIS-compliant cable modem registers with the CMTS, it sends a DHCP request, and the DHCP server sends a DHCP response that contains the name of the DOCSIS configuration file that the cable modem should download from the specified TFTP server. The cable modem downloads the DOCSIS configuration file and uses its parameters to register with the CMTS

When the Dynamic Shared Secret feature is enabled, the CMTS performs the following when it receives the DHCP messages:

- The CMTS creates a dynamically generated shared secret.
- In the default configuration, the CMTS takes the name of the DOCSIS configuration file and generates a new, randomized filename. This randomized filename changes every time the cable modem registers, which prevents the caching of DOCSIS configuration files by cable modems that are only semi-compliant with the DOCSIS specifications. You can disable this randomization of the filename by using the **nocrypt** option with the **cable dynamic-secret** command.
- The CMTS changes the IP address of the TFTP server that the cable modem should use to the IP address of the CMTS. This informs the cable modem that it should download its configuration file from the CMTS.
- The CMTS downloads the original DOCSIS configuration file from the originally specified TFTP server so that it can modify the file to use the newly generated dynamic secret.

When the cable modem downloads the DOCSIS configuration file, it receives the modified file from the CMTS. Because this file uses the one-time-use dynamically generated shared secret, the CMTS can verify that the cable modem is using this configuration file when it attempts to register with the CMTS.

Note The Dynamic Shared Secret feature does not support and is incompatible with, the use of the original shared secret or secondary shared secrets that are configured using the **cable shared-secondary-secret** and **cable shared-secret** commands.

Tip Although a user could attempt to circumvent these checks by downloading a DOCSIS configuration file from a local TFTP server, the cable modem would still fail the CMTS MIC verification.

Interaction with Different Commands

The Dynamic Shared Secret feature works together with a number of other commands to ensure network security and integrity:

- **cable shared-secret**—The DOCSIS specification allows service providers to use a shared-secret to ensure that cable modems are using only authorized DOCSIS configuration files.

The Dynamic Shared Secret feature is incompatible with **cable shared-secret**. Do not configure the **cable shared-secret** command when using the Dynamic Shared Secret feature

- **cable shared-secondary-secret**— The Dynamic Shared Secret feature is incompatible with **cable shared-secret**. Do not configure the **cable secondary-shared-secret** command when using the Dynamic Shared Secret feature

Performance Information

The Dynamic Shared Secret feature does not add any additional steps to the cable modem registration process, nor does it add any additional requirements to the current provisioning systems. This feature can have either a small negative or a small positive effect on the performance of the network provisioning system, depending on the following factors:

- The provisioning system (DHCP and TFTP servers) being used
- The number of cable modems that are coming online
- The vendor and software versions of the cable modems
- The number and size of the DOCSIS configuration files

Large-scale testing has shown that the Dynamic Shared Secret feature can affect the time it takes for cable modems to come online from 5% slower to 10% faster. The most significant factor in the performance of the provisioning process is the provisioning system itself. For this reason, Cisco recommends using Cisco Network Registrar (CNR) Release 3.5 or greater, which can provide significant performance improvements over generic DHCP and TFTP servers.

The second-most important factor in the performance of cable modem provisioning is the number and size of the DOCSIS configuration files. The size of the configuration file determines how long it takes to transmit the file to the cable modem, while the number of configuration files can impact how efficiently the system keeps the files in its internal cache, allowing it to reuse identical configuration files for multiple modems.

SNMP Support

Cisco IOS-XE 3.15.0S and later releases add the following SNMP support for the Dynamic Shared Secret feature:

- Adds the following MIB objects to the CISCO-DOCS-EXT-MIB:
  - cdxCmtsCmDMICMode—Sets and shows the configuration of the Dynamic Shared Secret feature for a specific cable modem (not configured, mark, lock, or reject).
  - cdxCmtsCmDMICLockQoS—Specifies the restrictive QoS profile assigned to a cable modem that has failed the Dynamic Shared Secret security checks, when the interface has been configured for lock mode.
  - cdxCmtsCmStatusDMICTable—Lists all cable modems that have failed the Dynamic Shared Secret security checks.
- An SNMP trap (cdxCmtsCmDMICLockNotification) can be sent when a cable modem is locked for failing the Dynamic Shared Secret security checks. The trap can be enabled using the **snmp-server enable traps cable dmic-lock** command.

Note The DMIC lock mode is disabled during a switchover event in HCCP N+1 Redundancy.

System Error Messages

The following system error messages provide information about cable modems that have failed the CMTS Message Integrity Check (MIC) when the Dynamic Shared Secret feature is enabled.

Message

```
%CBR-4-CMLOCKED
```

The cable modem's DOCSIS configuration file did not contain a Message Integrity Check (MIC) value that corresponds with the proper Dynamic Shared Secret that was used to encode it. The CMTS has, therefore, assigned a restrictive quality of service (QoS) configuration to this cable modem to limit its access to the network. The CMTS has also locked the cable modem so that it will remain locked in the restricted QoS configuration until it goes offline for at least 24 hours, at which point it is permitted to reregister and obtain normal service (assuming it is DOCSIS-compliant and using a valid DOCSIS configuration file).

This error message appears when the **cable dynamic-secret lock** command has been applied to a cable interface to enable the Dynamic Shared Secret feature for the DOCSIS configuration files on that cable interface. The cable modem has been allowed to register and come online, but with a QoS configuration that is limited to a maximum rate of 10 kbps for both the upstream and downstream flows. Check to ensure that this cable modem is not running old software that caches the previously used configuration file. Also check for a possible theft-of-service attempt by a user attempting to download a modified DOCSIS configuration file from a local TFTP server. The CM cannot reregister with a different QoS profile until it has been offline for 24 hours, without attempting to register, or you have manually cleared the lock using the **clear cable modem lock** command.

Message

```
%CBR-4-CMMARKED
```

The cable modem's DOCSIS configuration file did not contain a Message Integrity Check (MIC) value that corresponds with the proper dynamic shared secret that was used to encode it. The CMTS has allowed this modem to register and come online, but has marked it in the **show cable modem** displays with an exclamation point (!) so that the situation can be investigated.

This error message appears when the **cable dynamic-secret mark** command has been applied to a cable interface to enable the Dynamic Shared Secret feature for the DOCSIS configuration files on that cable interface. Check to ensure that this cable modem is not running old software that caches the previously used configuration file. Also check for a possible theft-of-service attempt by a user attempting to download a modified DOCSIS configuration file from a local TFTP server.

Message

```
%CBR-4-NOCFGFILE
```

The CMTS could not obtain the DOCSIS configuration file for this cable modem from the TFTP server. This message occurs when the Dynamic Shared Secret feature is enabled on the cable interface with the **cable dynamic-secret** command.

Verify that the CMTS has network connectivity with the TFTP server, and that the specified DOCSIS configuration file is available on the TFTP server. Check that the DHCP server is correctly configured to send the proper configuration filename in its DHCP response to the cable modem. Also verify that the DOCSIS configuration file is correctly formatted.

This problem could also occur if the TFTP server is offline or is overloaded to the point where it cannot respond promptly to new requests. It might also be seen if the interface between the CMTS and TFTP server is not correctly configured and flaps excessively.

Note This error indicates a problem with the provisioning system outside of the Cisco CMTS. Disabling the Dynamic Shared Secret feature does not clear the fault, nor does it allow cable modems to come online. You must first correct the problem with the provisioning system.

Benefits

The Dynamic Shared Secret feature provides the following benefits to cable service providers and their partners and customers:

Improves Network Security

Service providers do not need to worry about users discovering the shared secret value and using it to modify DOCSIS configuration files to give themselves higher levels of service. Even if a user were to discover the value of a dynamically generated shared secret, the user would not be able to use that shared secret again to register.

The generic TFTP server performance and error handling on the Cisco CMTS routers has been greatly improved to support the high performance that is required for rapidly provisioning cable modems.

Flexibility in Dealing with Possible Theft-of-Service Attempts

Service providers have the option of deciding what response to take when a DOCSIS configuration file fails its CMTS MIC check: mark that cable modem and allow the user online, reject the registration request and refuse to allow the user to come online until a valid DOCSIS configuration file is used, or lock the cable modem in a restricted QoS configuration until the modem remains offline for 24 hours. Locking malicious modems is the most effective deterrent against hackers, because it provides the maximum penalty and minimum reward for any user attempting a theft-of-service attack.

No Changes to Provisioning System Are Needed

Service providers can use the Dynamic Shared Secret feature without changing their provisioning or authentication systems. Existing DOCSIS configuration files can be used unchanged, and you do not need to change any existing shared secrets.

Tip If not already done, the service provider could also install access controls that allow only the CMTS routers to download DOCSIS configuration files from the TFTP servers.

No Changes to Cable Modems Are Needed

The Dynamic Shared Secret feature does not require any end-user changes or any changes to the cable modem configuration. This feature supports any DOCSIS compliant cable modem.

Note The Dynamic Shared Secret feature does not affect cable modems that are already online and provisioned. Cable modems that are already online when the feature is enabled or disabled remain online.

Simplifies Network Management

Service providers do not have to continually update the shared secrets on a cable interface whenever the files providing premium services become widely available. Instead, providers can use the same shared secret on a cable interface for significant periods of time, trusting in the Dynamic Shared Secret feature to provide unique, single-use shared secrets for each cable modem.

In addition, service providers do not have to manage unique DOCSIS configuration files for each cable modem. The same configuration file can be used for all users in the same service class, without affecting network security.

Related Features

The following features can be used with the Dynamic Shared Secret feature to enhance the overall security of the cable network.

- Baseline Privacy Interface Plus (BPI+) Authorization and Encryption—Provides a secure link between the cable modem and CMTS, preventing users from intercepting or modifying packets that are transmitted over the cable interface. BPI+ also provides for secure authorization of cable modems, using X.509 digital certificates, as well as a secure software download capability that ensures that software upgrades are not spoofed, intercepted, or altered.

How to Configure the Dynamic Shared Secret Feature

The following sections describe how to enable and configure the Dynamic Shared Secret feature, to disable the feature, to manually clear a lock on a cable modem, or dynamically upgrade firmware on the cable modems.

Note All procedures begin and end at the privileged EXEC prompt ("Router#").

Enabling and Configuring the Dynamic Shared Secret Feature

This section describes how to enable and configure the Dynamic Shared Secret feature on a cable interface.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
	Example: `Router(config)#`	

Step 2	**cable qos permission create**
	Example:
	`Router(config)# cable qos permission create`
	Example:
	`Router(config)#`
		(Optional) If you are using the **lock** option in Step 6, and if you are not specifying a specific QoS profile to be used, you must allow cable modems to create their own QoS profiles.

Step 3	**cable qos permission update**
	Example:
	`Router(config)# cable qos permission update`
	Example:
	`Router(config)#`
		(Optional) If you are using the **lock** option in Step 6, and if you are not specifying a specific QoS profile to be used, you must allow cable modems to update their own QoS profiles.

Step 4	**snmp-server enable traps cable dmic-lock**
	Example:
	`Router(config)# snmp-server enable traps cable dmic-lock`
	Example:
	`Router(config)#`
		(Optional) Enables the sending of SNMP traps when a cable modem fails a dynamic shared-secret security check.

Step 5	**interface cable** *interface*
	Example:
	`Router(config)# interface cable 3/0`
	Example:
	`Router(config-if)#`
		Enters interface configuration mode for the specified cable interface.

Step 6	**cable dynamic-secret** {**lock** [*lock-qos*] | **mark** | **reject**} [**nocrypt** **Example:** `Router(config-if)# cable dynamic-secret lock` **Example:**	Enables the Dynamic Shared Secret feature on the cable interface and configures it for the appropriate option: • **nocrypt**—(Optional) The Cisco CMTS does not encrypt the filenames of DOCSIS configuration files, but sends the files to CMs using their original names.

	Command or Action	Purpose

	`Router(config-if)# cable dynamic-secret lock 90`
	Example:
	`Router(config-if)# cable dynamic-secret mark`
	Example:
	`Router(config-if)# cable dynamic-secret reject`
	Example:
	`Router(config-if)#`
		• **lock**—Cable modems that fail the MIC verification are allowed online with a restrictive QoS profile. The cable modems must remain offline for 24 hours to be able to reregister with a different QoS profile.
		• *lock-qos* —(Optional) Specifies the QoS profile that should be assigned to locked cable modems. The valid range is 1 to 256, and the profile must have already been created. If not specified, locked cable modems are assigned a QoS profile that limits service flows to 10 kbps (requires Step 2 and Step 3).
		• **mark**—Cable modems that fail the MIC verification are allowed online but are marked in the **show cable modem** displays so that the situation can be investigated.
		• **reject**—Cable modems that fail the MIC verification are not allowed to register.
		Note Repeat Step 5 and Step 6 for each cable interface to be configured.

Step 7	**end**
	Example:
	`Router(config-if)# end`
	Example:
	`Router#`
		Exits interface configuration mode and returns to privileged EXEC mode.

What to do next

Note If you configure the Dynamic Shared Secret feature on any interface in a cable interface bundle, you should configure it on all interfaces in that same bundle.

Disabling the Dynamic Shared Secret on a Cable Interface

This section describes how to disable the Dynamic Shared Secret feature on a cable interface. The cable modem continues to be validated against any shared secret or secondary shared secrets that have been defined on the cable interface.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`Router# configure terminal`
	Example:
	`Router(config)#`
		Enters global configuration mode.

Step 2	**interface cable** *interface*
	Example:
	`Router(config)# interface cable 3/0`
	Example:
	`Router(config-if)#`
		Enters interface configuration mode for the specified cable interface.

Step 3	**no cable dynamic-secret**
	Example:
	`Router(config-if)# no cable dynamic-secret`
	Example:
	`Router(config-if)#`
		Disables the Dynamic Shared Secret feature on the cable interface.
		Note Repeat Step 2 and Step 3 for each cable interface to be configured.

Step 4	**end**
	Example:
	`Router(config-if)# end`
	Example:
	`Router#`
		Exits interface configuration mode and returns to privileged EXEC mode.

Excluding Cable Modems from the Dynamic Shared Secret Feature

This section describes how to exclude one or more cable modems from being processed by the Dynamic Shared Secret feature. The cable modem continues to be validated against any shared secret or secondary shared secrets that have been defined on the cable interface.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 2	**cable dynamic-secret exclude** {**oui** *oui-id* | **modem** *mac-address*}
	Example:
	`Router(config)# cable dynamic-secret exclude oui 00.01.B4`
	`Router(config)# cable dynamic-secret exclude modem 00d0.45ba.b34b`
		Excludes one or more cable modems from being processed by the Dynamic Shared Secret security checks, on the basis of their MAC addresses or OUI values:
		• **modem** *mac-address*—Specifies the hardware (MAC) address of one specific and individual cable modem to be excluded from the Dynamic Shared Secret feature. (You cannot specify a multicast MAC address.)
		• **oui** *oui-id*—Specifies the organization unique identifier (OUI) of a vendor, so that a group of cable modems from this vendor are excluded from the Dynamic Shared Secret feature. The OUI should be specified as three hexadecimal bytes separated by either periods or colons.
		Note Repeat this command for each cable modem MAC address or OUI vendor to be excluded.

Step 3	**exit** **Example:** `Router(config)# exit`	Exits the interface configuration mode and returns to privileged EXEC mode.

Clearing the Lock on One or More Cable Modems

This section describes how to manually clear the lock on one or more cable modems. This forces the cable modems to reinitialize, and the cable modems must reregister with a valid DOCSIS configuration file before being allowed online. If you do not manually clear the lock (using the **clear cable modem lock** command), the cable modem is locked in its current restricted QoS profile and cannot reregister with a different profile until it has been offline for at least 24 hours.

Procedure

	Command or Action	Purpose
Step 1	**clear cable modem** {*mac-addr* | *ip-addr* | **all** | **oui***string* | **reject**} **lock**	Clears the lock for the cable modems, which can be identified as follows:

	Command or Action	Purpose

	Example:
	`Router# clear cable modem 0001.0203.0405 lock`
	Example:
	`Router# clear cable modem all lock`
	Example:
	`Router# clear cable modem oui 00.00.0C lock`
	Example:
	`Router#`
		• *mac-addr* —Specifies the MAC address for one particular cable modem to be cleared.
		• *ip-addr* —Specifies the IP address for one particular cable modem to be cleared.
		• **all**—Clears the locks on all locked cable modems.
		• **oui** *string* —Clears the locks on all cable modems with a vendor ID that matches the specified Organizational Unique Identifier (OUI) string.
		• **reject**—Clears the locks on all cable modems that are currently in the reject state (which would occur if a locked cable modem went offline and attempted to reregister before 24 hours had elapsed).

What to do next

Tip A cable modem can also be unlocked by manually deleting the cable modem from all CMTS internal databases, using the **clear cable modem delete** command.

Upgrading Firmware on the Cable Modems

This section describes how to upgrade firmware on cable modems by dynamically inserting the correct TLV values in the DOCSIS configuration file that is downloaded by the cable modem. The DOCSIS configuration file contains the following TLV values:

- Software Upgrade Filename (TLV 9)—Specifies the filename of the firmware.
- Upgrade IPv4 TFTP Server (TLV21)—Specifies the IPv4 address of the TFTP server from where the modem downloads the DOCSIS configuration file.
- Upgrade IPv6 TFTP Server (TLV58)—Specifies the IPv6 address of the TFTP server from where the modem downloads the DOCSIS configuration file.

Note The TFTP server addresses are inserted only when the software upgrade filename (TLV9) is specified and when the TFTP server address (TLV21/TLV58) is either not specified or set to 0.

Before you begin

The Dynamic Shared Secret feature must be enabled first before you can upgrade the firmware on cable modems. See Enabling and Configuring the Dynamic Shared Secret Feature, on page 1148 for more information.

Note The command to enable or disable the Dynamic Shared Secret feature is available at the MAC domain level. However, the command to upgrade the firmware on cable modems is available at the global level.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`Router# configure terminal`
	Example:
	`Router(config)#`
		Enters the global configuration mode.

Step 2	cable dynamic-secret tftp insert-upgrade-server **Example:** `Router(config)# cable dynamic-secret tftp insert-upgrade-server`	Dynamically inserts the specific IPv4 or IPv6 TLV values in the DOCSIS configuration file to complete firmware upgrade on cable modems.

Step 3	**end**
	Example:
	`Router(config)# end`
	Example:
	`Router#`
		Exits the configuration mode and returns to the privileged EXEC mode.

What to do next

Note If you configure the Dynamic Shared Secret feature on an interface in a cable interface bundle, you should configure it on all the interfaces of that bundle.

How to Monitor the Dynamic Shared Secret Feature

This section describes the following procedures you can use to monitor and display information about the Dynamic Shared Secret feature:

Displaying Marked Cable Modems

When you configure a cable interface with the **cable dynamic-secret mark** command, cable modems that fail the dynamically generated CMTS MIC verification are allowed online, but are marked with an exclamation point (!) in the MAC state column in the **show cable modem** display. The exclamation point is also used to identify cable modems that were initially rejected, using the **cable dynamic-secret reject** command, but then reregistered using a valid DOCSIS configuration file.

For example, the following example shows that four cable modems are marked as having failed the CMTS MIC verification, but that they have been allowed online:

```
Router# show cable modems

MAC Address    IP Address      I/F       MAC         Prim RxPwr  Timing  Num BPI
                                         State       Sid  (db)   Offset  CPE Enb
0010.9507.01db 144.205.151.130 C5/1/0/U5 online(pt)  1     0.25    938   1    N
0080.37b8.e99b 144.205.151.131 C5/1/0/U5 online      2     -0.25   1268  0    N
0002.fdfa.12ef 144.205.151.232 C6/1/0/U0 online(pt)  13    -0.25   1920  1    N
0002.fdfa.137d 144.205.151.160 C6/1/0/U0 !online     16    -0.50   1920  1    N
0003.e38f.e9ab 144.205.151.237 C6/1/0/U0 !online     3     -0.50   1926  1    N
0003.e3a6.8173 144.205.151.179 C6/1/1/U2 offline     4     0.50    1929  0    N
0003.e3a6.8195 144.205.151.219 C6/1/1/U2 !online(pt) 22    -0.50   1929  1    N
0006.28dc.37fd 144.205.151.244 C6/1/1/U2 online(pt)  61    0.00    1925  2    N
0006.28e9.81c9 144.205.151.138 C6/1/1/U2 online(pt)  2     0.75    1925  1    N
0006.28f9.8bbd 144.205.151.134 C6/1/1/U2 online      25    -0.25   1924  1    N
0006.28f9.9d19 144.205.151.144 C6/1/1/U2 online(pt)  28    0.25    1924  1    N
0010.7bed.9b6d 144.205.151.228 C6/1/1/U2 online(pt)  59    0.25    1554  1    N
0002.fdfa.12db 144.205.151.234 C7/0/0/U0 online      15    -0.75   1914  1    N
0002.fdfa.138d 144.205.151.140 C7/0/0/U5 online      4     0.00    1917  1    N
0003.e38f.e85b 144.205.151.214 C7/0/0/U5 !online     17    0.25    1919  1    N
0003.e38f.f4cb 144.205.151.238 C7/0/0/U5 online(pt)  16    0.00   !2750  1    N
0003.e3a6.7fd9 144.205.151.151 C7/0/0/U5 online      1     0.25    1922  0    N
0020.4005.3f06 144.205.151.145 C7/0/0/U0 online(pt)  2     0.00    1901  1    N
0020.4006.b010 144.205.151.164 C7/0/0/U5 online(pt)  3     0.00    1901  1    N
0050.7302.3d83 144.205.151.240 C7/0/0/U0 online(pt)  18    -0.25   1543  1    N
00b0.6478.ae8d 144.205.151.254 C7/0/0/U5 online(pt)  44    0.25    1920  21   N
00d0.bad3.c0cd 144.205.151.149 C7/0/0/U5 online      19    0.25    1543  1    N
00d0.bad3.c0cf 144.205.151.194 C7/0/0/U0 online      13    0.00    1546  1    N
00d0.bad3.c0d5 144.205.151.133 C7/0/0/U0 online      12    0.50    1546  1    N
Router#
```

You can also use the **show cable modem rogue** command to display only those cable modems that have been rejected for failing the dynamic shared-secret authentication checks:

```
Router# show cable modem rogue
                                    Spoof TFTP
MAC Address    Vendor     Interface  Count Dnld Dynamic Secret
AAAA.7b43.aa7f Vendor1    C4/0/U5      2  Yes  45494DC933F8F47A398F69EE6361B017
AAAA.7b43.aa7f Vendor1    C4/0/U5      2  Yes  D47BCBB5494E9936D51CB0EB66EF0B0A
BBBB.7b43.aa7f Vendor2    C4/0/U5      2  No   8EB196423170B26684BF6730C099D271
AAAA.7b43.aa7f Vendor1    C4/0/U5      2  No   DF8FE30203010001A326302430120603
BBBB.7b43.aa7f Vendor2    C4/0/U5      2  No   300E0603551D0F0101FF040403020106
AAAA.7b43.aa7f Vendor1    C4/0/U5      2  Yes  820101002D1A264CE212A1BB6C1728B3
DDDD.7b43.aa7f Vendor4    C4/0/U5      2  Yes  7935B694DCA90BC624AC92A519C214B9
AAAA.7b43.aa7f Vendor1    C4/0/U5      2  No   3AB096D00D56ECD07D9B7AB662451CFF
Router#
```

Displaying the Current Dynamic Secrets

In Cisco IOS XE Everest 16.5.1 , the **verbose** option for the **show cable modem** command displays the dynamically generated shared secret (a 16-byte hexadecimal value) that was used in the cable modem's previous registration cycle. The display also shows if the cable modem failed the dynamic shared-secret check or did not download the DOCSIS configuration file from the TFTP server. If a cable modem is offline, its dynamic secret is shown as all zeros.

For example, the following example shows a typical display for a single cable modem that failed the dynamic shared-secret check:

```
Router# show cable modem 00c0.73ee.bbaa verbose

MAC Address                     : 00c0.73ee.bbaa
IP Address                      : 3.18.1.6
Prim Sid                        : 2
QoS Profile Index               : 6
Interface                       : C3/0/U0
Upstream Power                  : 0.00 dBmV (SNR = 26.92 dBmV)
Downstream Power                : 0.00 dBmV (SNR = ----- dBmV)
Timing Offset                   :  2812
Initial Timing Offset           : 2812
Received Power                  :  0.00
MAC Version                     : DOC1.0
Provisioned Mode                : DOC1.0
Capabilities                    : {Frag=N, Concat=N, PHS=N, Priv=BPI}
Sid/Said Limit                  : {Max Us Sids=0, Max Ds Saids=0}
Optional Filtering Support      : {802.1P=N, 802.1Q=N}
Transmit Equalizer Support      : {Taps/Symbol= 0, Num of Taps= 0}
Number of CPE IPs               : 0(Max CPE IPs = 1)
CFG Max-CPE                     : 1
Flaps                           : 26(Feb 14 02:35:39)
Errors                          : 0 CRCs, 0 HCSes
Stn Mtn Failures                : 6 aborts, 0 exhausted
Total US Flows                  : 1(1 active)
Total DS Flows                  : 1(1 active)
Total US Data                   : 0 packets, 0 bytes
Total US Throughput             : 0 bits/sec, 0 packets/sec
Total DS Data                   : 0 packets, 0 bytes
Total DS Throughput             : 0 bits/sec, 0 packets/sec
Active Classifiers              : 0 (Max = NO LIMIT)
Dynamic Secret                  : A3D1028F36EBD54FDCC2F74719664D3F
Router#
```

The following example shows a typical display for a single cable modem that is currently offline (the Dynamic Secret field shows all zeros):

```
Router# show cable modem 00C0.6914.8601 verbose

MAC Address                     : 00C0.6914.8601
IP Address                      : 10.212.192.119
Prim Sid                        : 6231
QoS Profile Index               : 2
Interface                       : C5/1/0/U3
Upstream Power                  : 0.00 dBmV (SNR = 30.19 dBmV)
Downstream Power                : 0.00 dBmV (SNR = ----- dBmV)
Timing Offset                   : 1831
Initial Timing Offset           : 1831
Received Power                  : !-2.25
MAC Version                     : DOC1.0
```

```
Provisioned Mode                    : DOC1.0
Capabilities                        : {Frag=N, Concat=Y, PHS=N, Priv=BPI}
Sid/Said Limit                      : {Max Us Sids=0, Max Ds Saids=0}
Optional Filtering Support          : {802.1P=N, 802.1Q=N}
Transmit Equalizer Support          : {Taps/Symbol= 0, Num of Taps= 0}
Number of CPE IPs                   : 4(Max CPE IPs = 4)
CFG Max-CPE                         : 4
Flaps                               : 20638(Feb 10 16:04:10)
Errors                              : 0 CRCs, 0 HCSes
Stn Mtn Failures                    : 108 aborts, 161 exhausted
Total US Flows                      : 1(1 active)
Total DS Flows                      : 1(1 active)
Total US Data                       : 236222 packets, 146630868 bytes
Total US Throughput                 : 0 bits/sec, 0 packets/sec
Total DS Data                       : 9 packets, 1114 bytes
Total DS Throughput                 : 0 bits/sec, 0 packets/sec
Active Classifiers                  : 0 (Max = NO LIMIT)
Dynamic Secret                      : 00000000000000000000000000000000
Router#
```

Note The Dynamic Secret field shown above is all zeros ("00000000000000000000000000000000"), which indicates that this cable modem is offline.

You can also use the following command to display all the dynamically generated shared secrets that are in use:

```
Router# show cable modem verbose | include Dynamic Secret

Dynamic Secret                      : 434330364346443446433038413132 37
Dynamic Secret                      : 308203E0308202C8A003020102021058
Dynamic Secret                      : 0D06092A864886F70D01010505003081
Dynamic Secret                      : 3037060355040A133044617461204F76
Dynamic Secret                      : 20496E7465726661636520537065636 9
Dynamic Secret                      : 00000000000000000000000000000000
Dynamic Secret                      : 040B130C4361626C65204D6F64656D73
Dynamic Secret                      : 53204361626C65204D6F64656D20526F
Dynamic Secret                      : 7574686F72697479301E170D30313032
Dynamic Secret                      : 313233353935395A308197310B300906
Dynamic Secret                      : 0A133044617461204F76657220436162
Dynamic Secret                      : 666163652053706563696666963617469
Dynamic Secret                      : 626C65204D6F64656D73313630340603
Dynamic Secret                      : 65204D6F64656D20526F6F7420436572
Dynamic Secret                      : 747930820122300D06092A864886F70D
Dynamic Secret                      : 010100C0EF369D7BDAB0A938E6ED29C3
Dynamic Secret                      : DA398BF619A11B3C0F64912D133CFFB6
Dynamic Secret                      : FFAD6CE01590ABF5A1A0F50AC05221F2
Dynamic Secret                      : 73504BCA8278D41CAD50D9849B56552D
Dynamic Secret                      : 05F4655F2981E031EB76C90F9B3100D1
Dynamic Secret                      : F4CB0BF4A13EA9512FDE4A2A219C27E9
Dynamic Secret                      : D47BCBB5494E9936D51CB0EB66EF0B0A
Dynamic Secret                      : 8EB196423170B26684BF6730C099D271
Dynamic Secret                      : DF8FE30203010001A326302430120603
Dynamic Secret                      : 300E0603551D0F0101FF040403020106
Dynamic Secret                      : 820101002D1A264CE212A1BB6C1728B3
Dynamic Secret                      : 7935B694DCA90BC624AC92A519C214B9
Dynamic Secret                      : 3AB096D00D56ECD07D9B7AB662451CFF
Dynamic Secret                      : 92E68CFD8783D58557E3994F23A8140F
Dynamic Secret                      : 225A3B01DB67AF0C3637A765E1E7C329
Dynamic Secret                      : 2BB1E6221B6D5596F3D6F506804C995E
```

```
Dynamic Secret                      : 45494DC933F8F47A398F69EE6361B017
Router#
```

Troubleshooting Cable Modems with Dynamic Shared Secret

If a cable modem is being marked as having violated the dynamic shared secret, you can enable the following debugs to get more information about the sequence of events that is occurring:

- **debug cable mac-address** *cm-mac-addr* **verbose**—Enables detailed debugging for the cable modem with the specific MAC address.
- **debug cable tlv**—Displays the contents of Type/Length/Value messages that are sent during the registration process.
- **debug cable dynamic-secret**—Displays debugging messages about dynamic shared secret operation.
- **debug tftp server events**—Displays debugging messages for the major events that occur with the Cisco CMTS router's onboard TFTP server.
- **debug tftp server packets**—Displays a packet dump for the DOCSIS configuration files that the TFTP server downloads to a cable modem.

Tip For more information about these debug commands, see the *Cisco CMTS Debugging Commands* chapter in the Cisco Broadband Cable Command Reference Guide, at the following URL: http://www.cisco.com/en/US/docs/ios/cable/command/reference/cbl_book.html

In addition, examine the messages in the router's log buffer for any helpful information. Use the **show logging** command to display the contents of the router's logging buffer to display these messages. You can limit the output to a specific hour and minute by using the **begin** output modifier. For example, to display only those messages that were recorded at 12:10, give the following command:

```
Router# show logging | begin 12:10
```

Note The exact format for the **begin** output modifier depends on the timestamp you are using for your logging buffer.

Configuration Examples for Dynamic Shared Secret

This section lists a typical configuration for the Dynamic Shared Secret feature.

Note These configurations also show a shared secret and secondary secret being configured on the cable interface. This is optional but highly recommended, because it adds an additional layer of security during the registration of cable modems.

Mark Configuration: Example

The following excerpt from a configuration for the cable interface on a Cisco CMTS router configures the cable interface so that cable modems that fail the CMTS MIC check are allowed to come online, but are marked with an exclamation point (!) in the **show cable modem** displays, so that the situation can be investigated further.

```
interface cable c5/1/0
 cable dynamic-secret mark
 ...
```

Lock Configuration: Example

The following excerpt from a configuration for the cable interface on a Cisco CMTS router configures the cable interface so that cable modems that fail the CMTS MIC check are allowed to come online, but are locked into a restrictive QoS configuration that limits the upstream and downstream service flows to a maximum rate of 10 kbps. A locked cable modem remains locked into the restrictive QoS configuration until the modem has remained offline for more than 24 hours, or until you have manually cleared it using the **clear cable modem lock** command.

```
cable qos permission create
cable qos permission update
...
interface cable c3/0
 cable dynamic-secret lock
 ...
```

Note If you use the **lock** option without specifying a specific QoS profile, you must allow cable modems to create and update QoS profiles, using the **cable qos permission** command. If you do not do this and continue to use the **lock** option without specifying a particular QoS profile, locked cable modems will not be allowed to register until the lock clears or expires.

The following example is the same except that it specifies that the locked cable modem should be assigned QoS profile 90. The cable modem remains locked with this QoS profile until the modem has remained offline for more than 24 hours, or until you have manually cleared it using the **clear cable modem lock** command. Because a specific QoS profile is specified, you do not need to use the **cable qos permission** command.

```
interface cable c3/0
 cable dynamic-secret lock 90
 ...
```

Note When a locked modem is cleared, it is automatically reset so that it reregisters with the CMTS. It is allowed online with the requested QoS parameters if it registers with a valid DOCSIS configuration that passes the Dynamic Shared Secret checks. However, the modem is locked again if it violates the DOCSIS specifications again.

Reject Configuration: Example

The following excerpt from a configuration for the cable interface on a Cisco CMTS configures the cable interface so that cable modems that fail the CMTS MIC check are rejected and not allowed to register. The cable modem must reregister using a DOCSIS configuration file with a CMTS MIC that matches one of the shared secret or secondary secret values. When it does come online, the CMTS also prints a warning message on the console and marks the cable modem in the **show cable modem** command with an exclamation point (!), so that this situation can be investigated.

```
interface cable c3/0
 cable dynamic-secret reject
 ...
```

Disabled Configuration: Example

The following excerpt from a configuration for the cable interface on a Cisco uBR7100 series router disables the Dynamic Shared Secret feature. In this configuration, the CMTS uses the shared secret and secondary shared secret values unchanged when verifying the CMTS MIC value for each DOCSIS configuration file.

```
interface cable c1/0
 no cable dynamic-secret
 ...
```

Additional References

For additional information related to Dynamic Shared Secret, refer to the following references:

Standards

Standards[9]	Title
SP-RFIv1.1-I09-020830	Data-over-Cable Service Interface Specifications Radio Frequency Interface Specification, version 1.1

[9] Not all supported standards are listed.

MIBs

MIBs[10]	MIBs Link
No new or modified MIB objects are supported by the Dynamic Shared Secret feature. • CISCO-DOCS-EXT-MIB—Includes attributes to configure the Dynamic Shared Secret feature and to generate traps when a cable modem fails the shared-secret security checks.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

[10] Not all supported MIBs are listed.

RFCs

RFCs[11]	Title
RFC 2233	DOCSIS OSSI Objects Support
RFC 2665	DOCSIS Ethernet MIB Objects Support
RFC 2669	Cable Device MIB

[11] Not all supported RFCs are listed.

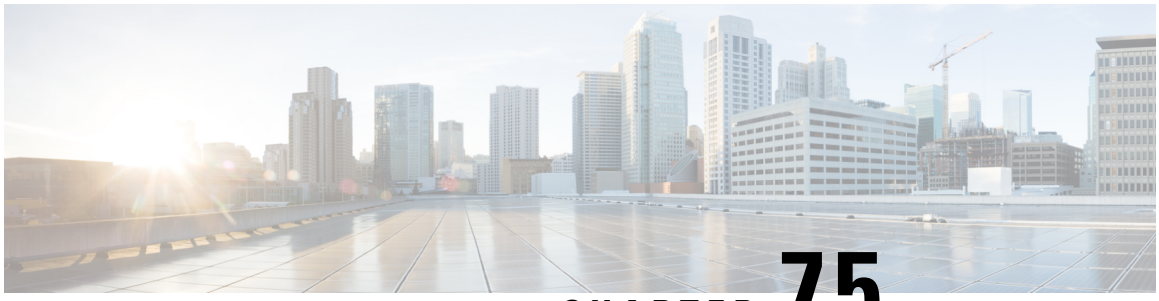
Feature Information for Dynamic Shared Secret

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 188: Feature Information for Downstream Interface Configuration

Feature Name	Releases	Feature Information
Dynamic shared secret	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router .



CHAPTER 75

Lawful Intercept Architecture

The Lawful Intercept (LI) feature supports service providers in meeting the requirements of law enforcement agencies to provide the ability to intercept Voice-over-Internet protocol (VoIP) or data traffic going through the edge routers. This document explains LI architecture, including Cisco Service Independent Intercept architecture and PacketCable Lawful Intercept architecture. It also describes the components of the LI feature and provides instructions on how to configure the LI feature in your system.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1163
- Prerequisites for Lawful Intercept, on page 1164
- Restrictions for Lawful Intercept, on page 1165
- Information About Lawful Intercept, on page 1165
- How to Configure Lawful Intercept, on page 1169
- Configuration Examples for Lawful Intercept, on page 1173
- Additional References, on page 1174
- Feature Information for Lawful Intercept, on page 1175

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 189: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Lawful Intercept

Access to the Cisco LI MIB view should be restricted to the mediation device and to system administrators who need to be aware of lawful intercepts on the router. To access the MIB, users must have level-15 access rights on the router.

Communication with Mediation Device

For the router to communicate with the mediation device to execute a lawful intercept, the following configuration requirements must be met:

- The domain name for both the router and the mediation device must be registered in the Domain Name System (DNS).

In DNS, the router IP address is typically the address of the TenGigabitEthernet5/1/0 or TenGigabitEthernet4/1/0 interface (depending on the slot in which the Supervisor is installed) on the router.

- The mediation device must have an access function (AF) and an access function provisioning interface (AFPI).
- You must add the mediation device to the Simple Network Management Protocol (SNMP) user group that has access to the CISCO-TAP2-MIB view. Specify the username of the mediation device as the user to add to the group.

When you add the mediation device as a CISCO-TAP2-MIB user, you can include the mediation device's authorization password if you want. The password must be at least eight characters in length.

Restrictions for Lawful Intercept

General Restrictions

There is no command-line interface (CLI) available to configure LI on the router. All error messages are sent to the mediation device as SNMP notifications. All intercepts are provisioned using SNMPv3 only.

Lawful Intercept does not support SUP HA. LI configuration needs to be reapplied after SUP switchover. An SNMP trap will be generated for this event.

Lawful Intercept MIBs

Only the mediation device and users who need to know about lawful intercepts are allowed to access the LI MIBs.

Due to its sensitive nature, the Cisco LI MIBs are only available in software images that support the LI feature. These MIBs are not accessible through the Network Management Software MIBs Support page (http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml).

SNMP Notifications

SNMP notifications for LI must be sent to User Datagram Protocol (UDP) port 161 on the mediation device, not port 162 (which is the SNMP default). For more information, see the Enabling SNMP Notifications for Lawful Intercept, on page 1171.

Information About Lawful Intercept

Introduction to Lawful Intercept

LI is the process by which law enforcement agencies (LEAs) conduct electronic surveillance as authorized by judicial or administrative order. Increasingly, legislation is being adopted and regulations are being enforced that require service providers (SPs) and Internet service providers (ISPs) to implement their networks to explicitly support authorized electronic surveillance. The types of SPs or ISPs that are subject to LI mandates vary greatly from country to country. LI compliance in the United States is specified by the Commission on Accreditation for Law Enforcement Agencies (CALEA).

Cisco supports two architectures for LI: PacketCable and Service Independent Intercept. The LI components by themselves do not ensure customer compliance with applicable regulations but rather provide tools that can be used by SPs and ISPs to construct an LI-compliant network.

Cisco Service Independent Intercept Architecture

The Cisco Service Independent Intercept Architecture Version 3.0 document describes implementation of LI for VoIP networks using the Cisco BTS 10200 Softswitch call agent, version 5.0, in a non-PacketCable

network. Packet Cable Event Message specification version 1.5-I01 is used to deliver the call identifying information along with version 2.0 of the Cisco Tap MIB for call content.

The Cisco Service Independent Intercept Architecture Version 2.0 document describes implementation of LI for VoIP networks using the Cisco BTS 10200 Softswitch call agent, versions 4.4 and 4.5, in a non-PacketCable network. Although not a PacketCable network, PacketCable Event Messages Specification version I08 is still used to deliver call identifying information, along with version 1.0 or version 2.0 of the Cisco Tap MIB for call content. The *Cisco Service Independent Intercept Architecture Version 2.0* document adds additional functionality for doing data intercepts by both IP address and session ID, which are both supported in version 2.0 of the Cisco Tap MIB (CISCO-TAP2-MIB).

The Cisco Service Independent Intercept Architecture Version 1.0 document describes implementation of LI for VoIP networks that are using the Cisco BTS 10200 Softswitch call agent, versions 3.5 and 4.1, in a non-PacketCable network. Although not a PacketCable network, PacketCable Event Message Specification version I03 is still used to deliver call identifying information, along with version 1.0 of the Cisco Tap MIB (CISCO-TAP-MIB) for call content. Simple data intercepts by IP address are also discussed.

PacketCable Lawful Intercept Architecture

The *PacketCable Lawful Intercept Architecture for BTS Version 5.0* document describes the implementation of LI for VoIP using Cisco BTS 10200 Softswitch call agent, version 5.0, in a PacketCable network that conforms to PacketCable Event Messages Specification version 1.5-I01.

The *PacketCable Lawful Intercept Architecture for BTS Versions 4.4 and 4.5* document describes the implementation of LI for VoIP using Cisco BTS 10200 Softswitch call agent, versions 4.4 and 4.5, in a PacketCable network that conforms to PacketCable Event Messages Specification version I08.

The PacketCable Lawful Intercept Architecture for BTS Versions 3.5 and 4.1 document describes the implementation of LI for voice over IP (VoIP) using Cisco Broadband Telephony Softswitch (BTS) 10200 Softswitch call agent, versions 3.5 and 4.1, in a PacketCable network that conforms to PacketCable Event Message Specification version I03.

The *PacketCable Control Point Discovery Interface Specification* document defines an IP-based protocol that can be used to discover a control point for a given IP address. The control point is the place where Quality of Service (QoS) operations, LI content tapping operations, or other operations may be performed.

Note The Cisco cBR router does not support PacketCable Communications Assistance for Law Enforcement Act (CALEA).

Cisco cBR Series Routers

The Cisco cBR series router support two types of LI: regular and broadband (per-subscriber). Regular wiretaps are executed on access subinterfaces and physical interfaces. Wiretaps are not required, and are not executed, on internal interfaces. The router determines which type of wiretap to execute based on the interface that the target's traffic is using.

LI on the Cisco cBR series routers can intercept traffic based on a combination of one or more of the following fields:

- Destination IP address and mask (IPv4 or IPv6 address)
- Destination port or destination port range

- Source IP address and mask (IPv4 or IPv6 address)
- Source port or source port range
- Protocol ID
- Type of Service (TOS)
- Virtual routing and forwarding (VRF) name, which is translated to a *vrf-tableid* value within the router.
- Subscriber (user) connection ID
- Cable modem
- MAC address

The LI implementation on the Cisco cBR series routers is provisioned using SNMP3 and supports the following functionality:

- Interception of communication content. The router duplicates each intercepted packet and then places the copy of the packet within a UDP-header encapsulated packet (with a configured CCCid). The router sends the encapsulated packet to the LI mediation device. Even if multiple lawful intercepts are configured on the same data flow, only one copy of the packet is sent to the mediation device. If necessary, the mediation device can duplicate the packet for each LEA.
- Interception of IPv4, IPv4 multicast, IPv6, and IPv6 multicast flows.

LI includes two ways of setting a MAC-based tap:

- On CPE—Only intercepts traffic whose source or destination match the MAC address of the CPE device.
- On CM—Intercepts all of the traffic behind the CM, including the CM traffic itself. This form of intercept might generate a lot of traffic to the mediation device.

VRF Aware LI

VRF Aware LI is the ability to provision a LI wiretap on IPv4 data in a particular Virtual Private Network (VPN). This feature allows a LEA to lawfully intercept targeted data within that VPN. Only IPv4 data within that VPN is subject to the VRF-based LI tap.

VRF Aware LI is available for the following types of traffic:

- ip2ip
- ip2tag (IP to MPLS)
- tag2ip (MPLS to IP)

To provision a VPN-based IPv4 tap, the LI administrative function (running on the mediation device) uses the CISCO-IP-TAP-MIB to identify the name of the VRF table that the targeted VPN uses. The VRF name is used to select the VPN interfaces on which to enable LI in order to execute the tap.

The router determines which traffic to intercept and which mediation device to send the intercepted packets based on the VRF name (along with the source and destination address, source and destination port, and protocol).

Note When using the Cisco-IP-TAP-MIB, if the VRF name is not specified in the stream entry, the global IP routing table is used by default.

Lawful Intercept- Redundant Mediation Devices

The Cisco cBR Series Converged Broadband Routers supports replicating Lawful Intercept (LI) packets to multiple Mediation Devices (MDs). To use this feature, multiple identical taps are configured. The Cisco cBR Series Converged Broadband Routers support up to two identical taps to replicate to two MDs. Only MAC- and CM-taps are supported with multiple MDs.

For a sample SNMP configuration command set to configure two identical taps to tap to two MDs, see Example: Configuring Lawful Intercept- Redundant Mediation Devices, on page 1174.

Lawful Intercept MIBs

Due to its sensitive nature, the Cisco LI MIBs are only available in software images that support the LI feature. These MIBs are not accessible through the Network Management Software MIBs Support page (http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml).

Restricting Access to the Lawful Intercept MIBs

Only the mediation device and users who need to know about lawful intercepts should be allowed to access the LI MIBs. To restrict access to these MIBs, you must:

1. Create a view that includes the Cisco LI MIBs.
2. Create an SNMP user group that has read-and-write access to the view. Only users assigned to this user group can access information in the MIBs.
3. Add users to the Cisco LI user groups to define who can access the MIBs and any information related to lawful intercepts. Be sure to add the mediation device as a user in this group; otherwise, the router cannot perform lawful intercepts.

For more information, see the Creating a Restricted SNMP View of Lawful Intercept MIBs module.

Note Access to the Cisco LI MIB view should be restricted to the mediation device and to system administrators who need to be aware of lawful intercepts on the router. To access the MIB, users must have level-15 access rights on the router.

Service Independent Intercept

Cisco developed the Service Independent Intercept (SII) architecture in response to requirements that support lawful intercept for service provider customers. The SII architecture offers well-defined, open interfaces between the Cisco equipment acting as the content Intercept Access Point (IAP) and the mediation device. The modular nature of the SII architecture allows the service provider to choose the most appropriate mediation

device to meet specific network requirements and regional, standards-based requirements for the interface to the law enforcement collection function.

The mediation device uses SNMPv3 to instruct the call connect (CC) IAP to replicate the CC and send the content to the mediation device. The CC IAP can be either an edge router or a trunking gateway for voice, and either an edge router or an access server for data.

Note The Cisco cBR router does not support encryption of lawful intercept traffic.

To increase the security and to mitigate any SNMPv3 vulnerability, the following task is required:

Restricting Access to Trusted Hosts (without Encryption)

SNMPv3 provides support for both security models and security levels. A security model is an authentication strategy that is set up for a user and the group in which the user resides. A security level is the permitted level of security within a security model. A combination of a security model and a security level will determine the security mechanism employed when handling an SNMP packet.

Additionally, the SNMP Support for the Named Access Lists feature adds support for standard named access control lists (ACLs) to several SNMP commands.

To configure a new SNMP group or a table that maps SNMP users to SNMP views, use the **snmp-server group** command in global configuration mode.

```
access-list my-list permit ip host 10.10.10.1
snmp-server group my-group v3 auth access my-list
```

In this example, the access list named **my-list** allows SNMP traffic only from 10.10.10.1. This access list is then applied to the SNMP group called **my-group** .

How to Configure Lawful Intercept

Although there are no direct user commands to provision lawful intercept on the router, you do need to perform some configuration tasks, such as providing access to LI MIBs, and setting up SNMP notifications. This section describes how to perform the required tasks:

Creating a Restricted SNMP View of Lawful Intercept MIBs

To create and assign users to an SNMP view that includes the Cisco lawful intercept MIBs, perform the steps in this section.

Before you begin

- You must issue the commands in global configuration mode with level-15 access rights.
- SNMPv3 must be configured on the device.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server view** *view-name MIB-name* **included** **Example:** `Device(config)# snmp-server view exampleView ciscoTap2MIB included`	Creates an SNMP view that includes the CISCO-TAP2-MIB (where *exampleView*is the name of the view to create for the MIB). • This MIB is required for both regular and broadband lawful intercept.
Step 4	**snmp-server view** *view-name MIB-name* **included** **Example:** `Device(config)# snmp-server view exampleView ciscoIpTapMIB included`	Adds the CISCO-IP-TAP-MIB to the SNMP view.
Step 5	**snmp-server view** *view-name MIB-name* **included** **Example:** `Device(config)# snmp-server view exampleView cisco802TapMIB included`	Adds the CISCO-802-TAP-MIB to the SNMP view.
Step 6	**snmp-server group** *group-name* **v3 noauth read** *view-name* **write** *view-name* **Example:** `Device(config)# snmp-server group exampleGroup v3 noauth read exampleView write exampleView`	Creates an SNMP user group that has access to the LI MIB view and defines the group's access rights to the view.
Step 7	**snmp-server user** *user-name group-name* **v3 auth md5** *auth-password* **Example:** `Device(config)# snmp-server user exampleUser exampleGroup v3 auth md5 examplePassword`	Adds users to the specified user group.

	Command or Action	Purpose
Step 8	**end** **Example:** `Device(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Where to Go Next

The mediation device can now access the lawful intercept MIBs and issue SNMP **set** and **get** requests to configure and run lawful intercepts on the router. To configure the router to send SNMP notification to the mediation device, see the Enabling SNMP Notifications for Lawful Intercept.

Enabling SNMP Notifications for Lawful Intercept

SNMP automatically generates notifications for lawful intercept events. To configure the router to send lawful intercept notifications to the mediation device, perform the steps in this section.

Before you begin

- You must issue the commands in global configuration mode with level-15 access rights.
- SNMPv3 must be configured on the router.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server host** *ip-address* **community-string udp-port** *port notification-type* **Example:** `Device(config)# snmp-server 10.2.2.1 community-string udp-port 161 udp`	Specifies the IP address of the mediation device and the password-like community-string that is sent with a notification request. • For lawful intercept, the **udp-port** must be 161 and not 162 (the SNMP default).
Step 4	**snmp-server enable traps snmp authentication linkup linkdown coldstart warmstart**	Configures the router to send RFC 1157 notifications to the mediation device.

	Command or Action	Purpose
	Example: `Device(config)# snmp-server enable traps snmp authentication linkup linkdown coldstart warmstart`	• These notifications indicate authentication failures, link status (up or down), and router restarts.
Step 5	**end** **Example:** `Device(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Disabling SNMP Notifications

To disable SNMP notifications on the router, perform the steps in this section.

Note To disable lawful intercept notifications, use SNMPv3 to set the CISCO-TAP2-MIB object cTap2MediationNotificationEnable to false(2). To reenable lawful intercept notifications through SNMPv3, reset the object to true(1).

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no snmp-server enable traps** **Example:** `Device(config)# no snmp-server enable traps`	Disables all SNMP notification types that are available on your system.
Step 4	**end** **Example:** `Device(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Provisioning a MAC Intercept for Cable Modems Using SNMPv3

1. Configure the c802tapStreamInterface object.
2. Set the following bit flags in the c802tapStreamFields object:
   - dstMacAddress (bit 1)
   - srcMacAddress (bit 2)
   - cmMacAddress (bit 6)—The cmMacAddress bit field is newly introduced for cable modem support and determines whether the intercept is a CPE-based or CM-based intercept.
3. Configure the following objects with the same CM MAC address value:
   - c802tapStreamDestinationAddress
   - c802tapStreamSourceAddress

Provisioning a MAC Intercept for a CPE Device Using SNMPv3

1. Configure the c802tapStreamInterface object.
2. Set the following bit flags in the c802tapStreamFields object:
   - dstMacAddress (bit 1)
   - srcMacAddress (bit 2)
3. Configure the following objects with the same CPE MAC address value:
   - c802tapStreamDestinationAddress
   - c802tapStreamSourceAddress

Configuration Examples for Lawful Intercept

Example: Enabling Mediation Device Access Lawful Intercept MIBs

The following example shows how to enable the mediation device to access the lawful intercept MIBs. It creates an SNMP view (tapV) that includes four LI MIBs (CISCO-TAP2-MIB, CISCO-IP-TAP-MIB, CISCO-802-TAP-MIB, and CISCO-USER-CONNECTION-TAP-MIB). It also creates a user group that has read, write, and notify access to MIBs in the tapV view.

```
snmp-server view tapV ciscoTap2MIB included
snmp-server view tapV ciscoIpTapMIB included
snmp-server view tapV cisco802TapMIB included
snmp-server view tapV ciscoUserConnectionTapMIB included
snmp-server group tapGrp v3 noauth read tapV write tapV notify tapV
snmp-server user MDuser tapGrp v3 auth md5 MDpasswd
snmp-server engineID local 1234
```

Example: Configuring Lawful Intercept- Redundant Mediation Devices

Lawful Intercept is configured using SNMPv3. The following example shows SNMP configuration command set to configure two identical taps to tap two MDs:

- Setup MD1:

```
setany -v3 -timeout 30 -retries 3 10.12.0.34 user1 cTap2MediationStatus.1 -i 4
cTap2MediationDestAddressType.1 -i 1 cTap2MediationTimeout.1 -o 07:E0:04:01:B:15:1A:0
cTap2MediationTransport.1 -i 1 cTap2MediationSrcInterface.1 -i 0
cTap2MediationDestAddress.1 -o 0a:0a:00:35 cTap2MediationDestPort.1 -g 63
```

- Setup CM tap:

```
setany -v3 -timeout 30 -retries 3 10.12.0.34 user1 c802tapStreamStatus.1.2 -i 4
c802tapStreamFields.1.2 -o 62 c802tapStreamInterface.1.2 -i -1
c802tapStreamDestinationAddress.1.2 -o "c8 fb 26 a5 55 98" c802tapStreamSourceAddress.1.2
 -o "c8 fb 26 a5 55 98"
setany -v3 -timeout 30 -retries 3 10.12.0.34 user1 cTap2StreamStatus.1.2 -i 5
cTap2StreamType.1.2 -i 2 cTap2StreamInterceptEnable.1.2 -i 1 cTap2StreamStatus.1.2 -i
4
```

- Setup MD2:

```
setany -v3 -timeout 30 -retries 3 10.12.0.34 user1 cTap2MediationStatus.2 -i 4
cTap2MediationDestAddressType.2 -i 1 cTap2MediationTimeout.2 -o 07:E0:03:03:7:15:1A:0
cTap2MediationTransport.2 -i 1 cTap2MediationSrcInterface.2 -i 0
cTap2MediationDestAddress.2 -o 0a:0a:00:06 cTap2MediationDestPort.2 -g 63
```

- Setup CM tap:

```
setany -v3 -timeout 30 -retries 3 10.12.0.34 user1 c802tapStreamStatus.2.2 -i 4
c802tapStreamFields.2.2 -o 62 c802tapStreamInterface.2.2 -i -1
c802tapStreamDestinationAddress.2.2 -o "c8 fb 26 a5 55 98" c802tapStreamSourceAddress.2.2
 -o "c8 fb 26 a5 55 98"
setany -v3 -timeout 30 -retries 3 10.12.0.34 user1 cTap2StreamStatus.2.2 -i 5
cTap2StreamType.2.2 -i 2 cTap2StreamInterceptEnable.2.2 -i 1 cTap2StreamStatus.2.2 -i
4
```

- Get tapped packets count:

```
getmany -v3 -timeout 30 -retries 3 10.12.0.34 user1 cTap2StreamInterceptedPackets
```

Additional References

Related Documents

Related Topic	Document Title
Configuring SNMP Support	*Configuring SNMP Support*
Security commands	*Cisco IOS Security Command Reference*

Standards and RFCs

Standard/RFC	Title
PacketCable™ Control Point Discovery Interface Specification	*PacketCable ™ Control Point Discovery Interface Specification* (PKT-SP-CPD-I02-061013)
RFC-3924	*Cisco Architecture for Lawful Intercept in IP Networks*

MIBs

MIB	MIBs Link
• CISCO-TAP2-MIB • CISCO-IP-TAP-MIB • CISCO-802-TAP-MIB • CISCO-USER-CONNECTION-TAP-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

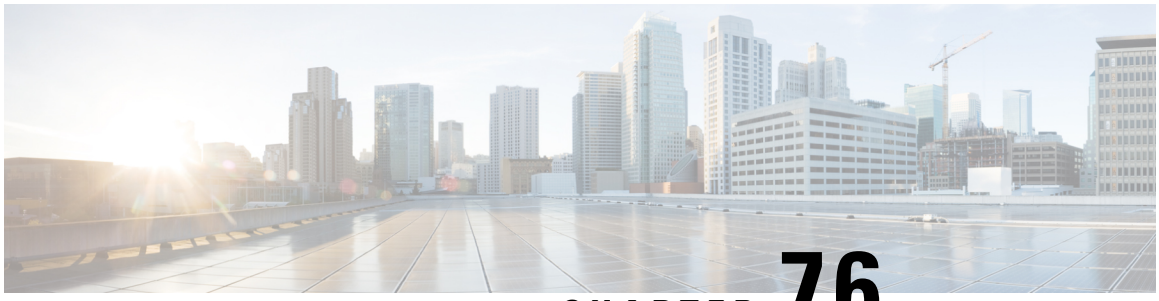
Feature Information for Lawful Intercept

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 190: Feature Information for Lawful Intercept

Feature Name	Releases	Feature Information
Lawful intercept - Redundant mediation devices	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 76

Cable Monitoring Feature for Cisco cBR Series Routers

After you configure cable monitoring, the router forwards copies of selected packets on the cable interface to an external LAN analyzer attached to another interface on the Cisco CMTS router. This command can help in troubleshooting network and application problems.

Note This feature does not monitor traffic for the purpose of preventing denial-of-service attacks and other types of network attacks. Even after configuring the cable monitoring feature, the traffic continues to its original destination, and only copies of the selected packets are forwarded to the CALEA server or LAN analyzer.

Note This feature doesn't support line card high availability (LCHA).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Overview of Cable Monitor Command for cBR, on page 1178
- Configuring Cable Monitoring for cBR Routers, on page 1178
- Capturing Sniffed Packets, on page 1180
- Feature Information for Cable Monitoring, on page 1183

Overview of Cable Monitor Command for cBR

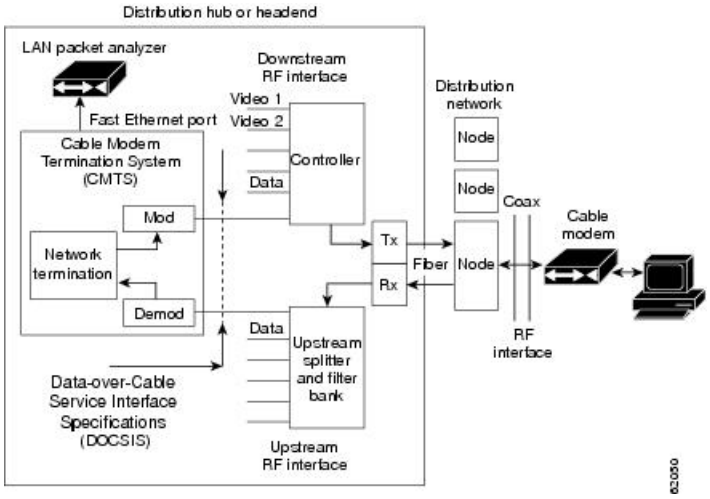
The **cable monitor** command sends copies of packets for specific types of traffic that is sent over a particular cable interface to a LAN analyzer, for use in troubleshooting network problems. This command can select packets to be forwarded using one or more of the following parameters:

- Either incoming or outbound packets
- Packets that match a specific MAC address (source and destination)
- Packets with a specific Service ID (SID)

Packets can also be timestamped to aid in troubleshooting. The packets are then forwarded out of the specified 10 Gigabit Ethernet port to the LAN analyzer for additional analysis.

The figure below illustrates a LAN packet analyzer attached to a Fast Ethernet port in a DOCSIS two-way configuration.

Figure 34: LAN Packet Analyzer in a DOCSIS Two-Way Configuration



Note The WAN port used for cable monitoring should be exclusively used by the LAN packet analyzer.

Configuring Cable Monitoring for cBR Routers

To enable the cable traffic monitoring feature on a particular cable interface, use the following procedure, starting in privileged EXEC mode.

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	Example:
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	Example:
	`Router(config)#`
		Enters global configuration mode.

Step 3	**cable monitor**
	Example:
	`Router(config)# cable monitor`
	Example:
	`Router(config-cable-monitor)#`
		Enters cable monitor configuration mode.

Step 4	**sniff card <slot num> <ds/us> <sniff point> <filter> dest cmon-tunnel <cmon-tunnel num>**
	Example:
	Downstream traffic: For each channel
	`Router(config-cable-monitor)sniff card 3 outbound`
	`docsis integrated-Cable 3/0/0:0 dest cmon-tunnel 3`
	Example:
	Downstream traffic: For each wideband channel
	`Router(config-cable-monitor)sniff card 3 outbound`
	`pre-docsis wideband-Cable 3/0/0:0 dest cmon-tunnel 3`
	Example:
	Downstream traffic: For each MAC address
	`Router(config-cable-monitor)sniff card 3 outbound`
	`docsis mac-address 0100.5e01.0101 dest cmon-tunnel 3`
	Example:
	Upstream traffic: For each channel
		Configures the card to forward the sniffed packets.
		• **slot number**—Slot number of the line card
		• **ds/us**—Downstream or upstream
		• **sniff point**—Sniff point in downstream or upstream FPGA (field-programmable gate array)
		• **filter**—Packet type filter
		• **dest cmon-tunnel**—Cable monitor tunnel for captured packets
		• **cmon-tunnel num**—Cable monitor tunnel number for capture packets

	Command or Action	Purpose

	`Router(config-cable-monitor)# sniff card 3 incoming post-docsis upstream-cable 3/0/0 us-channel 0 dest cmon-tunnel 3`
	Example:
	Upstream traffic: For each MAC address (cable modem or CPE)
	`Router(config-cable-monitor)#sniff card 3 incoming post-docsis mac-address e448.c70c.9c27 dest cmon-tunnel 3`
	Example:
	Upstream traffic: For MD/SID
	`Router(config-cable-monitor)#sniff card 3 incoming post-docsis cable 3/0/0 sid 12 upstream 0 dest cmon-tunnel 3`	

Step 5	**end**
	Example:
	`Router(config)# end`
	Example:
	`Router#`
		Exits global configuration mode.

What to do next

You can capture and forward the sniffed packets to an external server or a local hard disk. For more details, see Capturing Sniffed Packets, on page 1180.

Capturing Sniffed Packets

To forward the captured traffic to an external server, you should configure a tunnel. The external server might not be directly connected and can be away from CMTS.

To capture sniffed packets, you can follow one of these procedures:

- Capture output packets using an external host
- Capture packets by locating the hard disk

Capturing Sniffed Packets on an External Host

To forward the captured traffic to an external server, you should configure a tunnel. The external server might not be directly connected and can be away from CMTS.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`Router# configure terminal`
	Example:
	`Router(config)#`
		Enters global configuration mode.

Step 2	**interface cmon-tunnel number**
	Example:
	`Router(config)# interface CMON-Tunnel 3`
	`Router(config-if)#`
		Enters the interface cmon-tunnel mode to capture sniffed packets.

Step 3	**tunnel destination IP address**, **tunnel source IP address**
	Example:
	`Router(config-if)#tunnel destination 10.10.21.11`
	`Router(config-if)#tunnel source 10.10.21.1`
		Configures destination IP address and the source IP address for an external host to capture output packets.

Step 4	**end**
	Example:
	`Router(config)# end`
	Example:
	`Router#`
		Exits global configuration mode.

What to do next

Decode the captured packets by using a Wireshark plug-in.

Capturing Sniffed Packets on a Local Hard Drive

To forward the captured traffic to a local hard disk, use the following procedure.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`Router#configure CMON-Tunnel 3`
	Example:
	`Router(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 2	**interface cmon-tunnel number**
	Example:
	`Router(config)#interface CMON-Tunnel 3`
	`Router(config-if)#`
		Enters the interface cmon-tunnel mode.

Step 3	**mode buffer** **Example:** `Router(config-if)#mode buffer`	Enables mode buffer in the cmon-tunnel to capture packets by locating the hard disk.

Step 4	**end**
	Example:
	`Router(config-if)#end`
	`Router#`
		Exits global configuration mode.

Step 5	**show platform software interface fp active name-string CMON-Tunnel number**
	Example:
	`Router# show platform software interface fp active name-string CMON-Tunnel3`
	`Name: CMON-Tunnel3, ID: 131074, QFP ID: 11745, Schedules: 0`
	`Type: CABLE-MONITOR, State: enabled, SNMP ID: 0, MTU: 0`
	`IP Address: 0.0.0.0`
	`IPV6 Address: ::`
	`Flags: unknown`
	`ICMP Flags: unreachables, no-redirects, no-info-reply, no-mask-reply`
	`ICMP6 Flags: unreachables, no-redirects`
	`SMI enabled on protocol(s): UNKNOWN`
	`Authenticated-user:`
	`FRR linkdown ID: 65535`
	`Monitor Type: 0, Instance ID: 3, Mode: 3`
	`Monitor Tunnel Source: 0.0.0.0,`
	`Destination: 0.0.0.0`
	`vNet Name: , vNet Tag: 0, vNet Extra Information: 0`
	`Dirty: unknown`
	`AOM dependency sanity check: PASS`
	`AOM Obj ID: 24094`
		Gets the QFP ID.

Step 6	**test platform hardware qfp active feature docsis cmon-copy 3 QFP_ID**
	Example:
	`Router# test platform hardware qfp active feature docsis cmon-copy 3 11745`
	`Router #dir harddisk: | in CMON`
	`50 -rw- 24 Mar 5 2020 12:33:42 +02:00 CMON_3_20200305-123342.pcap`
		Uses the QFP ID to copy the buffer to the harddisk.

What to do next

Decode the captured packets by using a Wireshark plug-in.

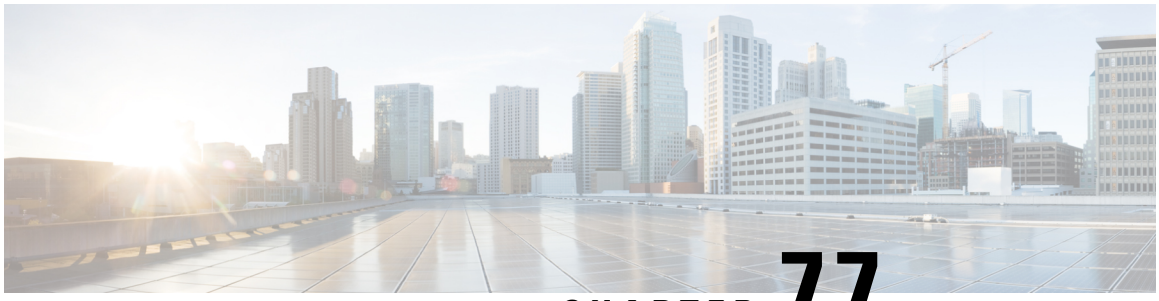
Feature Information for Cable Monitoring

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 191: Feature Information for Cable Monitoring

Feature Name	Releases	Feature Information
Cable Monitoring	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 77

Source-Based Rate Limit

The Source-Based Rate Limit (SBRL) feature prevents congestion of packets on the forwarding processor (FP) to the Route Processor (RP) interface, which can be caused by denial of service (DoS) attacks directed at the Cisco CMTS or by faulty hardware.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1185
- Prerequisites for Source-Based Rate Limit, on page 1186
- Restrictions for Source-Based Rate Limit, on page 1186
- Information About Source-Based Rate Limit, on page 1186
- How to Configure Source-Based Rate Limit, on page 1187
- Verifying the Source-Based Rate Limit Configuration, on page 1194
- Configuration Example for Source-Based Rate Limit, on page 1197
- Conversion of Divert Rate Limit Configuration on the Cisco uBR10012 Router to SBRL Configuration on the Cisco cBR Series Routers, on page 1199
- Additional References, on page 1202
- Feature Information for Source-Based Rate Limit, on page 1202

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 192: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Source-Based Rate Limit

- You must configure Control-Plane Policing (CoPP) for WAN-side SBRL.

Restrictions for Source-Based Rate Limit

- WAN-IP and Subscriber MAC address entities are identified using a hash, and hash collisions can occur between two (or more) entities.
- The Cisco cBR router does not perform special processing for hash collisions. The sources that hash-collide are rate-limited as if they are from the same source.
- The QOS group 99 is reserved for SBRL and cannot be used for other class maps.

Information About Source-Based Rate Limit

Source-Based Rate Limit (SBRL) feature operates on the punt path in CPP. SBRL identifies and rate-limits the packet streams that can overload the punt path or RP.

Punted packets are sent from the FP to the RP through the FP-to-RP queues. Denial of service (DoS) can occur when:

- The FP-to-RP queues are congested
- The RP cannot process punted packets fast enough

In both cases, the valid punted packets are not processed properly. These situations can be caused deliberately by DoS attacks or by faulty external hardware.

Packet streams identified by SBRL are rate-limited according to configured parameters. Rate-limiting occurs in CPP before the packets reach the FP-to-RP queues. This protects the RP, and also allows other valid punted packets to reach the RP.

By default, SBRL is disabled on the Cisco cBR router. SBRL has a separate configuration for the WAN-side and the subscriber-side.

WAN-Side Source-Based Rate Limit

WAN-side SBRL uses Control Plane Policing (CoPP). CoPP specifies the WAN-side packet streams that are directed for SBRL. Both trusted and untrusted sites can be specified using CoPP. Using CoPP, you can specify unlimited trusted sites. Access control list (ACL) is used to specify the trusted sites.

WAN-side SBRL also supports the quarantine functionality. When a packet stream enters quarantine, all punts from the packet stream are dropped for the configured period.

Subscriber-Side Source-Based Rate Limit

The subscriber-side SBRL configuration is global and does not need to be configured on each cable interface. The Cisco cBR router also supports per-cause subscriber-side configuration for Layer 3 mobility.

Note The default subscriber-side per-cause rate for Layer 3 mobility is 4 packets per second. The subscriber-side per-cause rate can be modified, however, it cannot be disabled.

How to Configure Source-Based Rate Limit

This section contains the following:

Configuring WAN-Side Source-Based Rate Limit

You must enable WAN-side SBRL in two parts:

1. Configure Control Plane Policing (CoPP) to specify which packets are subject to SBRL.
2. Configure WAN-side SBRL to set the rate-limiting parameters for the specified punt-causes.

In the CoPP policy map, the special action **set qos-group 99** denotes that the packets matching a particular class are subject to WAN-side SBRL. This means that the QOS group 99 is globally reserved for SBRL, and must not be used in other policy-maps.

Packets matching a class without **set qos-group 99** bypass WAN-side SBRL. This means that CoPP is also used to specify trusted traffic streams that are not subject to WAN-side SBRL.

All punted packets are subject to CoPP. So, you must ensure that subscriber-side traffic does not match a trusted class.

WAN-side SBRL identifies traffic streams by hashing the punt cause, VRF index, and source IP address. This value is used as the index for rate-limiting. The router does not perform special processing for hash collisions, so hash-colliding streams are treated as if they are from the same stream.

By default, WAN-side SBRL is disabled.

Restrictions

- All the punted packets are subject to CoPP and punt-policing.

This section contains the following:

Configuring Control Plane Policing

Punted packets matching the trusted class bypass WAN-side SBRL. The rest of the WAN-side punts are sent to WAN-side SBRL.

Note The following example shows a simple trusted class.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* **permit** *protocol* {**any** | **host** {*address* | *name*}} {**any** | **host** {*address* | *name*}} **tos** *tos* **Example:** `Router(config)# access-list 130 permit ip 192.168.1.10 0.0.0.0 192.168.1.11 0.0.0.0 tos 4`	Configures an access list for filtering frames by protocol type. **Note** Since all the punted packets are subject to CoPP, you must ensure that subscriber-side traffic does not match a trusted class.
Step 4	**class-map** *class-map-name* **Example:** `Router(config)# class-map match-all sbrl_v4_trusted`	Creates a class-map and enters QoS class-map configuration mode.

	Command or Action	Purpose
Step 5	**match access-group** *access-list-index* **Example:** `Router(config-cmap)# match access-group 130`	Specifies access groups to apply to an identity policy. The range of is from 1 to 2799.
Step 6	**exit** **Example:** `Router(config-cmap)# exit`	Exits QoS class-map configuration mode and returns to global configuration mode.
Step 7	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map copp_policy`	Specifies a service policy and enters QoS policy-map configuration mode.
Step 8	**class** *class-map-name* **Example:** `Router(config)# class sbrl_v4_trusted`	Enters QoS policy-map class configuration mode.
Step 9	**police rate** *units* **pps conform-action** *action* **exceed-action** *action* **Example:** `Router(config-pmap-c)# police rate 1000 pps conform-action transmit exceed-action transmit`	Polices traffic destined for the control plane at a specified rate. **Note** The rate is irrelevant if both the configured actions are **transmit**.
Step 10	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class police configuration mode
Step 11	**class class-default** **Example:** `Router(config-pmap)# class class-default`	Specifies the action to take on the packets that do not match any other class in the policy map.
Step 12	**set qos-group 99** **Example:** `Router(config-pmap-c)# set qos-group 99`	Enables WAN-side SBRL for the packets that match this class.
Step 13	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode
Step 14	**exit** **Example:** `Router(config-pmap)# exit`	Exits policy-map configuration mode

	Command or Action	Purpose
Step 15	**control-plane** [**host** | **transit** | **cef-exception**] **Example:** `Router(config)# control-plane`	Associates or modifies attributes (such as a service policy) that are associated with the control plane of the router and enters control plane configuration mode.
Step 16	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-cp)# service-policy input copp_policy`	Attaches a policy map to a control plane.
Step 17	**end** **Example:** `Router(config-cp)# end`	Exits control plane configuration mode and returns to privileged EXEC mode.

Enabling WAN-Side Source-Based Rate Limit

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**platform punt-sbrl wan punt-cause** *punt-cause* **rate** *rate* **Example:** `Router(config)# platform punt-sbrl wan punt-cause 10 rate 4`	Configures WAN-side rate limit. • **punt-cause** *punt-cause*—Specifies the punt cause. The range is from 1 to 107. • **rate** *rate*—Specifies the rate in packets per second. The range is from 1 to 256, specified in powers-of-2.

Configuring WAN-Side Quarantine

The WAN-side quarantine extends the WAN-side SBRL configuration. When a traffic stream enters quarantine, all punted packets in the stream are dropped for the configured period.

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**platform punt-sbrl wan punt-cause** *punt-cause* **rate** *rate* **quarantine-time** *time* **burst-factor** *burst-factor* **Example:** `Router(config)# platform punt-sbrl wan punt-cause 10 rate 4 quarantine-time 10 burst-factor 500`	Configures quarantine for the WAN-side packet stream. • **punt-cause** *punt-cause*—Specifies the punt cause. The range is from 1 to 107. • **rate** *rate*—Specifies the rate limit in packets per second. The range is from 1 to 256, specified in powers-of-2. • **quarantine-time** *time*—Specifies the quarantine time, in minutes. The range is from 1 to 60. • **burst-factor** *burst-factor*—Specifies the burst-factor, in number of packets. The range is from 50 to 1000.

Example

When (*burst-factor* x *rate*) packets arrive at a rate faster than *rate*, the packet stream enters quarantine.

For example, during a DoS attack, when the following occurs:

- Punted packets from a WAN-side source are arrive at 100 packets per second.
- WAN-side SBRL is configured with a rate of 4 packets per second, quarantine time of 10 minutes, and burst-factor of 500 packets.

The packet rate is significantly higher than the configured rate. Therefore, when 2000 (4 x 500) packets have arrived, the packet stream enters into quarantine. Quarantine is activated at 20 seconds (2000 packets per 100 packets per second), and all punted packets from the stream are dropped for 10 minutes. After 10 minutes, the quarantine is deactivated.

The quarantine calculations restart immediately. So, if the scanning attack is continuous, quarantine is reactivated after the next 20 seconds.

Configuring Subscriber-Side Source-Based Rate Limit

This section contains the following:

Configuring Subscriber-Cable Modem Source-Based Rate Limit

Subscriber-cable modem SBRL identifies traffic streams by using the slot, MAC domain, and Service ID (SID) associated with the packet (that is, *slot/MD/SID*). All punts from this *slot/MD/SID* are aggregated and rate-limited as configured.

By default, subscriber-CM SBRL is disabled.

Before you begin

Restrictions

- All the punted packets are subject to CoPP and punt-policing.
- Layer 3 mobility punts are not subject to subscriber-cable modem SBRL. Layer 3 mobility punts are subject to the subscriber-MAC address SBRL.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**platform punt-sbrl subscriber rate** *rate* **Example:** `Router(config)# platform punt-sbrl subscriber rate 4`	Configures subscriber-cable modem rate in packets per second. The range is from 1 to 256, specified in powers-of-2.

Configuring Subscriber-MAC Address Source-Based Rate Limit

Subscriber-MAC address SBRL identifies traffic streams by hashing the punt cause and the source MAC address. This value is used as the index for rate-limiting. The Cisco cBR router does not perform special processing for hash collisions. So, the hash-colliding packet streams are rate-limited as if they are from the same packet stream.

The default rate for Layer 3 mobility punts is 4 packets per second.

Before you begin

Restrictions

- All the punted packets are subject to CoPP and punt-policing.
- Subscriber-MAC address SBRL applies only to subscriber-side Layer 3 mobility punts.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**platform punt-sbrl subscriber punt-cause** *punt-cause* **rate** *rate* **Example:** `Router(config)# platform punt-sbrl subscriber punt-cause 99 rate 2`	Configures subscriber-MAC address SBRL. • **punt-cause** *punt-cause*—Specifies the punt cause. The punt cause for Layer 3 mobility is 99. • **rate** *rate*—Specifies the rate limit in packets per second. The range is from 1 to 256, specified in powers-of-2.

Configuring Punt Policing

The punt policer aggregates all packets (both subscriber-side and WAN-side) with the specified punt cause, and rate-limits them according to the configured parameters.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**platform punt-policer** *punt-cause punt-rate* [**high**] **Example:** `Router(config)# platform punt-policer 1 10`	Configures punt policing. • *punt-cause*—Punt cause. The range is from 1 to 107. • *punt-rate*—Rate limit in packets per second. The range is from 10 to 146484. • **high**—(Optional) Specifies that the punt policing is performed only for high priority traffic.

Verifying the Source-Based Rate Limit Configuration

- **show running-config | include punt-sbrl**—Displays the SBRL configuration.

  Following is a sample output of the command:

```
Router# show running-config | include punt-sbrl

platform punt-sbrl wan punt-cause 11 rate 8
platform punt-sbrl wan punt-cause 24 rate 4
platform punt-sbrl subscriber rate 8
```

- **show access-lists** —Displays the access list information for verifying CoPP configuration.

  Following is a sample output of the command:

```
Router# show access-lists

Extended IP access list 120
    10 permit ip any any dscp af31
    20 permit ip any any dscp cs2
    30 permit ip any any dscp af21
    40 permit ip 68.86.0.0 0.1.255.255 any
IPv6 access list TRUSTEDV6
    permit ipv6 2001:558::/32 any sequence 10
```

- **show policy-map** *policy-map-name*—Displays the information for the policy map.

  Following is a sample output of the command:

```
Router# show policy-map copp_policy

  Policy Map copp_policy
    Class sbrl_trusted
     police rate 1000 pps
       conform-action transmit
       exceed-action transmit
    Class class-default
      set qos-group 99
```

- **show policy-map control-plane**—Displays the control plane policy map information.

  Following is a sample output of the command:

```
Router# show policy-map control-plane

 Control Plane

  Service-policy input: copp_policy

    Class-map: sbrl_trusted (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group 120
      Match: access-group name TRUSTEDV6
      police:
          rate 1000 pps, burst 244 packets
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
```

```
        transmit
      conformed 0 pps, exceeded 0 pps

  Class-map: class-default (match-any)
    28 packets, 4364 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: any
    QoS Set
      qos-group 99
        Marker statistics: Disabled
```

- **show platform hardware qfp active infrastructure punt sbrl**—Displays the SBRL statistics.

  Following is a sample output of the command:

```
Router# show platform hardware qfp active infrastructure punt sbrl

SBRL statistics

Subscriber CM
  drop-cnt  evict-cnt    SID  Interface
-------------------------------------------------
         1          1      5  Cable3/0/0
       982        982      5  Cable3/0/0

Subscriber MAC-addr
  nothing to report

WAN-IPv4
  drop-cnt  evict-cnt  quar  VRF  cause  IP-address
-----------------------------------------------------
    456788     456788     0    0    050  1.2.0.66

WAN-IPv6
  drop-cnt  evict-cnt  quar  VRF  cause  IP-address
-------------------------------------------------------------------------
    129334     129334     1    0    011  3046:1829:fefb::ddd1
       965        965     0    0    011  2001:420:2c7f:fc01::3
```

Note The value of *quar* is either 0 or 1. The value 1 indicates that quarantine is activated. The *quar* value is updated only when a packet from the source is dropped. If a source enters quarantine, and then stops sending packets, the *quar* value remains 1. However, the *drop-cnt* does not increment.

Note The SBRL statistics algorithm stores the data for the worst offenders. Sources that drop only a few packets are displayed in the table initially, but may be overwritten if the *drop-cnt* does not increase continuously. The *evict-cnt* increases in tandem with *drop-cnt*, but begins to decrease when a source is no longer being actively rate-limited. When the *evict-cnt* drops below 10, the record may be overwritten.

- **show platform hardware qfp active infrastructure punt statistics type global-drop**—Displays the global punt policer statistics.

  Following is a sample output of the command:

```
Router# show platform hardware qfp active infrastructure punt statistics type global-drop

Global Drop Statistics

  Number of global drop counters = 22

  Counter ID  Drop Counter Name                              Packets
  ------------------------------------------------------------------
  000         INVALID_COUNTER_SELECTED                      0
  001         INIT_PUNT_INVALID_PUNT_MODE                   0
  002         INIT_PUNT_INVALID_PUNT_CAUSE                  0
  003         INIT_PUNT_INVALID_INJECT_CAUSE                0
  004         INIT_PUNT_MISSING_FEATURE_HDR_CALLBACK        0
  005         INIT_PUNT_EXT_PATH_VECTOR_REQUIRED            0
  006         INIT_PUNT_EXT_PATH_VECTOR_NOT_SUPPORTED       0
  007         INIT_INJ_INVALID_INJECT_CAUSE                 0
  008         INIT_INJ_MISSING_FEATURE_HDR_CALLBACK         0
  009         PUNT_INVALID_PUNT_CAUSE                       0
  010         PUNT_INVALID_COMMON_HDR_VERSION               0
  011         PUNT_INVALID_PLATFORM_HDR_VERSION             0
  012         PUNT_PATH_NOT_INITIALIZED                     0
  013         PUNT_GPM_ALLOC_FAILURE                        0
  014         PUNT_TRANSITION_FAILURE                       0
  015         PUNT_DELAYED_PUNT_PKT_SB_NOT_IN_USE           0
  016         PUNT_CAUSE_GLOBAL_POLICER                     0
  017         INJ_INVALID_INJECT_CAUSE                      0
  018         INJ_INVALID_COMMON_HDR_VERSION                0
  019         INJ_INVALID_PLATFORM_HDR_VERSION              0
  020         INJ_INVALID_PAL_HDR_FORMAT                    0
  021         PUNT_GPM_TX_LEN_EXCEED                        0
```

- **show platform hardware qfp active infrastructure punt summary** [**threshold** *threshold-value*]—Displays the punt path rate-limiting summary.

  Following is a sample output of the command:

```
Router# show platform hardware qfp active infrastructure punt summary

Punt Path Rate-Limiting summary statistics

Subscriber-side
ID   punt cause           CPP punt        CoPP  ARPfilt/SBRL   per-cause      global
-------------------------------------------------------------------------------------
017  IPv6 Bad hop limit         22           0             0           0           0
050  IPv6 packet                13           0             0           0           0
080  CM not online             335           0             0           0           0

WAN-side
ID   punt cause           CPP punt        CoPP        SBRL   per-cause      global
-------------------------------------------------------------------------------------
017 IPv6 Bad hop limit         471           0           0           0           0
018 IPV6 Hop-by-hop Options  29901           0           0        1430           0
024 Glean adjacency            111           0           0           0           0
025 Mcast PIM signaling         19           0           0           0           0
050 IPv6 packet                 11           0           0           0           0
```

- **show platform software punt-policer**—Displays the punt policer configuration and statistics.

Following is a sample output of the command:

```
Router# show platform software punt-policer

Per Punt-Cause Policer Configuration and Packet Counters

Punt                              Configured (pps) Conform Packets   Dropped Packets
Cause Description                 Normal   High    Normal     High   Normal     High
------------------------------------------------------------------------------------
  2   IPv4 Options                4000     3000    0          0      0          0
  3   Layer2 control and legacy   40000    10000   16038      0      0          0
  4   PPP Control                 2000     1000    0          0      0          0
  5   CLNS IS-IS Control          2000     1000    0          0      0          0
  6   HDLC keepalives             2000     1000    0          0      0          0
  7   ARP request or response     2000     1000    0          49165  0          0
  8   Reverse ARP request or re... 2000    1000    0          0      0          0
  9   Frame-relay LMI Control     2000     1000    0          0      0          0
 10   Incomplete adjacency        2000     1000    0          0      0          0
 11   For-us data                 40000    5000    279977     0      0          0
 12   Mcast Directly Connected ... 2000    1000    0          0      0          0
 . . .
```

- **show platform hardware qfp active infrastructure punt policer summary**—Displays the punt policer summary.

  Following is a sample output of the command:

```
Router# show platform hardware qfp active infrastructure punt policer summary

QFP Punt Policer Config Summary


Policer  Rate     PeakRate   ConformBurst  ExceedBurst   Scaling
Handle   (pps)    (pps)      (pps)         (pps)         Factor
-------------------------------------------------------------------------
001      300000   0          2288          2288          0
002      4000     0          4000          0             0
003      3000     0          3000          0             0
004      40000    0          40000         0             0
005      10000    0          10000         0             0
006      2000     0          2000          0             0
007      1000     0          1000          0             0
008      2000     0          2000          0             0
009      1000     0          1000          0             0
010      2000     0          2000          0             0
011      1000     0          1000          0             0
012      2000     0          2000          0             0
013      1000     0          1000          0             0
014      2000     0          2000          0             0
 . . .
```

Configuration Example for Source-Based Rate Limit

Example: WAN-Side SBRL Configuration

```
access-list 120 permit ip any any dscp af31
access-list 120 permit ip any any dscp cs2
access-list 120 permit ip any any dscp af21
access-list 120 permit ip 192.168.1.10 0.1.255.255 any
```

```
ipv6 access-list TRUSTEDV6
  permit ipv6 any any dscp af31
  permit ipv6 any any dscp cs2
  permit ipv6 any any dscp af21
  permit ipv6 2001:558::/32 any


class-map match-all sbrl_trusted_v4
  match access-group 120

class-map match-all sbrl_trusted_v6
  match access-group name TRUSTEDV6


policy-map copp_policy
  ! IPv4 trusted:
  !   Specified rate is irrelevant.
  !   No special action; these packets bypass WAN-side SBRL.
  class sbrl_trusted_v4
    police rate 1000 pps conform transmit exceed transmit
  ! IPv6 trusted:
  !   Specified rate is irrelevant.
  !   No special action; these packets bypass WAN-side SBRL.
  class sbrl_trusted_v6
    police rate 1000 pps conform transmit exceed transmit

  ! add other classes here, if necessary

  ! Special action to activate WAN-side SBRL for this class.
  class class-default
    set qos-group 99


control-plane
  service-policy input copp_policy


! punt-cause 11 is FOR_US, punt-cause 24 is GLEAN_ADJ
platform punt-sbrl wan punt-cause 11 rate 4
platform punt-sbrl wan punt-cause 24 rate 4
```

Example: Subscriber-Side SBRL Configuration

```
platform punt-sbrl subscriber rate 4
```

Example: SBRL Configuration

```
…
platform punt-sbrl wan punt-cause 11 rate 4
platform punt-sbrl wan punt-cause 18 rate 16 quarantine-time 10 burst-factor 500
platform punt-sbrl wan punt-cause 24 rate 4
platform punt-sbrl subscriber rate 4
…
access-list 120 permit ip any any dscp af31
access-list 120 permit ip any any dscp cs2
access-list 120 permit ip any any dscp af21
access-list 120 permit ip 192.168.1.10 0.1.255.255 any
…
ipv6 access-list TRUSTEDV6
permit ipv6 any any dscp af31
```

```
permit ipv6 any any dscp cs2
permit ipv6 any any dscp af21
permit ipv6 2001:558::/32 any
…
policy-map copp_policy
class sbrl_trusted_v4
  police rate 1000 pps conform-action transmit  exceed-action transmit
 class sbrl_trusted_v6
  police rate 1000 pps conform-action transmit  exceed-action transmit
 class class-default
  set qos-group 99
…
control-plane
service-policy input copp_policy
…
```

Conversion of Divert Rate Limit Configuration on the Cisco uBR10012 Router to SBRL Configuration on the Cisco cBR Series Routers

Divert Rate Limit Configuration on the Cisco uBR10012 Router

The following is a sample Divert Rate Limit (DRL) configuration on the Cisco uBR10012 router:

```
service divert-rate-limit ip fib_rp_glean rate 4 limit 4
service divert-rate-limit ip fib_rp_dest rate 4 limit 4
service divert-rate-limit ip fib_rp_punt rate 4 limit 4
service divert-rate-limit ipv6 ipv6_rp_dest rate 4 limit 4
service divert-rate-limit ipv6 ipv6_rp_punt rate 4 limit 4
service divert-rate-limit ipv6 ipv6_rp_glean rate 4 limit 4
service divert-rate-limit ipv6 icmpv6 rate 4 limit 4

service divert-rate-limit trusted-site 0.0.0.0 0.0.0.0 tos 0x68 mask 0xFF
service divert-rate-limit trusted-site 0.0.0.0 0.0.0.0 tos 0x40 mask 0xFF
service divert-rate-limit trusted-site 68.86.0.0 255.254.0.0 tos 0x0 mask 0x0
service divert-rate-limit trusted-site 0.0.0.0 0.0.0.0 tos 0x48 mask 0xFF
service divert-rate-limit trusted-site-ipv6 ::/0 traffic-class 0x40 mask 0xFF
service divert-rate-limit trusted-site-ipv6 ::/0 traffic-class 0x48 mask 0xFF
service divert-rate-limit trusted-site-ipv6 ::/0 traffic-class 0x68 mask 0xFF
service divert-rate-limit trusted-site-ipv6 2001:558::/32 traffic-class 0x0 mask 0x0

interface Cablex/y/z
   cable divert-rate-limit rate 4 limit 30
```

In Cisco IOS Release 12.2(33)SCH2, the **divert-rate-limit max-rate wan** command was introduced on the Cisco uBR10012 router. This configuration limits the aggregate rate of diverted packets on the WAN-side, on a per-divert-code basis. The following is the recommended best-practice configuration for the **divert-rate-limit max-rate wan** command:

```
service divert-rate-limit max-rate wan fib_rp_glean rate 5000
service divert-rate-limit max-rate wan fib_rp_punt rate 5000
service divert-rate-limit max-rate wan fib_rp_dest rate 40000

service divert-rate-limit max-rate wan ipv6_fib_glean rate 5000
```

```
service divert-rate-limit max-rate wan ipv6_fib_punt rate 5000
service divert-rate-limit max-rate wan ipv6_fib_dest rate 40000
```

SBRL Configuration on the Cisco cBR Series Routers

The DRL functionality is called as Source-Based Rate Limit (SBRL) on the Cisco cBR Series Routers. The punt-path has three layers of protection:

- CoPP, on page 1200
- SBRL, on page 1201
- Punt Policer, on page 1201

CoPP

CoPP is used to specify the trusted sites and activate WAN-side SBRL. However, since CoPP applies to all punted packets, you must ensure that cable-side punts do not match the trusted sites.

The following is a sample CoPP configuration, which is equivalent to the configuration on the Cisco uBR10012 router:

```
access-list 120 permit ip any any dscp af31
access-list 120 permit ip any any dscp cs2
access-list 120 permit ip any any dscp af21
access-list 120 permit ip 68.86.0.0 0.1.255.255 any

ipv6 access-list TRUSTEDV6
   permit ipv6 any any dscp af31
   permit ipv6 any any dscp cs2
   permit ipv6 any any dscp af21
   permit ipv6 2001:558::/32 any

class-map match-all sbrl_trusted_v4
   match access-group 120

class-map match-all sbrl_trusted_v6
   match access-group name TRUSTEDV6

policy-map copp_policy
   class sbrl_trusted_v4
     police rate 1000 pps conform transmit exceed transmit
   class sbrl_trusted_v6
     police rate 1000 pps conform transmit exceed transmit
   class class-default
     set qos-group 99

control-plane
   service-policy input copp_policy
```

Note

- The **set qos-group 99** command activates SBRL for the specified class.
- The police rate for **sbrl_trusted_v***x* is irrelevant, as both actions are set to **transmit**.
- You can add other trusted sites, as necessary.

SBRL

The subscriber-side SBRL configuration is a single command in global configuration mode. The *limit* cannot be configured as the hardware policer is used. Therefore, we recommend that you configure a higher *rate* initially.

For WAN-side SBRL, the Cisco cBR Series routers do not have separate IPv4 and IPv6 configurations as the punt causes are shared between IPv4 and IPv6. The *limit* cannot be configured as the hardware policer is used. Therefore, we recommend that you configure a higher *rate* initially. In the following sample configuration, the punt cause 24 is for *Glean adjacency* and punt cause 11 is for *For-us data*, which are equivalent to *x*_rp_glean and *x*_rp_dest, respectively on the Cisco uBR10012 router.

```
platform punt-sbrl subscriber rate 16

platform punt-sbrl wan punt-cause 11 rate 8
platform punt-sbrl wan punt-cause 24 rate 8
```

Note

- The *fib-punt* punt cause is used in the Cisco uBR10012 router for packets destined to the management Ethernet. This punt cause is not used on the Cisco cBR Series routers.
- The Cisco cBR Series routers do not have an equivalent punt cause for ICMPV6. In the Cisco uBR10012 routers, ICMPv6 packets must be processed by the Route Processor to generate the checksum. In the Cisco cBR Series routers, ICMPv6 is processed in the control-plane. However, ICMPv6 punts can be identified and rate-limited (in aggregate) using CoPP.

Punt Policer

The punt policer operates on all punt causes and is fully configurable. The punt policer is not divided into WAN-side and subscriber-side. All packets with a given punt cause are aggregated and rate-limited as configured.

Following are the default settings (best-practice configuration) for the punt policer on the Cisco cBR Series routers:

```
punt-cause                      LO     HI

CPP_PUNT_CAUSE_GLEAN_ADJ        2000   5000
CPP_PUNT_CAUSE_FOR_US           40000  5000
```

Note

- The equivalent punt cause for *fib-glean* (on the Cisco uBR10012 router) is *GLEAN_ADJ/HI* on the Cisco cBR Series routers.
- The equivalent punt cause for *fib-dest* (on the Cisco uBR10012 router) is *FOR_US/LO* on the Cisco cBR Series routers.

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

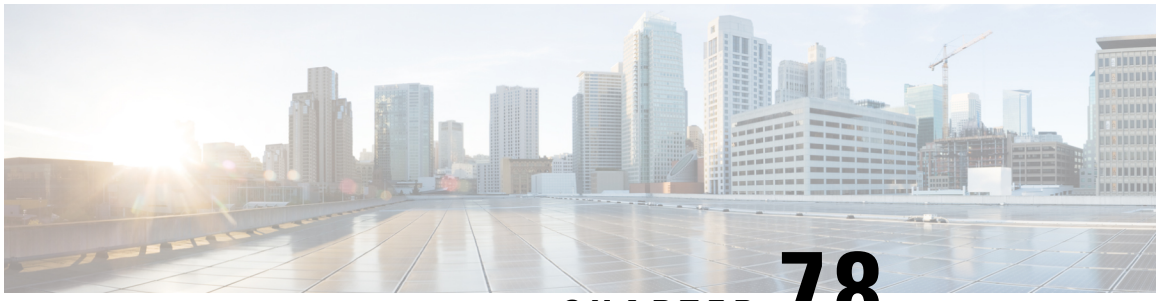
Feature Information for Source-Based Rate Limit

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 193: Feature Information for Source-Based Rate Limit

Feature Name	Releases	Feature Information
Source-based rate limit	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 78

Cable Duplicate MAC Address Reject

The Cable Duplicate MAC Address Reject feature is a DOCSIS 1.1-compliant security enhancement that helps to eliminate denial-of-service (DOS) attacks that are caused by cloned cable modems. A clone is presumed to be one of two physical cable modems on the same Cisco CMTS router with the same HFC interface MAC address. The cloned cable modem may be DOCSIS 1.0 or later, and may be semi-compliant or non-compliant with portions of the DOCSIS specifications.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1203
- Prerequisites for Cable Duplicate MAC Address Reject, on page 1204
- Restrictions for Cable Duplicate MAC Address Reject, on page 1205
- Information About Cable Duplicate MAC Address Reject, on page 1205
- How to Configure EAE and BPI+ Enforcement Features, on page 1208
- Configuration Example for EAE and BPI+ Enforcement Policies, on page 1210
- Verifying EAE and BPI+ Enforcement Policies, on page 1211
- System Messages Supporting Cable Duplicate MAC Address Reject, on page 1211
- Additional References, on page 1212
- Feature Information for Cable Duplicate MAC Address Reject, on page 1212

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 194: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Cable Duplicate MAC Address Reject

The Cable Duplicate MAC Address Reject feature entails the following behaviors and prerequisites on the DOCSIS-compliant network:

- The Cisco CMTS router requires that the legitimate cable modem is Baseline Privacy Interface Plus (BPI+) compliant, meaning that it can come to one of the following four online states when provisioned with a DOCSIS configuration file containing at least one BPI+ related type, length, value (TLV). For brevity, this document refers to these states as online(p_).
- The Cisco CMTS router gives priority to any cable modem that registers to the Cisco CMTS router in any of the following four states:
  - online(pt)
  - online(pk)
  - online(ptd)
  - online(pkd)

The Cisco CMTS router drops registration requests from another device that purports to use the same MAC address as an already operational modem that is in one of these four states.

Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 31 shows the hardware compatibility prerequisites for this feature.

Note The hardware components introduced in a given Cisco IOS Release are supported in all subsequent releases unless otherwise specified.

Restrictions for Cable Duplicate MAC Address Reject

- If the cable modem is not provisioned to use DOCSIS BPI+, as characterized by not coming online with the above initialization states of online(p_), then the existing behavior of the Cisco CMTS router remains unchanged. The Cisco CMTS router does not attempt to distinguish between two cable modems if the provisioning system does not provide a DOCSIS configuration file specifying BPI+ be enabled.
- When this feature is enabled, the Cisco CMTS router issues security breach notice in a log message in the cable logging layer2events log, or the generic log if the **cable logging layer2events** command is not configured on the Cisco CMTS router.

Information About Cable Duplicate MAC Address Reject

The Cable Duplicate MAC Address Reject feature is enabled by default on the Cisco CMTS router, and has no associated configuration commands. This feature creates a new log message, which appears in the system log by default.

This document also describes the following security features that are associated with the Cable Duplicate MAC Address Reject feature:

Early Authentication and Encryption

The Early Authentication and Encryption (EAE) feature enables the Cisco CMTS router to authenticate DOCSIS 3.0 cable modems immediately after completion of the ranging process, and encrypt all of the registration packets including DHCP and TFTP traffic. This security feature, compatible only with DOCSIS 3.0 cable modems, was introduced to help multiple service operators (MSOs) prevent theft of service.

This feature is enabled only for cable modems that initialize on a downstream channel on which the Cisco CMTS router is transmitting MAC Domain Descriptor (MDD) messages. The Cisco CMTS router uses TLV type 6 in the MDD MAC message to signal EAE to a cable modem. If this feature is enabled, only the authenticated cable modems are allowed to continue their initialization process and subsequently admitted to the network. The early authentication and encryption process involves the following:

- Authentication of the cable modem (that is the BPI+ authorization exchanges) after the ranging process.
- Traffic encryption key (TEK) exchanges for the cable modem primary Security Association Identifier (SAID).
- Encryption of IP provisioning traffic and Multipart Registration Request (REG-REQ-MP) messages during cable modem initialization.

Note When Early Authentication and Encryption is enabled, BPI will revert back to DES-56 even if the hardware supports AES-128.

EAE Enforcement Policies

The Cisco CMTS router supports the following EAE enforcement policies:

- No EAE enforcement (Policy 1)—EAE is disabled and the Cisco CMTS router cannot enforce EAE on any cable modem.
- Ranging-based EAE enforcement (Policy 2)—EAE is enforced on all DOCSIS 3.0 cable modems that range with a B-INIT-RNG-REQ MAC message.
- Capability-based EAE enforcement (Policy 3)—EAE is enforced on all DOCSIS 3.0 cable modems that range with a B-INIT-RNG-REQ MAC message in which the EAE capability flag is set using the .
- Total EAE enforcement (Policy 4)—EAE is enforced on all cable modems irrespective of the EAE capability flag status.

The EAE enforcement policies are mutually exclusive. By default, EAE is disabled on the Cisco CMTS router.

EAE Exclusion

You can exclude cable modems from EAE enforcement using the **cable privacy eae-exclude** command in the global configuration mode. Cable modems in the EAE exclusion list are always exempted from EAE enforcement. You can remove cable modems from the exclusion list using the no form of the **cable privacy eae-exclude** command.

BPI+ Security and Cloned Cable Modems

The BPI+ Security and Cloned Cable Modems feature prioritizes cable modems that are online with BPI+ security over new cable modem registration requests that use the same cable modem MAC address. As a result, the legitimate cable modem with BPI+ security certificates that match the HFC MAC address does not experience service disruption, even if a non-compliant cable modem with the same HFC MAC address attempt to register.

The cloned cable modem detection function requires that a cable modem use DOCSIS 1.1 or a later version and should be provisioned with BPI+ enabled. That is, one BPI+ type, length, value (TLV) must be included in the DOCSIS configuration file. All DOCSIS 1.0, DOCSIS 1.1, and later cable modems that are provisioned without DOCSIS BPI+ enabled continue to use the legacy DOCSIS behavior, and experience a DoS attack when a cloned cable modem appears on the Cisco CMTS router.

This cloned cable modem detection function mandates that a cable modem provisioned with BPI+ and DOCSIS 1.1 QoS must register with BPI+ and not use BPI. The commonly available non-DOCSIS-compliant cable modems contain an option to force registration in BPI as opposed to BPI+ mode even when DOCSIS 1.1 QoS and BPI+ are specified in the DOCSIS configuration file.

Logging of Cloned Cable Modems

Cloned cable modems are detected and tracked with system logging. The Logging of Cloned Cable Modem feature is enabled by default. Due to the large number of DOCSIS Layer 2 messages typically seen in a production network, a separate log is available to segregate these messages. By default, cloned cable modem messages are placed in the cable logger, cable layer2events logging. If you disable this feature using the no form of the **cable logging layer2events** command in global configuration mode, then the cloned cable modem messages are placed in the system log (syslog).

A cloned cable modem might attempt dozens of registration attempts in a short period of time. In order to suppress the number of log messages generated, the Cisco CMTS router suppresses clone detected messages for approximately 3 minutes under certain conditions.

The log message provides the cable interface and MAC address of the cable modem attempting to register when another physical modem with that same MAC address is already in a state of online(p_) elsewhere on the Cisco CMTS router.

DOCSIS 3.0 BPI+ Policy Enforcement

The DOCSIS 3.0 BPI+ Policy Enforcement feature was introduced to prevent cable modem MAC address cloning and theft of service. This feature enables a Cisco CMTS router to validate the MAC address of each cable modem. To enforce BPI+ on cable modems, you must configure one of the following enforcement policies per MAC domain on the router:

- 1.1 Style Configuration File Parameters and Capability (Policy 1)—The Cisco CMTS router enforces BPI+ on cable modems that register with a DOCSIS 1.1 configuration file with parameters indicating BPI+ is enabled with or without TLV 29. To configure this policy, the privacy support modem capability TLV (type 5.6) in the DOCSIS configuration file must be set to BPI+ support. This policy forces BPI+ on a cable modem that is BPI+ capable and provisioned with DOCSIS1.1 configuration file. A cable modem that signals these capabilities during registration is blocked from accessing the network until the modem completes BPI+ negotiation.
- 1.1 Style Configuration File Parameters (Policy 2)—The Cisco CMTS router enforces BPI+ on cable modems that register with a DOCSIS 1.1 configuration file with parameters indicating BPI+ is enabled with or without TLV 29. A cable modem that registers with this type of configuration file is blocked from accessing the network until the modem completes BPI+ negotiation.
- 1.1 Style Configuration File (Policy 3)—The Cisco CMTS router enforces BPI+ on cable modems that register with a DOCSIS 1.1 configuration file. This means that if you provision a DOCSIS 1.1 configuration file with security disabled (privacy flag is not present in the configuration file), all DOCSIS 1.1 and 2.0 cable modems are blocked from accessing the network. Only the DOCSIS 3.0 cable modems that have security enabled implicitly will pass this check if the privacy flag is not present in the configuration file.
- Total enforcement (Policy 4)—The Cisco CMTS router enforces BPI+ on all cable modems. This means that all cable modems that do not run BPI+ are blocked from accessing the network.

Note You can configure only one enforcement policy at a time per MAC domain. If you configure one policy after another, the latest policy supersedes the already existing policy. For example, if you want Policy 2 to take over Policy 1, you can directly configure the former without disabling the latter.

These enforcement policies are implemented based on CableLabs Security Specification, CM-SP-SECv3.0-I13-100611. You can configure these enforcement policies using the **cable privacy bpi-plus-policy** command in cable interface configuration mode. The cable modems that do not comply with the configured policy can still come online but they cannot access the DOCSIS network and some dual stack cable modems may not get both the IPv4 and IPv6 addresses.

Policies 1, 2, and 3 support a mixed network of DOCSIS 1.0 (including DOCSIS Set-top Gateway), DOCSIS 1.1, and later cable modems. Policy 4 is the most effective configuration for preventing cable modem MAC address cloning as this policy enforces BPI+ on all cable modems. Policy 4 blocks all DOCSIS 1.0 cable modems as they do not register in BPI+ mode. Therefore, if Policy 4 is used, you must upgrade all authorized DOCSIS 1.0 cable modems or remove them from the network.

BPI+ Policy Enforcement Exclusion

You can exclude cable modems (DOCSIS 1.0 and later versions) from BPI+ policy enforcement based on their MAC addresses, using the **cable privacy bpi-plus-exclude** command in global configuration mode. You can exclude a maximum of 30 cable modems per MAC domain.

How to Configure EAE and BPI+ Enforcement Features

This section provides information on how to configure the following BPI+ enforcement features:

Configuring EAE Enforcement Policies

By default, EAE is disabled on the Cisco CMTS router. You can configure EAE enforcement policies using the **cable privacy eae-policy** command in cable interface configuration mode.

Note EAE enforcement policies are enabled only for the DOCSIS 3.0 cable modems that initialize on a downstream channel.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** {*slot*/*cable-interface-index* | *slot*/*subslot*/*cable-interface-index*} **Example:** `Router(config)# interface cable 6/0/1`	Enters interface configuration mode.
Step 4	**cable privacy eae-policy** {**capability-enforcement** | **disable-enforcement** | **ranging-enforcement** | **total-enforcement**} **Example:** `Router(config-if)# cable privacy eae-policy total-enforcement`	Specifies EAE enforcement policies on DOCSIS 3.0 cable modems.

	Command or Action	Purpose
Step 5	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Configuring BPI+ Enforcement Policies

The BPI+ enforcement policies are configured per MAC domain to prevent cable modem MAC address cloning and theft of service.

Before you begin

The customer premise equipment (CPE) must use DHCP to acquire IP addresses to access the network, or the statically assigned IP addresses must be managed appropriately.

Note Only a single enforcement policy can be applied per MAC domain.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** *{slot/subslot/port |slot/port}* **Example:** `Router(config)# interface cable 5/1/0`	Specifies the cable interface line card on a Cisco CMTS router.
Step 4	**cable privacy bpi-plus-policy {capable-enforcement | d11-enabled-enforcement | d11-enforcement | total-enforcement}** **Example:**	Specifies the BPI+ enforcement policies per MAC domain.

	Command or Action	Purpose
	`Router (config-if)# cable privacy bpi-plus-policy total-enforcement`	
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to Privileged EXEC mode.

Configuring AES-128 for non-MTC DOCSIS3.0 Cable Modem

This feature is enabled by default. To disable this feature, follow the steps below:

```
enable
configure terminal
no cable privacy non-mtc-aes128
end
```

Verifying AES-128 for non-MTC DOCSIS3.0 Cable Modem

To verify whether AES-128 is supported for non-MTC DOCSIS3.0 Cable Modem, use **show running-config** command as shown in the example below:

```
Router# show running-config | include cable privacy non-mtc-aes128
no cable privacy non-mtc-aes128
```

Troubleshooting Tips

Use the following debug commands to troubleshoot BPI+ policy enforcement configuration:

- **debug cable mac-address**—Provides debugging information about a specific cable modem.
- **debug cable bpiatp**—Enables debugging of the BPI handler.

Configuration Example for EAE and BPI+ Enforcement Policies

The following example shows how to configure an EAE enforcement policy on the Cisco cBR-8 router:

```
Router# configure terminal
Router(config)# interface cable 8/1/0
Router (config-if)# cable privacy eae-policy capability-enforcement
Router (config-if)# cable privacy eae-policy ranging-enforcement
Router (config-if)# cable privacy eae-policy total-enforcement
```

The following example shows how to configure a BPI+ enforcement policy at slot/subslot/port 5/1/0 on the Cisco cBR-8 router:

```
Router# configure terminal
Router(config)# interface cable 5/1/0
Router (config-if)# cable privacy bpi-plus-policy total-enforcement
```

Verifying EAE and BPI+ Enforcement Policies

Use the following show commands to verify EAE and BPI+ enforcement configurations:

- **show interface cable privacy**
- **show cable privacy**
- **show cable modem access-group**

To verify which EAE policy is configured on the Cisco CMTS router, use the **show interface cable privacy** command.

To verify which cable modems are excluded from EAE enforcement on the Cisco CMTS router, use the **show cable privacy** command.

To verify BPI+ enforcement policies, use the **show interface cable privacy** command.

Note A character "*" is placed before the online state to identify modems that have not satisfied the bpi-plus-policy.

What to Do Next

The Cloned Cable Modem Detection feature relates to multiple BPI+ certificate and DOCSIS 1.1 factors.

System Messages Supporting Cable Duplicate MAC Address Reject

The following example illustrates logged events for the Cloned Cable Modem Detection feature on a Cisco cBR-8 router.

In the below scenario, there are two cable modems with MAC addresses that have been cloned:

- For MAC address 000f.66f9.48b1, the legitimate cable modem is on C5/0/0 upstream 0, and the cloned cable modem is on C7/0/0.
- For MAC address 0013.7116.e726, the legitimate cable modem is on C7/0/0 upstream 0, and the cloned cable modem is also on the same interface.
- In the below example, the CMMOVED message occurred because the cloned cable modem for MAC address 000f.66f9.48b1 came online before the legitimate cable modem.
- There is no CMMOVED message for the cable modem on interface C7/0/0 with MAC address 0013.7116.e726 because the legitimate cable modem came online with state of online(pt) before the cloned cable modem attempted to come online.

```
Dec 5 13:08:18: %CBR-6-CMMOVED: Cable modem 000f.66f9.48b1 has been moved from interface
Cable7/0/0 to interface C able5/0/0.
```

```
Dec 5 13:08:44: %CBR-5-CLONED_CM_DETECTED: Cloned CM with MAC address 0013.7116.e726
connection attempt rejected o n Cable7/0/0 U0
Dec 5 13:10:48: %CBR-5-CLONED_CM_DETECTED: Cloned CM with MAC address 000f.66f9.48b1
connection attempt rejected on Cable7/0/0 U1
Dec 5 13:12:37: %CBR-5-CLONED_CM_DETECTED: Cloned CM with MAC address 0013.7116.e726
connection attempt rejected on Cable7/0/0 U0
Dec 5 13:18:28: %CBR-5-CLONED_CM_DETECTED: Cloned CM with MAC address 0013.7116.e726
connection attempt rejected on Cable7/0/0 U0
Dec  5 13:18:28: %CBR-5-CLONED_CM_DETECTED: Cloned CM with MAC address 0013.7116.e726
connection attempt rejected o n Cable7/0/0 U0
```

The following example of the **show cable modem** command illustrates additional cable modem information for the above scenario involving the specified MAC addresses:

```
Router# show cable modem 000f.66f9.48b1
MAC Address    IP Address      I/F       MAC          Prim RxPwr  Timing Num BPI
                                          State         Sid  (dBmv) Offset CPE Enb
000f.66f9.48b1 4.222.0.253     C5/0/0/U0 online(pt)    24    0.50  1045    1   Y
```

Additional References

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

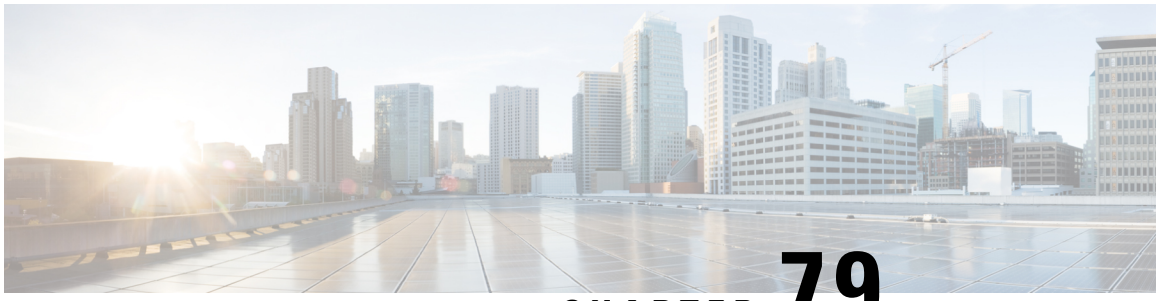
Feature Information for Cable Duplicate MAC Address Reject

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 195: Feature Information for Cable Duplicate MAC Address Reject

Feature Name	Releases	Feature Information
Cable Duplicate MAC Address Reject	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.
AES-128 for non-MTC DOCSIS 3.0 Cable Modem	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 79

Cable ARP Filtering

This document describes the Cable ARP Filtering feature for the Cisco Cable Modem Termination System (CMTS). This feature enables service providers to filter Address Resolution Protocol (ARP) request and reply packets, to prevent a large volume of such packets from interfering with the other traffic on the cable network.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1215
- Prerequisites for Cable ARP Filtering, on page 1216
- Restrictions for Cable ARP Filtering, on page 1216
- Information About Cable ARP Filtering, on page 1217
- How to Configure Cable ARP Filtering, on page 1219
- Configuration Examples for Cable ARP Filtering, on page 1226
- Additional References, on page 1229
- Feature Information for Cable ARP Filtering, on page 1229

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 196: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Cable ARP Filtering

No specialequipment or softwareis needed to use the Cable ARP Filtering feature.

Restrictions for Cable ARP Filtering

Cisco cBR-8 Router Restrictions

- The Cisco cBR-8 router maintains ARP filtering statistics on the Supervisor (SUP) module. Statistics are viewed with the **show cable arp-filter** command for a specified interface. When a switchover event occurs, as in SUP Redundancy, these ARP filtering statistics are reset to zero.
- The Cable ARP Filter feature is not configurable per subinterface.

FP ARP Filter Restrictions

- The FP microcode must be enhanced to provide the rate limiting functionality for ARP filtering in FP.
- The ARP Filter in FP feature is not configurable per subinterface.

Information About Cable ARP Filtering

Overview

Theft-of-service and denial-of-service (DNS) attacks have become increasingly common in cable broadband networks. In addition, virus attacks are becoming more common, and users are often unaware that their computers have become infected and are being used to continue the attacks on the network.

One sign that often appears during these attacks is an unusually high volume of Address Resolution Protocol (ARP) packets. The user or virus repeatedly issues ARP requests, trying to find the IP addresses of additional computers that might be vulnerable to attack.

ARP requests are broadcast packets, so they are broadcast to all devices on that particular network segment. In some cases, a router can also forward ARP broadcasts to an ARP proxy for further processing.

This problem is also made worse because some low-end routers commonly used by subscribers for home networks can also incorrectly respond to all ARP requests, which generates even more traffic. Until these customer premises equipment (CPE) devices can be upgraded with firmware that is compliant to the appropriate Request for Comments (RFC) specifications, service providers need to be able to deal with the incorrectly generated or forwarded traffic.

In addition, the Cisco CMTS router automatically monitors ARP traffic and enters the IP addresses found in ARP requests into its own ARP table, in the expectation that a device will eventually be found with that IP address. Unacknowledged IP addresses remain in the router's ARP table for 60 seconds, which means that a large volume of ARP traffic can fill the router's ARP table.

This process can create a large volume of ARP traffic across the network. In some situations, the volume of ARP requests and replies can become so great that it can throttle other traffic and occupy most of the Cisco CMTS router's processing time, hampering efforts by technicians to recover their network.

The router cannot use fast-switching to process ARP packets, but must instead forward them to the route processor (RP). Because of this, processing a large volume of ARP traffic can also prevent the router from handling normal traffic.

Filtering ARP Traffic

To control the volume of ARP traffic on a cable interface, you can configure the **cable arp filter** command to specify how many ARP packets are allowed per Service ID (SID) during a user-specified time period. You can configure separate thresholds for ARP request packets and for ARP reply packets.

When a cable interface is configured to filter ARP packets, it maintains a table of the number of ARP request or reply packets that have been received for each SID. If a SID exceeds the maximum number of packets during the window time period, the Cisco CMTS drops the packets until a new time period begins.

Note If using bundled cable interfaces, the Cable ARP Filtering feature is configured on the primary and subordinate interfaces separately. This allows you to configure the feature only on the particular interfaces that require it. In addition, you can configure the feature with different threshold values, allowing you to customize the feature for each interface's traffic patterns.

Monitoring Filtered ARP Traffic

After ARP filtering has been enabled on a cable interface, you can then use the service **divert-rate-limit** command to display the devices that are generating excessive amounts of ARP traffic. These devices could be generating this traffic for any of the following reasons:

- Cable modems that are running software images that are either not DOCSIS-compliant or that have been hacked to allow theft-of-service attacks.
- CPE devices that are either performing a theft-of-service or denial-of-service attack, or that have been infected with a virus that is searching for other computers that can be infected.
- Routers or other devices that mistakenly reply to or forward all ARP requests.

After identifying the specific devices that are generating this traffic, you can use whatever techniques are allowed by your service level agreements (SLAs) to correct the problem.

Linksys Wireless-Broadband Router (BEFW11S4)

The Linksys Wireless-B Broadband Router, Model number BEFW11S4 version 4 with 1.44.2 firmware, incorrectly sends its own ARP reply packet for every ARP request packet it receives, instead of replying only to the ARP requests that are specifically for itself. Customers with these routers should upgrade the firmware to the latest revision to fix this bug. To upgrade the firmware, go to the download section on the Linksys website.

Note It is extremely important that non-compliant CPE devices be updated to firmware that correctly handles ARP and other broadcast traffic. Even one or two non-compliant devices on a segment can create a significant problem with dropped packets, impacting all of the other customers on that segment.

ARP Filtering in FP

ARP filter feature is performed on SUP FP complex. When enabled, this FP complex filters ARP packets for identified ARP offenders, decreasing the ARP punt rate and RP CPU usage. It also provides the user with clearer separation in ARP filtering by utilizing source MAC addresses instead of SIDs.

The filter logic now filters by source MAC address instead of by SID. Currently, the modem MAC addresses are excluded from having their ARPs filtered, but Multimedia Terminal Adapters (MTAs) and other non-offending CPEs can still (statistically) have ARPs filtered because all ARPs appear to come from the same SID. Therefore, filtering by source MAC address will isolate the filtering to the offensive devices. By doing so, a customer who has Voice-over-IP (VoIP) service via an MTA and an infected CPE will not have MTA issues while being contacted by the service provider in regards to the infected CPE.

ARP offenders will still be allowed to use ARP to avoid complete loss of Internet connectivity through their configured or provisioned gateway address. Because of this, it is expected that the “ARP Input” process will still show a few percentage points of CPU usage, but the net interrupt CPU usage will decrease.

Note ARP filtering in FP is enabled by default on Cisco cBR-8 router.

Filtering ARP Traffic in FP

When ARP traffic in FP is enabled, a lightweight algorithm executing on the RP is used to identify ARP offenders by the source MAC address or the SID. All offending source MAC addresses or SIDs are then programmed by the ARP Filter control module into the FP ucode divert rate limiting module (ARP offenders are still allowed to perform ARP transactions, but only at the configured filtering rate).

Offending source MAC addresses or SIDs are filtered in FP for a minimum of 50 minutes (ten 5-minute intervals with no occurring offenses). Utilizing the existing ARP Filter CLI tools, the cable operator can obtain enough information about the modem and CPE to contact the end user to request the necessary anti-virus software installation or firmware upgrade for the CPE.

Note If the offending device is not “repaired” or shut off, it will remain in the FP ARP Filter indefinitely.

The FP ARP rate limiter is designed to filter a maximum of 16,000 ARP offenders. If this pool of 16,000 filterable entities is exhausted, then the entity is filtered on the RP. The CLI statistics will distinguish mac addresses filtered on the RP verses FP.

Because of possible mac address hash collisions, ARP offenders that cannot be programmed into the FP ARP rate limiter will still be filtered in FP by SID. Since the hash is done by source mac address and SID, such devices can actually moved back to mac address filtering by deleting the associated modem and forcing it back online with a new SID (this merely a possibility and is not expected to be a common practice).

ARP packets with a source mac address that is not “known” to the CMTS as a modem or CPE will be filtered by their SID in FP. Therefore, there will never be an unusual ARP packet source that will NOT be filtered in FP. False ARP packets with invalid operation codes will be filtered as if they are an ARP Reply.

How to Configure Cable ARP Filtering

Use the following procedures to determine whether ARP filtering is required and to configure ARP filtering on one or more cable interfaces.

Monitoring ARP Processing

Use the following steps to monitor how the router is processing ARP traffic and whether the volume of ARP packets is a potential problem.

Procedure

Step 1 To discover the CPU processes that are running most often, use the **show process cpu sorted** command and look for the ARP Input process:

Example:

```
Router# show process cpu sorted

CPU utilization for five seconds: 99%/28%; one minute: 93%; five minutes: 90%
 PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min TTY Process
  19   139857888  44879804       3116 31.44% 28.84% 28.47%   0 ARP Input
```

```
 154    74300964   49856254       1490 20.29% 19.46% 15.78%   0 SNMP ENGINE
  91    70251936    1070352      65635  8.92%  9.62%  9.59%   0 CEF process
  56    17413012   97415887        178  3.01%  3.67%  3.28%   0 C10K BPE IP Enqu
  78    24985008   44343708        563  3.68%  3.47%  3.24%   0 IP Input
  54     6075792    6577800        923  0.90%  0.67%  0.65%   0 CMTS SID mgmt ta
...
```

In this example, the ARP Input process has used 31.44 percent of the CPU for the past five seconds. Total CPU utilization is also at 99 percent, indicating that a major problem exists on the router.

Note As a general rule, the ARP Input process should use no more than one percent of CPU processing time during normal operations. The ARP Input process could use more processing time during certain situations, such as when thousands of cable modems are registering at the same time, but if it uses more than one percent of processing time during normal operations, it probably indicates a problem.

Step 2 To monitor only the ARP processes, use the **show process cpu | include ARP** command:

Example:

```
Router# show process cpu | include ARP

  19   139857888   44879804       3116 31.44% 28.84% 28.47%   0 ARP Input
 110           0          1          0  0.00%  0.00%  0.00%   0 RARP Input
```

Step 3 To monitor the number of ARP packets being processed, use the **show ip traffic** command.

Example:

```
Router# show ip traffic | begin ARP

ARP statistics:
  Rcvd: 11241074 requests, 390880354 replies, 0 reverse, 0 other
  Sent: 22075062 requests, 10047583 replies (2127731 proxy), 0 reverse
```

Repeat this command to see how rapidly the ARP traffic increases.

Step 4 If ARP traffic appears to be excessive, use the **show cable arp-filter** command to display ARP traffic for each cable interface, to identify the interfaces that are generating the majority of the traffic.

Example:

```
Router# show cable arp-filter Cable5/0/0

ARP Filter statistics for Cable5/0/0:
  Rcvd Replies: 177387 total, 0 unfiltered, 0 filtered
  Sent Requests For IP: 68625 total, 0 unfiltered, 0 filtered
  Sent Requests Proxied: 7969175 total, 0 unfiltered, 0 filtered
```

In the above example, the unfiltered and filtered counters show zero, which indicates that ARP filtering has not been enabled on the cable interface. After ARP filtering has been enabled with the **cable arp filter** command, you can identify the specific devices that are generating excessive ARP traffic by using the **service divert-rate-limit** command (see the Identifying the Sources of Major ARP Traffic, on page 1222).

Enabling ARP Filtering

Use the following procedure to enable ARP filtering on a particular cable interface.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** *x/y* **Example:** `Router(config)# interface cable 5/1`	Enters interface configuration mode for the specified cable interface.
Step 4	**cable arp filter reply-accept** *number window-size* **Example:** `Router(config-if)# cable arp filter reply-accept 2 2`	Configures the cable interface to accept only the specified *number* of ARP reply packets every *window-size* seconds for each active Service ID (SID) on that interface. The cable interface drops ARP reply packets for a SID that would exceed this number. (The default behavior is to accept all ARP reply packets.)
Step 5	**cable arp filter request-send** *number window-size* **Example:** `Router(config-if)# cable arp filter request-send 3 1`	Configures the cable interface to send only the specified *number* of ARP request packets every *window-size* seconds for each active SID on that interface. The cable interface drops ARP requests for a SID that would exceed this number. (The default behavior is to send all ARP request packets.) **Note** Repeat Step 3 through Step 5 to enable ARP filtering on other cable interfaces. Primary and subordinate interfaces in a cable bundle must be configured separately.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose

Identifying the Sources of Major ARP Traffic

After you have begun filtering ARP traffic on a cable interface, use the following procedure to identify the cable modems or CPE devices that are generating or forwarding major amounts of ARP traffic.

Tip The Linksys Wireless-B Broadband Router, Model number BEFW11S4 version 4 with 1.44.2 firmware, has a known problem in which it incorrectly generates an ARP reply for every ARP request packet it receives. See the Linksys Wireless-Broadband Router (BEFW11S4) guide for information on how to resolve this problem.

Procedure

Step 1 To discover the devices that are responsible for generating or forwarding more ARP requests on a specific cable interface than a specified minimum number of packets, use the **show cable arp-filter requests-filtered** command where *number* is the threshold value for the number of packets being generated:

Example:

```
show cable arp-filter cable interface requests-filtered number
```

For example, to display the devices that have generated more than 100 ARP request packets, enter the following command:

Example:

```
Router# show cable arp-filter cable 5/1/0 requests-filtered 100

Sid  MAC Address    IP Address      Req-Filtered   Req-For-IP-Filtered  Rep-Filtered
1    0006.2854.72d7 10.3.81.4       12407          0                    0
81   00C0.c726.6b14 10.3.81.31      743            0                    0
```

Step 2 Repeat the **show cable arp-filter** command to show how quickly the devices are generating the ARP packets.

Step 3 To discover the devices that are responsible for generating or forwarding more ARP replies on a specific cable interface than a specified minimum number of packets, use the **show cable arp-filter replies-filtered** command where *number* is the threshold value for the number of packets being generated:

Example:

```
show cable arp-filter cable interface requests-filtered number
```

For example, to display the devices that have generated more than 200 ARP reply packets, enter the following command:

Example:

```
Router# show cable arp-filter cable 5/0/0 replies-filtered 200

Sid  MAC Address    IP Address      Req-Filtered   Req-For-IP-Filtered  Rep-Filtered
2    0006.53b6.562f 10.11.81.16      0             0                    2358
191  0100.f31c.990a 10.11.81.6       0             0                    11290
```

Step 4 (Optional) If a particular cable modem is generating or forwarding excessive ARP replies, contact the customer to see if they are using a Linksys Wireless-B Broadband Router, Model number BEFW11S4. If so, this router could be running old firmware that is incorrectly generating excessive ARP packets, and the customer should upgrade their firmware. For more information, see the Linksys Wireless-Broadband Router (BEFW11S4) guide

Step 5 Repeat this command during each filter period (the time period you entered with the **cable arp filter** command) to show how quickly the devices are generating the ARP packets.

Step 6 (Optional) The ARP reply and request packet counters are 16-bit counters, so if a very large number of packets are being generated on an interface, these counters could wrap around to zero in a few hours or even a few minutes. Clearing the ARP counters eliminates stale information from the display and makes it easier to see the worst offenders when you suspect ARP traffic is currently creating a problem on the network.

To eliminate the modems that are not currently triggering the ARP filters and to isolate the worst current offenders, use the **clear counters cable** *interface* command to reset all of the interface counters to zero. Then the **show cable arp-filter** commands clearly identify the SIDs of the modems that are currently forwarding the most ARP traffic.

For example, the following example indicates that a number of modems are forwarding a large enough volume of ARP traffic that they have triggered the ARP packet filters:

Example:

```
Router# show cable arp-filter cable 5/1/0 requests-filtered 10

Sid  MAC Address    IP Address      Req-Filtered   Req-For-IP-Filtered  Rep-Filtered
1    0006.2854.72d7 10.3.81.4       8              0                    0
23   0007.0e02.b747 10.3.81.31      32             0                    0
57   0007.0e03.2c51 10.3.81.31      12407          0                    0
...
81   00C0.c726.6b14 10.3.81.31      23             0                    0
```

SID 57 shows the largest number of packets, but it is not immediately apparent if this modem is causing the current problems. After clearing the counters though, the worst offenders are easily seen:

Example:

```
Router# clear counter cable 5/1/0

Clear show interface counters on this interface [confirm] y

08:17:53.968: %CLEAR-5-COUNTERS: Clear counter on interface Cable5/1/0 by console
Router# show cable arp cable 5/1/0

ARP Filter statistics for Cable3/0:
  Replies Rcvd: 0 total. 0 unfiltered, 0 filtered
  Requests Sent For IP: 0 total. 0 unfiltered, 0 filtered
  Requests Forwarded: 0 total. 0 unfiltered, 0 filtered

Router# show cable arp-filter cable 5/1/0 requests-filtered 10
```

```
Sid  MAC Address    IP Address      Req-Filtered   Req-For-IP-Filtered  Rep-Filtered
Router# show cable arp-filter cable 5/1/0 requests-filtered 10

Sid  MAC Address    IP Address      Req-Filtered   Req-For-IP-Filtered  Rep-Filtered
57   0007.0e03.2c51 10.3.81.31      20             0                    0
81   00C0.c726.6b14 10.3.81.31      12             0                    0
Router# show cable arp-filter cable 5/1/0 requests-filtered 10

Sid  MAC Address    IP Address      Req-Filtered   Req-For-IP-Filtered  Rep-Filtered
57   0007.0e03.2c51 10.3.81.31      31             0                    0
81   00C0.c726.6b14 10.3.81.31      18             0                    0
```

Step 7 (Optional) If the Req-For-IP-Filtered column shows the majority of ARP packets, use the **show cable arp-filter ip-requests-filtered** command to display more details about the CPE device that is generating this traffic. Then use the **debug cable mac-address** and **debug cable arp filter** commands to display detailed information about this particular traffic; for example:

Example:

```
Router# show cable arp-filter c5/0/0 ip-requests-filtered 100

Sid  MAC Address    IP Address      Req-Filtered Req-For-IP-Filtered Rep-Filtered
1    0007.0e03.1f59 50.3.81.3       0            37282               0
Router# debug cable mac-address 0007.0e03.1f59

Router# debug cable arp filter

Router#
Apr 23 23:03:23.565: ARP for IP Filter=F sid 1 s 0000.0000.0049 d 0005.00e5.3610 sip
50.3.81.13 dip 50.3.82.173 prot 6 len 46 SrcP 445 DstP 445
Apr 23 23:03:23.565: ARP for IP Filter=F sid 1 s 0000.0000.0049 d 0005.00e5.3610 sip
50.3.81.13 dip 50.3.82.174 prot 6 len 46 SrcP 445 DstP 445
Apr 23 23:03:23.565: ARP for IP Filter=F sid 1 s 0000.0000.0049 d 0005.00e5.3610 sip
50.3.81.13 dip 50.3.82.175 prot 6 len 46 SrcP 445 DstP 445
[additional output omitted]...
```

This example shows that the CPE device at IP address 50.3.81.13 is sending packets to TCP port 445 to every IP address on the 50.3.82.0 subnet, in a possible attempt to find a computer that has Microsoft Windows file-sharing enabled.

Step 8 After determining the specific devices that are generating excessive ARP traffic, you can take whatever action is allowed by your company's service level agreements (SLAs) to correct the problem.

Examples

In this example, two cable interfaces, C5/0/0 and C7/0/0, are joined in the same bundle, which means the interfaces share the same broadcast traffic. Separate devices on each interface are generating excessive ARP traffic:

- The device at MAC address 000C.2854.72D7 on interface C7/0/0 is generating or forwarding a large volume of ARP requests. Typically, this device is a cable modem that is forwarding the ARP requests that are being generated by a CPE device behind the modem. The CPE device

could be attempting a theft-of-service or denial-of-service attack, or it could be a computer that has been infected by a virus and is trying to locate other computers that can be infected.

- The device at MAC address 000C.53B6.562F on Cable 5/0/0 is responding to a large number of ARP requests, which could indicate that the device is a router that is running faulty software.

The following commands identify the device on the C7/0/0 interface that is generating the excessive ARP requests:

```
Router# show cable arp-filter c7/0/0

ARP Filter statistics for Cable7/0/0:
  Replies Rcvd: 3 total. 3 unfiltered, 0 filtered
  Requests Sent For IP: 0 total. 0 unfiltered, 0 filtered
  Requests Forwarded: 27906 total. 562 unfiltered, 27344 filtered
Router# show cable arp-filter c7/0/0 requests-filtered 100

Sid  MAC Address    IP Address      Req-Filtered   Req-For-IP-Filtered  Rep-Filtered
1    000C.2854.72d7 50.3.81.4       62974          0                    0
```

The following commands identify the device on the C5/0/0 interface that is generating the excessive ARP replies:

```
Router# show cable arp-filter c5/0/0

ARP Filter statistics for Cable5/0/0:
  Replies Rcvd: 2400 total. 456 unfiltered, 1944 filtered
  Requests Sent For IP: 0 total. 0 unfiltered, 0 filtered
  Requests Forwarded: 26 total. 26 unfiltered, 0 filtered
Router# show cable arp-filter c5/0/0 replies-filtered 100

Sid  MAC Address    IP Address      Req-Filtered   Req-For-IP-Filtered  Rep-Filtered
2    000C.53b6.562f 50.3.81.6       0              0                    2097
```

Clearing the Packet Counters

To clear the packet counters on an interface, which includes the ARP packet counters, use the **clear counters cable** *interface* command. You can also clear the packet counters on all interfaces by using the **clear counters** command without any options. This allows you to use the **show cable arp** commands to display only the CPE devices that are currently generating the most traffic.

Note The **clear counters** command clears all of the packet counters on an interface, not just the ARP packet counters.

Identifying ARP Offenders in FP

When the FP ARP Filter feature is enabled, use the **show cable arp-filter** *interface* command to generate a list of ARP offenders.

cBR-8 Outputs in FP

When the FP ARP Filter feature is enabled, the cBR-8 output formatting displays the modem and the CPE addresses on a single line, in addition to the following columns:

- M/S—This column shows if packets are being filtered by MAC address or SID. A majority of these columns will show MAC address.
- Rate—This column shows the packet rate for FP-filtered packets in the last 5 minutes monitoring time window. Rate is not calculated for RP-filtered packets.
- Pro—This column will identify the processor that performed the filtering with either “RP” or “FP.” On the cBR-8, it is expected that 99.9% of Pro fields will show “FP.”

The following is a sample output for an ARP request on a cBR-8 in FP:

```
Router# show cable arp-filter Bundle1 requests-filtered 40
Interface Cable5/0/0 - none
Interface Cable6/0/2
Sid  CPE Mac        CPE IP          Modem MAC      Modem IP        M/S Rate Pro REQS
4    00d0.b75a.822a 50.3.81.56      0007.0e03.9cad 50.3.81.15      MAC -    RP  46
4    00d0.b75a.822a 50.3.81.56      0007.0e03.9cad 50.3.81.15      MAC 25   FP  5012
5    00b0.d07c.e51d 50.3.81.57      0007.0e03.1f59 50.3.81.13      MAC -    RP  64000
6    -              -               0006.2854.7347 50.3.81.4       MAC 101  FP  5122
7    -              -               0006.2854.72d7 50.3.81.11      SID -    FP  961205
Interface Cable7/0/0 - none
```

This sample output demonstrates the following:

- SID 4 shows a CPE filtered in FP. The threshold specified is low enough to show the packets that were filtered on the RP as the offender was being identified. A high enough threshold would not have shown the RP-filtered packets. The ARP packet rate of 25 is shown for FP-filtered packets.
- SID 5 shows a CPE filtered on the RP. This is extremely unusual and only occurs when the maximum number of FP-filterable entities has been reached.
- SID 6 shows a modem filtered in FP (CPE MAC or CPE IP are not shown).
- SID 7 shows ARP packets from an “unknown” source MAC address filtered by SID in FP.

The counts for requests, replies, and requests for IP will no longer be shown on a single line in order to keep the line concise and less than 90 characters in length.

The “REQs” column is now stated as “REPs” in the case of ARP replies. The column will show “REQ-IP” in cases involving ARP requests for IP.

Requests being sent by the CMTS due to encroaching IP packets, “ip-requests-filtered”, will still be filtered on the RP and not in FP, with Access Control Lists (ACLs) used to defeat IP-based scanning traffic, and the IP punt rate limiting feature for cBR-8 used to decrease the punt rate for such traffic. The ARP Filter can still be used to perform analysis of these IP traffic streams.

Configuration Examples for Cable ARP Filtering

This section provides the following examples of how to configure the Cable ARP Filtering features:

ARP Filtering Configuration on an Individual Cable Interface: Example

The following example shows a typical configuration of a cable interface that is configured for the Cable ARP Filtering feature:

```
!
interface Cable5/0/0
 ip address 192.168.100.1 255.255.255.0 secondary
 ip address 192.168.110.13 255.255.255.0
 cable downstream annex B
 cable downstream modulation 256qam
 cable downstream interleave-depth 32
 cable downstream channel-id 0
 cable upstream 0 frequency 6000000
 cable upstream 0 power-level 0
 cable upstream 0 channel-width 3200000 200000
 cable upstream 0 minislot-size 16
 cable upstream 0 modulation-profile 6 7
 no cable upstream 0 shutdown
 cable upstream 1 frequency 26000000
 cable upstream 1 power-level 0
 cable upstream 1 channel-width 3200000 200000
 cable upstream 1 minislot-size 4
 cable upstream 1 modulation-profile 6 7
 no cable upstream 1 shutdown
 cable upstream 2 frequency 15008000
 cable upstream 2 power-level 0
 cable upstream 2 channel-width 3200000 200000
 cable upstream 2 minislot-size 4
 cable upstream 2 modulation-profile 6 7
 cable upstream 2 shutdown
 cable upstream 3 spectrum-group 25
 cable upstream 3 channel-width 3200000 200000
 cable upstream 3 minislot-size 4
 cable upstream 3 modulation-profile 1
 cable upstream 3 shutdown
 cable upstream 4 frequency 21008000
 cable upstream 4 power-level 0
 cable upstream 4 channel-width 3200000 200000
 cable upstream 4 minislot-size 16
 cable upstream 4 modulation-profile 1
 no cable upstream 4 shutdown
 cable upstream 5 spectrum-group 25
 cable upstream 5 channel-width 3200000 200000
 cable upstream 5 minislot-size 4
 cable upstream 5 modulation-profile 1
 cable upstream 5 shutdown
 cable arp filter request-send 4 2
 cable arp filter reply-accept 4 2
end
```

ARP Filtering Configuration on Bundled Cable Interfaces: Example

The following example shows a typical configuration of a cable interface bundle that is also using the Cable ARP Filtering feature. Both the primary and subordinate interface are configured separately, allowing you to configure the feature only on the particular interfaces that require it. In addition, you can configure the feature with different threshold values, allowing you to customize the feature for each interface's traffic patterns.

```
!
interface Cable5/0/0
 description Master cable interface
 ip address 10.3.130.1 255.255.255.0 secondary
 ip address 10.3.131.1 255.255.255.0 secondary
 ip address 10.3.132.1 255.255.255.0 secondary
 ip address 10.3.133.1 255.255.255.0 secondary
```

```
 ip address 10.3.81.1 255.255.255.0
 ip helper-address 10.14.0.4
 load-interval 30
 cable bundle 1 master
 cable downstream annex B
 cable downstream modulation 64qam
 cable downstream interleave-depth 32
 cable downstream frequency 441000000
 cable downstream channel-id 0
 cable upstream 0 frequency 5008000
 cable upstream 0 power-level 0
 cable upstream 0 channel-width 1600000
 cable upstream 0 minislot-size 4
 cable upstream 0 modulation-profile 1
 no cable upstream 0 shutdown
 cable upstream 1 channel-width 1600000
 cable upstream 1 minislot-size 4
 cable upstream 1 modulation-profile 1
 cable upstream 1 shutdown
 cable upstream 2 channel-width 1600000
 cable upstream 2 minislot-size 4
 cable upstream 2 modulation-profile 1
 cable upstream 2 shutdown
 cable upstream 3 channel-width 1600000
 cable upstream 3 minislot-size 4
 cable upstream 3 modulation-profile 1
 cable upstream 3 shutdown
 cable arp filter request-send 4 2
 cable arp filter reply-accept 4 2
!
interface Cable7/0/0
 description Slave cable interface--Master is C5/0/0
 no ip address
 cable bundle 1
 cable downstream annex B
 cable downstream modulation 64qam
 cable downstream interleave-depth 32
 cable downstream frequency 562000000
 cable downstream channel-id 0
 no cable downstream rf-shutdown
 cable upstream 0 connector 0
 cable upstream 0 frequency 5008000
 cable upstream 0 power-level 0
 cable upstream 0 channel-width 1600000
 cable upstream 0 minislot-size 4
 cable upstream 0 modulation-profile 21
 no cable upstream 0 shutdown
 cable upstream 1 connector 1
 cable upstream 1 channel-width 1600000
 cable upstream 1 minislot-size 4
 cable upstream 1 modulation-profile 21
 cable upstream 1 shutdown
 cable upstream 2 connector 2
 cable upstream 2 channel-width 1600000
 cable upstream 2 minislot-size 4
 cable upstream 2 modulation-profile 21
 cable upstream 2 shutdown
 cable upstream 3 connector 3
 cable upstream 3 channel-width 1600000
 cable upstream 3 minislot-size 4
 cable upstream 3 modulation-profile 21
 cable upstream 3 shutdown
 cable arp filter request-send 20 5
```

```
 cable arp filter reply-accept 20 5
end
```

ARP Filtering in FP Default Configuration: Example

The following example shows the default configuration of a cable interface for the ARP Filtering in FP feature.

```
interface Bundle1
 cable arp filter request-send 3 2
 cable arp filter reply-accept 3 2
end
```

Additional References

The following sections provide references related to the Cable ARP Filtering feature.

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html
Source-Based Rate Limit	http://www.cisco.com/c/en/us/td/docs/cable/cbr/configuration/guide/b_cmts_sec_and_cable_mon_features_cbr/source-based_rate_limit.html
show platform hardware qfp active infrastructure punt summary command	http://www.cisco.com/c/en/us/td/docs/cable/cmts/cmd_ref/b_cmts_cable_cmd_ref/b_cmts_cable_cmd_ref_chapter_010100.html

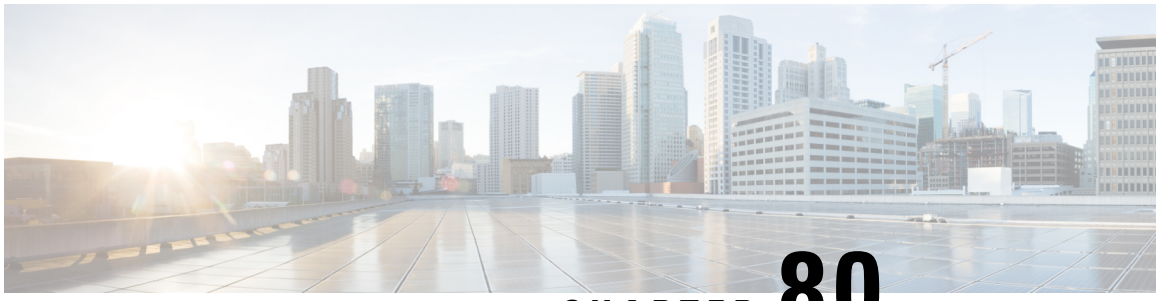
Feature Information for Cable ARP Filtering

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 197: Feature Information for the Cable ARP Filtering Feature

Feature Name	Releases	Feature Information
Cable ARP Filtering	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 80

Subscriber Management Packet Filtering Extension for DOCSIS 2.0

The Cisco converged broadband router supports management of data packet filtering based on the subscriber's preferences and criteria. Packet filtering enhances security to the cable network by allowing only the specific packets to flow to the Customer Premise Equipment (CPE) while dropping the unwanted data packets from the cable network.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1231
- Prerequisites for Configuring Subscriber Management Packet Filtering, on page 1232
- Restriction for Configuring Subscriber Management Packet Filtering, on page 1232
- Information About Configuring Subscriber Management Packet Filtering, on page 1233
- How to Configure Subscriber Management Packet Filtering, on page 1233
- Configuration Examples for Subscriber Management Packet Filtering, on page 1236
- Additional References, on page 1237
- Feature Information for Subscriber Management Packet Filtering, on page 1238

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 198: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Configuring Subscriber Management Packet Filtering

The software prerequisites for the subscriber management packet filtering feature are:

- The latest software image is loaded and working on the Cable Modem Termination System (CMTS) and the cable modems (CM).
- The configuration information on the main supervisor (SUP) and the standby SUP should be the same before the switchover.

Restriction for Configuring Subscriber Management Packet Filtering

- This feature can define up to 254 filtering groups. The number of filters in each group is 255.

Information About Configuring Subscriber Management Packet Filtering

A filter group specifies what filters are applied to the packets going to or coming from each specific CM or CPE device. It defines the rules or criteria to filter or drop a packet. Every packet that has to be filtered can either be accepted to send or filtered to be dropped. The criteria to filter a packet depends on the subscriber's preferences. The filter group can be applied to different subscriber management groups.

Cable subscriber management can be established using the following configuration methods:

- CMTS router configuration (via CLI)
- SNMP configuration

The process of configuring the subscriber management packet filtering is:

1. The packet filter group defines the action for a packet. The packet can be let to go to the CPE or dropped off the cable network based on the subscriber's packet criteria.
2. The CM sends a registration request to the CMTS. The registration request contains provisioning information that defines the association of a Packet Filtering Group (PFG) with the CM and its subscribers.
3. The specific downstream or upstream PFGs are used to bind the CM, CPE, embedded Multimedia Terminal Adaptor (eMTA), embedded Set-Top Box (eSTB) and embedded portal server (ePS) to a specific PFG.
4. The CMTS identifies the CPE device based on the CPE's DHCP information.

Note For the filter group to work for CMs, a CM must re-register after the CMTS router is configured.

How to Configure Subscriber Management Packet Filtering

This section describes the configuration tasks that are performed to manage subscriber packet filtering on the Cisco CMTS platforms. You can use the command-line interface (CLI) commands to complete the configuration.

Configuring the Filter Group

This section describes the tasks to configure the packet filter group. Follow the summary steps to complete the configuration.

To create, configure, and activate a DOCSIS filter group that filters packets on the basis of the TCP/IP and UDP/IP headers, use the cable filter group command in global configuration mode.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose

	`Router> enable`
	Example:
	`Router#`	

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	Example:
	`Router(config)#`
		Enters global configuration mode.

Step 3	**cable filter group group-id index index-num [option option-value]** **Example:** `Router(config)# cable filter group 10 index 10 src-ip 10.7.7.7`	Creates, configures, and activates a DOCSIS filter group that filters packets.

Defining the Upstream and Downstream MTA Filter Group

This section describes the configuration tasks to define the upstream and downstream subscriber management filter groups for an embedded Multimedia Terminal Adaptor (eMTA.) Follow the summary steps to complete the configuration.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	cable submgmt default filter-group mta {downstream | upstream} group-id **Example:** `Router(config)# cable submgmt default filter-group mta downstream 130`	Defines the upstream and downstream subscriber management filter groups for an MTA.

Defining the Upstream and Downstream STB Filter Group

This section describes the configuration tasks to define the upstream and downstream subscriber management filter groups for a Set-Top Box (STB.) Follow the summary steps to complete the configuration.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	cable submgmt default filter-group stb {downstream | upstream} group-id **Example:** `Router(config)# cable submgmt default filter-group stb downstream 20`	Defines the upstream and downstream subscriber management filter groups for an STB.

Defining the Upstream and Downstream PS Filter Group

This section describes the configuration tasks to define the upstream and downstream subscriber management filter groups for a Portal Server (PS.) Follow the summary steps to complete the configuration.

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	Example:
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	Example:
	`Router(config)#`
		Enters global configuration mode.

Step 3	cable submgmt default filter-group ps {downstream | upstream} group-id **Example:** `Router(config)# cable submgmt default filter-group ps downstream 10`	Defines the upstream and downstream subscriber management filter groups for a portal server.

Configuration Examples for Subscriber Management Packet Filtering

This section describes a sample configuration example for configuring the subscriber management packet filtering.

Configuring the Filter Group: Example

The following example shows configuration of a filter group that drops packets with a source IP address of 10.7.7.7 and a destination IP address of 10.8.8.8, and a source port number of 2000 and a destination port number of 3000. All protocol types and ToS and TCP flag values are matched:

```
Router(config)# cable filter group 10 index 10 src-ip 10.7.7.7
Router(config)# cable filter group 10 index 10 src-mask 255.255.0.0
Router(config)# cable filter group 10 index 10 dest-ip 10.8.8.8
Router(config)# cable filter group 10 index 10 dest-mask 255.255.0.0
```

```
Router(config)# cable filter group 10 index 10 ip-proto 256
Router(config)# cable filter group 10 index 10 src-port 2000
Router(config)# cable filter group 10 index 10 dest-port 3000
Router(config)# cable filter group 10 index 10 tcp-flags 0 0
Router(config)# cable filter group 10 index 10 match-action drop
```

Defining the Upstream and Downstream MTA Filter Group: Example

The following example shows configuration of an upstream and downstream MTA filter group.

```
Router# configure terminal
Router(config)# cable submgmt default filter-group mta downstream 10
```

Defining the Upstream and Downstream STB Filter Group: Example

The following example shows configuration of an upstream and downstream STB filter group.

```
Router#configure terminal
Router(config)#cable submgmt default filter-group stb downstream 20
```

Defining the Upstream and Downstream PS Filter Group: Example

The following example shows configuration of an upstream and downstream portal server filter group.

```
Router#configure terminal
Router(config)#cable submgmt default filter-group ps downstream 10
```

Additional References

The following sections provide references related to configuring the subscriber management packet filtering feature.

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

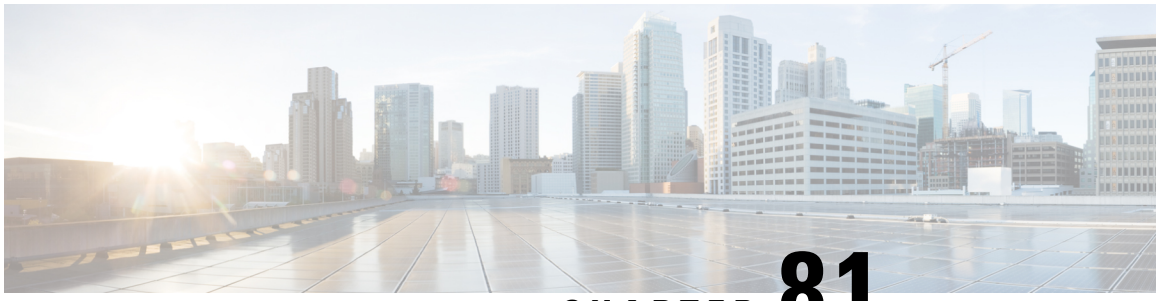
FeatureInformationforSubscriberManagementPacketFiltering

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 199: Feature Information for Subscriber Management Packet Filtering

Feature Name	Releases	Feature Information
Subscriber management packet filtering	Cisco IOS XE Everest 16.5.1	This feature was integrated into Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router



CHAPTER 81

MAC Filtering

This feature enables/disables MAC address filter on the backhaul interface.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1239
- Information About MAC Filtering, on page 1240
- How to Configure MAC Filtering, on page 1240
- Configuration Examples for MAC Filtering, on page 1244
- Feature Information for MAC Filtering, on page 1244

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 200: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About MAC Filtering

With this feature, only the packet whose destination MAC address is the MAC address of the router interface can be forwarded. It supports 32 unicast filter entries per interface. It is disabled by default.

Note When port-channel is enabled, MAC filtering must be enabled on backhaul interface to take effect.

Note When both dot1q l2vpn and MAC filtering are enabled on backhaul interface, only 1 unicast filter entry is supported per backhaul interface. The MAC filtering is only supported for non-l2vpn unicast packets.

How to Configure MAC Filtering

This section describes the configuration tasks that are performed to manage MAC filtering. You can use the command-line interface (CLI) commands to complete the configuration.

Configuring MAC Filtering

To configure MAC filtering, follow the steps below:

```
enable
configure terminal
interface tenGigabitEthernet slot/subslot/port
mac-addr-filter
end
```

Verifying MAC Filtering

To verify the MAC filtering configuration on the backhaul interface, use**show running-config interface** command as shown below:

```
Router# show running-config interface tenGigabitEthernet 4/1/0
Building configuration...

Current configuration : 73 bytes
!
interface TenGigabitEthernet4/1/0
 no ip address
 mac-addr-filter
end
```

To verify the MAC filtering status on a specific SUP slot and SUP-PIC bay, use**show platform software iomd** command as shown below:

```
Router# show platform software iomd 4/4 mac-filter
IOMD (Input Output Module Driver) Mac Filter Status


port: 0     promiscuous mode:     unicast:  enable   multicast:  enable      broadcast:
enable
            Input Drop cnt:                       0      Total Drop cnt:
    0
            Entry Number:   1
    Index      Mode    Action            Entry MAC           Entry MASK             Match
Count
       00    enable      pass    c4:14:3c:16:7c:04    ff:ff:ff:ff:ff:ff
    0


port: 1     promiscuous mode:     unicast:  enable   multicast:  enable      broadcast:
enable
            Input Drop cnt:                       0      Total Drop cnt:
    0
            Entry Number:   1
    Index      Mode    Action            Entry MAC           Entry MASK             Match
Count
       00    enable      pass    c4:14:3c:16:7c:05    ff:ff:ff:ff:ff:ff
 1729


port: 2     promiscuous mode:     unicast:  enable   multicast:  enable      broadcast:
enable
            Input Drop cnt:                       0      Total Drop cnt:
    0
            Entry Number:   1
```

```
    Index      Mode    Action            Entry MAC           Entry MASK             Match
Count
       00    enable      pass    c4:14:3c:16:7c:06    ff:ff:ff:ff:ff:ff
    0


port: 3     promiscuous mode:     unicast:  enable   multicast:  enable      broadcast:
enable
            Input Drop cnt:                     0      Total Drop cnt:
    0
            Entry Number:   1
    Index      Mode    Action            Entry MAC           Entry MASK             Match
Count
       00    enable      pass    c4:14:3c:16:7c:07    ff:ff:ff:ff:ff:ff
    0


port: 4     promiscuous mode:     unicast:  enable   multicast:  enable      broadcast:
enable
            Input Drop cnt:                     0      Total Drop cnt:
    0
            Entry Number:   1
    Index      Mode    Action            Entry MAC           Entry MASK             Match
Count
       00    enable      pass    c4:14:3c:16:7c:08    ff:ff:ff:ff:ff:ff
    0


port: 5     promiscuous mode:     unicast:  enable   multicast:  enable      broadcast:
enable
            Input Drop cnt:                     0      Total Drop cnt:
    0
            Entry Number:   1
    Index      Mode    Action            Entry MAC           Entry MASK             Match
Count
       00    enable      pass    c4:14:3c:16:7c:09    ff:ff:ff:ff:ff:ff
   15


port: 6     promiscuous mode:     unicast:  enable   multicast:  enable      broadcast:
enable
            Input Drop cnt:                     0      Total Drop cnt:
    0
            Entry Number:   1
    Index      Mode    Action            Entry MAC           Entry MASK             Match
Count
       00    enable      pass    c4:14:3c:16:7c:0a    ff:ff:ff:ff:ff:ff
    0


port: 7     promiscuous mode:     unicast:  enable   multicast:  enable      broadcast:
enable
            Input Drop cnt:                     0      Total Drop cnt:
    0
            Entry Number:   1
    Index      Mode    Action            Entry MAC           Entry MASK             Match
Count
       00    enable      pass    c4:14:3c:16:7c:0b    ff:ff:ff:ff:ff:ff
    0
```

If the MAC filtering is disabled, the output of the **show platform software iomd** command is shown as below:

```
Router# show platform software iomd 4/5 mac-filter
IOMD (Input Output Module Driver) MAC filter Status
```

```
port: 0     promiscuous mode:     unicast:  enable   multicast:  enable      broadcast:
enable
            Input Drop cnt:                    0      Total Drop cnt:
    0
            Entry Number:   0

port: 1     promiscuous mode:     unicast:  enable   multicast:  enable      broadcast:
enable
            Input Drop cnt:                    0      Total Drop cnt:
    0
            Entry Number:   0

port: 2     promiscuous mode:     unicast:  enable   multicast:  enable      broadcast:
enable
            Input Drop cnt:                    0      Total Drop cnt:
    0
            Entry Number:   0

port: 3     promiscuous mode:     unicast:  enable   multicast:  enable      broadcast:
enable
            Input Drop cnt:                    0      Total Drop cnt:
    0
            Entry Number:   0

port: 4     promiscuous mode:     unicast:  enable   multicast:  enable      broadcast:
enable
            Input Drop cnt:                    0      Total Drop cnt:
    0
            Entry Number:   0

port: 5     promiscuous mode:     unicast:  enable   multicast:  enable      broadcast:
enable
            Input Drop cnt:                    0      Total Drop cnt:
    0
            Entry Number:   0

port: 6     promiscuous mode:     unicast:  enable   multicast:  enable      broadcast:
enable
            Input Drop cnt:                    0      Total Drop cnt:
    0
            Entry Number:   0

port: 7     promiscuous mode:     unicast:  enable   multicast:  enable      broadcast:
enable
            Input Drop cnt:                    0      Total Drop cnt:
    0
            Entry Number:   0
```

Configuration Examples for MAC Filtering

This section describes a sample configuration example for configuring the MAC filtering.

```
router> enable
router# configure terminal
router(config)# interface tenGigabitEthernet 4/1/0
router(config-if)# mac-addr-filter
router(config-if)# end
```

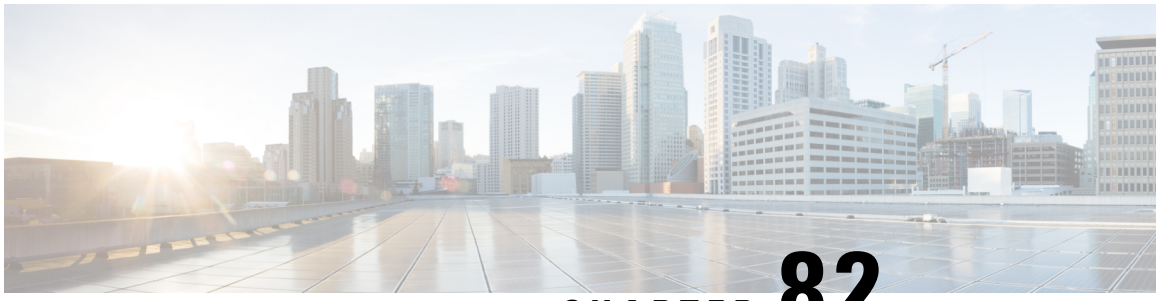
Feature Information for MAC Filtering

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 201: Feature Information for MAC Filtering

Feature Name	Releases	Feature Information
MAC Filtering	Cisco IOS XE Everest 16.5.1	This feature was introduced in Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Routers.



CHAPTER 82

IPv6 Segment Routing on Cisco cBR

In Cisco Converged Broadband Router, IPv6 Segment Routing is available as a sub mode of IPv6 address configuration.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1245
- Information about IPv6 Segment Routing, on page 1246
- How to Configure IPv6 Segment Routing, on page 1247
- Configuration Examples, on page 1249
- Feature Information for IPv6 Segment Routing, on page 1250

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 202: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information about IPv6 Segment Routing

IPv6 Segment Routing (SR) is an SDN technology supporting IPv6 forwarding. In SR, a source or edge router performs source routing of traffic and encodes it as a segment list in an IPv6 routing extension header. The network is not required to maintain a per-application or per-flow state.

Any IPv6 capable node in a network may forward IPv6 traffic with an SR extension header to the first segment in the segment list without supporting IPv6 Segment Routing (SRv6).

At the node that hosts the current segment in the segment list, SRv6 is configured to modify the destination address of the traffic containing the SR extension header and destined to that segment ID. As part of SRv6 final processing, the next segment ID in the SR extension header is written to the destination address of the packet and a lookup is performed to forward the traffic to the new destination address.

The forwarding and SRv6 end processing continues at nodes hosting the segment IDs in the SR extension header until the last segment in the list is removed and the traffic is delivered to its ultimate destination.

Restriction for Configuring IPv6 Segment Routing

Configuring duplicate IPv6 addresses on the same interface is not allowed.

How to Configure IPv6 Segment Routing

Configuring IPv6 Segment Routing on cBR

To configure IPv6 segment routing, use the following procedure.

1. Enter segment-routing sub mode when configuring an IPv6 address on an interface.

```
enable
configure terminal
interface type [slot_#/]port_#
ipv6 address ipv6_address_prefix/prefix_length
ipv6 address ipv6_address_prefix/prefix_length segment-routing
```

2. Define a local prefix as an SID

```
ipv6-sr prefix-sid
exit
```

Verifying IPv6 Segment Routing Configuration

The following example shows how to verify SRv6 configuration:c

```
Router#sh run
*Oct 17 13:13:23.975: %SYS-5-CONFIG_I: Configured from console by console
Router#sh run | sec Ether
interface Ethernet0/0
 no ip address
 shutdown
 ipv6 address 2001::2001/64 segment-routing >>>>>>>
  ipv6-sr prefix-sid >>>>>>>
```

Configure Multiple IPv6 Addresses for Segment Routing

To configure multiple IPv6 addresses for SRv6 under the same interface, use the following commands.

1. Enter segment-routing sub mode when configuring an IPv6 address on an interface.

```
enable
configure terminal
interface type [slot_#/]port_#
ipv6 address ipv6_address_prefix/prefix_length segment-routing
```

2. Define a local prefix as an SID.

```
ipv6-sr prefix-sid
exit
```

Verifying IPv6 Segment Routing Configuration on Multiple IPv6 Addresses

The following example shows how to verify SRv6 configuration for multiple IPv6 addresses:

```
Router#sh run | sec Ether
interface Ethernet0/0
```

```
 no ip address
 shutdown
 ipv6 address 2001:db8:110::/64 segment-routing  >>> submode 1
  ipv6-sr prefix-sid
 ipv6 address 2001:db9:111::/64 segment-routing  >>> submode 2
  ipv6-sr prefix-sid
interface Ethernet0/1
 no ip address
 shutdown
interface Ethernet0/2
 no ip address
 shutdown
interface Ethernet0/3
 no ip address
 shutdown
interface Ethernet1/0
 no ip address
 shutdown
interface Ethernet1/1
 no ip address
 shutdown
interface Ethernet1/2
```

Disabling Prefix SID

To disable the local prefix SID associated to the segment ID, use the following commands.

```
enable
configure terminal
interface type [slot_#/]port_#
ipv6 address ipv6_address_prefix/prefix_length segment-routing
no ipv6-sr prefix-sid
end
```

Verifying whether Prefix SID is Disabled

The following example shows how to verify whether the prefix SID is disabled:

```
Router#sh run | sec Ether
interface Ethernet0/0
 no ip address
 shutdown
 ipv6 address 110::110/64 segment-routing >>> "ipv6-sr prefix sid" is no longer present
ipv6 address 111::111/64 segment-routing
ipv6-sr prefix-sid
```

Disabling SRv6 for a Prefix-SID

To disable SRv6 configuration for an IPv6 address and remove the IPv6 address, use the following command:

```
enable
configure terminal
interface type [slot_#/]port_#
no ipv6 address ipv6_address_prefix/prefix_length segment-routing
end
```

Verifying whether SRv6 is Disabled and Prefix SID Removed

The following example shows how to verify whether SRv6 is disabled and the prefix SID is removed for the prefix SID.

```
Router#sh run |
*Oct 17 13:17:51.523: %SYS-5-CONFIG_I: Configured from console by console
Router#sh run | sec Ether
interface Ethernet0/0
 no ip address
 shutdown
 ipv6 address 110::110/64 segment-routing
ipv6 address 111::111/64 segment-routing is entirely removed from ethernet0/0
```

Configuration Examples

This section provides examples for IPv6 Segment Routing.

Example: Configuring IPv6 Segment Routing on Cisco cBR

```
Router#conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#inter Ether0/0
Router(config-if)#ipv6 address 110::110/64 ?
  anycast
  eui-64
  segment-routing
  <cr>
Router(config-if)#ipv6 address 110::110/64 segment-routing
Router(config-if-sr-ipv6)#?
ipv6 address segment-routing mode configuration commands:
  default  Set a command to its defaults
  exit     Exit from SR submode
  ipv6-sr  Request options specific to IPV6 segment-routing
  no       Negate a command or set its defaults
Router(config-if-sr-ipv6)#ipv6-sr ?
  prefix-sid  Set host prefix as IPv6 SR identifier prefix-sid
Router(config-if-sr-ipv6)#ipv6-sr prefix-sid
Router(config-if-sr-ipv6)#exit
Router(config-if)#exit
Router(config)#exit
Router#
```

Example: Configure Multiple IPv6 Addresses for SRv6

```
Router#conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#inter Ether 0/0
Router(config-if)# ipv6 address 110::110/64 segment-routing
Router(config-if)# ipv6 address 111::111/64 segment-routing
Router(config-if-sr-ipv6)#ipv6-sr prefix-sid
Router(config-if-sr-ipv6)#end
```

Example: Disabling Prefix SID

```
Router#conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#inter Ether0/0
```

```
Router(config-if)#inter Ether0/0
Router(config-if)#ipv6 address 110::110/64 segment-routing
Router(config-if-sr-ipv6)#no ipv6-sr prefix-sid
Router(config-if-sr-ipv6)#end
```

Example: Disabling SR with an Active Prefix SID

```
Router#conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#inter Ether0/0
Router(config-if)#no ipv6 address 111::111/64 segment-routing
Router(config-if)#end
```

Feature Information for IPv6 Segment Routing

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

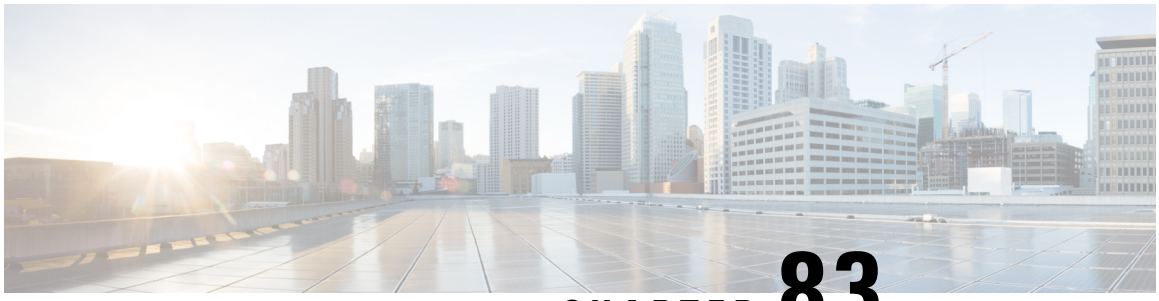
Table 203: Feature Information for IPv6 Segment Routing

Feature Name	Releases	Feature Information
IPv6 Segment Routing	Cisco IOS XE Everest 16.5.1	This feature was introduced in Cisco IOS XE Everest 16.5.1 on the Cisco cBR Series Converged Broadband Router s.

PART XII

Troubleshooting and Network Management Configuration

- Call Home, on page 1253
- SNMP Support over VPNs—Context-Based Access Control, on page 1319
- SNMP Engine Enhancement, on page 1331
- Onboard Failure Logging, on page 1337
- Control Point Discovery, on page 1343
- IPDR Streaming Protocol, on page 1353
- Usage-Based Billing (SAMIS), on page 1365
- Frequency Allocation Information for the Cisco CMTS Routers, on page 1409
- Flap List Troubleshooting, on page 1421
- Maximum CPE and Host Parameters, on page 1441
- SNMP Background Synchronization, on page 1451
- Online Offline Diagnostics, on page 1461



CHAPTER 83

Call Home

Call Home offers proactive diagnostics and real-time alerts on select Cisco devices, which provides higher network availability and increased operational efficiency. Smart Call Home is a secure connected service of Cisco SMARTnet for the Cisco cBR routers.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1254
- Prerequisites for Call Home, on page 1254
- Restrictions for Call Home, on page 1255
- Information About Call Home, on page 1255
- How to Configure Call Home, on page 1257
- Configuring Diagnostic Signatures, on page 1281
- Verifying the Call Home Configuration, on page 1289
- Configuration Example for Call Home, on page 1294
- Default Settings, on page 1299
- Alert Groups Trigger Events and Commands, on page 1300
- Message Contents, on page 1304
- Additional References, on page 1317
- Feature Information for Call Home, on page 1318

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 204: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Call Home

- Contact e-mail address (required for full registration with Smart Call Home, optional if Call Home is enabled in anonymous mode), phone number (optional), and street address information (optional) should be configured so that the receiver can determine the origin of messages received.

Note Contact e-mail address is not required if you enable Smart Call Home by enabling smart licensing.

- At least one destination profile (predefined or user-defined) must be configured. The destination profile(s) configured depends on whether the receiving entity is a pager, e-mail, or automated service such as Cisco Smart Call Home.
  - If the destination profile uses e-mail message delivery, you must specify a Simple Mail Transfer Protocol (SMTP) server.
  - Configuring the trustpool certificate authority (CA) is not required for HTTPS server connection as the trustpool feature is enabled by default.
- The router must have IP connectivity to an e-mail server or the destination HTTP(S) server.
- To use Cisco Smart Call Home service, an active service contract covering the device is required to provide full Smart Call Home service.

Note An active service contract is only required for full Smart Call Home services like automatically raising a Cisco Technical Assistance Center (TAC) case.

Restrictions for Call Home

- If there is no IP connectivity or if the interface in the VRF to the profile destination is down, Smart Call Home messages cannot be sent.
- Smart Call Home operates with any SMTP server.
- You can configure up to five SMTP servers for Smart Call Home.

Information About Call Home

The Call Home feature provides email-based and web-based notification of critical system events. A versatile range of message formats are available for optimal compatibility with pager services, standard email, or XML-based automated parsing applications. Common uses of this feature may include direct paging of a network support engineer, email notification to a network operations center, XML delivery to a support website, and use of Cisco Smart Call Home services for direct case generation with the Cisco Systems Technical Assistance Center (TAC).

The Call Home feature can deliver alert messages containing information on configuration, environmental conditions, inventory, syslog, snapshot, and crash events.

The Call Home feature can deliver alerts to multiple recipients, referred to as *Call Home destination profiles* , each with configurable message formats and content categories. A predefined destination profile (CiscoTAC-1) is provided, and you also can define your own destination profiles. The CiscoTAC-1 profile is used to send alerts to the backend server of the Smart Call Home service, which can be used to create service requests to Cisco TAC, the service will depend on the Smart Call Home service support in place for your device and the severity of the alert.

Flexible message delivery and format options make it easy to integrate specific support requirements.

Benefits of Call Home

- Automatic execution and attachment of relevant CLI command output.
- Multiple message format options such as the following:
  - Short Text—Suitable for pagers or printed reports.
  - Full Text—Fully formatted message information suitable for human reading.
  - XML—Matching readable format using Extensible Markup Language (XML). The XML format enables communication with the Cisco TAC.
- Multiple concurrent message destinations.
- Multiple message categories including configuration, crash, diagnostic, environment, inventory, snapshot, and syslog.
- Filtering of messages based on severity and pattern matching.
- Scheduling of periodic message sending.

Obtaining Smart Call Home Services

If you have a service contract directly with Cisco, you can register for the Smart Call Home service. Smart Call Home analyzes Smart Call Home messages and provides background information and recommendations. For critical issues, Automatic Service Requests are generated with the Cisco TAC.

Smart Call Home offers the following features:

- Continuous device health monitoring and real-time alerts.
- Analysis of Smart Call Home messages and, if needed, Automatic Service Request generation routed to the correct TAC team, including detailed diagnostic information to speed problem resolution.
- Secure message transport directly from your device or through an HTTP proxy server or a downloadable Transport Gateway (TG). You can use a TG aggregation point to support multiple devices or in cases where security dictates that your devices may not be connected directly to the Internet.
- Web-based access to Smart Call Home messages and recommendations, inventory, and configuration information for all Smart Call Home devices provides access to associated field notices, security advisories, and end-of-life information.

You need the following items to register for Smart Call Home:

- SMARTnet contract number for your router.
- Your e-mail address
- Your Cisco.com username

For information about how to configure Call Home to work with the Smart Call Home service, see the Smart Call Home.

Anonymous Reporting

Smart Call Home is a service capability included with many Cisco service contracts and is designed to assist customers resolve problems more quickly. In addition, the information gained from crash messages helps Cisco understand equipment and issues occurring in the field. If you decide not to use Smart Call Home, you can still enable Anonymous Reporting to allow Cisco to securely receive minimal error and health information from the device. If you enable Anonymous Reporting, your customer identity will remain anonymous, and no identifying information is sent.

Note When you enable Anonymous Reporting, you acknowledge your consent to transfer the specified data to Cisco or to vendors operating on behalf of Cisco (including countries outside the United States). Cisco maintains the privacy of all customers. For information about how Cisco treats personal information, see the Cisco Privacy Statement at Cisco Online Privacy Statement

When Call Home is configured in an anonymous way, only crash, inventory, and test messages are sent to Cisco. No identifying information is sent.

For more information about what is sent in these messages, see the Alert Group Trigger Events and Commands section.

Smart Licensing

Smart Licensing uses the Smart Call Home service.

The Smart Licensing service is an alternative licensing architecture to Cisco Software Licensing (CSL). Smart Licensing uses the Cisco Smart Software Manager as a backend tool for managing licenses. Smart Call Home must be configured before using the Smart Licensing. By default, Smart Licensing and Smart Call Home are enabled on the Cisco cBR routers.

For more information about Smart Licensing, see Cisco Smart Licensing on the Cisco cBR Router.

How to Configure Call Home

Configuring Smart Call Home (Single Command)

Smart Call Home is enabled by default on the router. The CiscoTAC-1 profile to send data to Cisco is also enabled by default.

You do not need to use the single command to enable Smart Call Home on the router unless you need to change to anonymous mode or add HTTP proxy using the single command.

To enable all Call Home basic configurations using a single command, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	Device# **configure terminal**	
Step 2	**call-home reporting** { **anonymous** | **contact-email-addr** *email-address* } [**http-proxy** { *ipv4-address* | *ipv6-address* | **name** } **port** *port number*] **Example:** Device(config)# **call-home reporting contact-email-addr email@company.com**	Enables all Call Home basic configurations using a single command. • **anonymous** —Enables Call-Home TAC profile to only send crash, inventory, and test messages and send the messages in an anonymous way. • **contact-email-addr** —Enables Smart Call Home service full reporting capability and sends a full inventory message from Call-Home TAC profile to Smart Call Home server to start full registration process. • **http-proxy** { *ipv4-address* | *ipv6-address* | **name** —An ipv4 or ipv6 address or server name. Maximum length is 64. • **port** *port number* —Port number. Range is 1 to 65535. **Note** HTTP proxy option allows you to make use of your own proxy server to buffer and secure internet connections from your devices. **Note** After successfully enabling Call Home either in anonymous or full registration mode using the **call-home reporting** command, an inventory message is sent out. If Call Home is enabled in anonymous mode, an anonymous inventory message is sent out. If Call Home is enabled in full registration mode, a Full Inventory message for full registration mode is sent. For more information about what is sent in these messages, see Alert Groups Trigger Events and Commands, on page 1300.

Configuring Call Home

For security reasons, we recommend that you use the HTTPS transport options, due to the additional payload encryption that HTTPS offers. The Transport Gateway software is downloadable from Cisco.com and is available if you require an aggregation point or a proxy for connection to the Internet.

The implementation on the router supports the trustpool feature (embedded CA certificates in IOS images). The trustpool feature simplifies configuration to enable Smart Call Home service on configured devices. It eliminates the requirement of manually configuring the trustpool and provides automatic update of the CA certificate should it change in the future.

Enabling and Disabling Call Home

To enable or disable the Call Home feature, complete the following steps:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	**service call-home** **Example:** `Router(config)#` **`service call-home`**	Enables the Call Home feature.
Step 3	**no service call-home** **Example:** `Router(config)#` **`no service call-home`**	Disables the Call Home feature.

Configuring Contact Information

Each router must include a contact e-mail address. You can optionally include a phone number, street address, contract ID, customer ID, and site ID.

To assign the contact information, complete the following steps:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router>` **`configure terminal`**	Enters global configuration mode.
Step 2	**call-home** **Example:** `Router(config)#` **`call-home`**	Enters call home configuration mode.
Step 3	**contact-email-addr** *email-address* **Example:** `Router(cfg-call-home)#` **`contact-email-addr username@example.com`**	Assigns the customer's e-mail address. Enter up to 200 characters in e-mail address format with no spaces.
Step 4	**phone-number** + *phone-number* **Example:** `Router(cfg-call-home)#` **`phone-number +1-222-333-4444`**	(Optional) Assigns the customer's phone number. **Note** The number must begin with a plus (**+**) prefix, and may contain only dashes (-) and numbers. Enter up to 16 characters. If you include spaces, you must enclose your entry within double quotation marks (" ").
Step 5	**street-address** *street-address* **Example:** `Router(cfg-call-home)#` **`street-address "1234 Any Street, Any city, Any state, 12345"`**	(Optional) Assigns the customer's street address where RMA equipment can be shipped. Enter up to 200 characters. If you include spaces, you must enclose your entry within double quotation marks (" ").
Step 6	**customer-id** *text* **Example:** `Router(cfg-call-home)#` **`customer-id Customer1234`**	(Optional) Identifies the customer ID. Enter up to 64 characters. If you include spaces, you must enclose your entry within double quotation marks (" ").

	Command or Action	Purpose
Step 7	**site-id** *text* **Example:** `Router(cfg-call-home)#` **`site-id Site1ManhattanNY`**	(Optional) Identifies the customer site ID. Enter up to 200 characters. If you include spaces, you must enclose your entry within double quotation marks (“ ”).
Step 8	**contract-id** *text* **Example:** `Router(cfg-call-home)#` **`contract-id Company1234`**	(Optional) Identifies the customer’s contract ID for the router. Enter up to 64 characters. If you include spaces, you must enclose your entry within double quotation marks (“ ”).

Configuring Destination Profiles

A destination profile contains the required delivery information for an alert notification. At least one destination profile is required. You can configure multiple destination profiles of one or more types.

You can create and define a new destination profile or copy and use the predefined destination profile. If you define a new destination profile, you must assign a profile name. You can control which profile to be used for Smart Licensing by enabling or disabling smart-licensing data of that profile. Only one active profile can have smart-license data enabled.

Note If you use the Smart Call Home service, the destination profile must use the XML message format.

A destination profile includes the following information:

- Profile name—String that uniquely identifies each user-defined destination profile. The profile name is limited to 31 characters and is not case-sensitive. You cannot use **all** as a profile name.
- Transport method—Transport mechanism, either email or HTTP (including HTTPS), for delivery of alerts.
  - For user-defined destination profiles, email is the default, and you can enable either or both transport mechanisms. If you disable both methods, email is enabled.
  - For the predefined CiscoTAC-1 profile, you can enable either transport mechanism, but not both.
- Destination address—The actual address related to the transport method by which the alert should be sent. You can change the destination of the CiscoTAC-1 profile.
- Message formatting—The message format used for sending the alert. The format options for a user-defined destination profile are long-text, short-text, or XML. The default is XML. For the predefined CiscoTAC-1 profile, only XML is allowed.
- Message size—The maximum destination message size. The valid range is 50 to 3,145,728 bytes. The default is 3,145,728 bytes.

- Reporting method—You can choose which data to report for a profile. You can enable reporting of Smart Call Home data or Smart Licensing data, or both. Only one active profile is allowed to report Smart Licensing data at a time.
- Anonymous reporting—You can choose for your customer identity to remain anonymous, and no identifying information is sent.
- Subscribing to interesting alert-groups—You can choose to subscribe to alert-groups highlighting your interests.
- Message severity level—The Call Home severity level that the alert must meet before a Call Home message is generated to all e-mail addresses in the destination profile. An alert is not generated if the Call Home severity level of the alert is lower than the message severity level set for the destination profile.

You can also configure a destination profile to allow periodic inventory update messages by using the inventory alert group.

A pre-defined destination profile CiscoTAC-1 is supported. It supports the XML message format. This profile is preconfigured with Cisco Smart Call Home server HTTPS URL, email address to reach the server, maximum message size, and message severity level for each alert group.

Important We recommend that you do not use the message severity level 0. If you use message severity level 0, all syslogs will trigger Call Home messages, which can cause CPU and memory issues.

This section contains the following:

Creating a New Destination Profile

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**call-home** **Example:** `Router(config)# call-home`	Enters Call Home configuration mode.
Step 4	**profile** *name* **Example:** `Router(cfg-call-home)# profile profile1`	Enters the Call Home destination profile configuration mode for the specified destination profile. If the specified destination profile does not exist, it is created.

	Command or Action	Purpose
Step 5	**destination transport-method** { **email** | **http** } **Example:** `Router(cfg-call-home-profile)# destination transport-method email`	(Optional) Enables the message transport method. • **email** —Sets the e-mail message transport method. • **http** —Sets the HTTP message transport method. **Note** The **no** option disables the method.
Step 6	**destination address** { **email** *email-address* | **http** *url* } **Example:** `Router(cfg-call-home-profile)# destination address email myaddress@example.com`	Configures the destination e-mail address or URL to which Call Home messages are sent. **Note** When entering a destination URL, include either **http://** or **https://**, depending on whether the server is a secure server. If the destination is a secure server, you must also configure a trustpool CA.
Step 7	**destination preferred-msg-format** { **long-text** | **short-text** | **xml** } **Example:** `Router(cfg-call-home-profile)# destination preferred-msg-format xml`	(Optional) Configures a preferred message format. The default is XML. • **long-text** —Configures the long text message format. • **short-text** —Configures the short text message format. • **xml** —Configures the XML message format.
Step 8	**destination message-size** *bytes* **Example:** `Router(cfg-call-home-profile)# destination message-size 3,145,728`	(Optional) Configures a maximum destination message size for the destination profile.
Step 9	**active** **Example:** `Router(cfg-call-home-profile)# active`	Enables the destination profile. By default, the profile is enabled when it is created. If you activate a profile which enables smart-licensing data while smart-licensing data is already being reported in another active profile, you will receive an error message.
Step 10	**reporting** { **all** | **smart-call-home-data** | **smart-licensing-data** } **Example:**	Configures the type of data to report for a profile. You can select either to report Smart Call Home data or Smart Licensing data. Selecting

	Command or Action	Purpose
	`Router(cfg-call-home-profile)# reporting smart-call-home-data`	the **all** option reports data for both types of data.
Step 11	**end** **Example:** `Router(cfg-call-home-profile)# end`	Returns to privileged EXEC mode.
Step 12	**show call-home profile** { *name* | **all** } **Example:** `Router# show call-home profile profile1`	Displays destination profile configuration for specified profile or all configured profiles.
Step 13	**show call-home smart-licensing** **Example:** `Router# show call-home smart-licensing`	Displays the current Call Home Smart Licensing settings for the configured destination profiles.
Step 14	**show call-home smart-licensing statistics** **Example:** `Router# show call-home smart-licensing statistics`	Displays the Call Home Smart Licensing statistics.

Copying a Destination Profile

You can create a new destination profile by copying an existing profile

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**call-home** **Example:** `Router(config)# call-home`	Enters Call Home configuration mode.
Step 4	**copy profile** *source-profile target-profile* **Example:** `Router(cfg-call-home)# copy profile profile1 profile2`	Creates a new destination profile with the same configuration settings as the existing destination profile. • *source-profile* —Name of the source destination profile.

	Command or Action	Purpose
		• *target-profile* —Name of the target or new destination profile.

Renaming a Destination Profile

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**call-home** **Example:** `Router(config)# call-home`	Enters Call Home configuration mode.
Step 4	**rename profile** *source-profile target-profile* **Example:** `Router(cfg-call-home)# rename profile profile1 profile2`	Renames the existing destination profile. • *source-profile* —Name of the source destination profile. • *target-profile* —Name of the target destination profile.

Setting Profiles to Anonymous Mode

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**call-home** **Example:** `Router(config)# call-home`	Enters Call Home configuration mode.

	Command or Action	Purpose
Step 4	**profile** *name* **Example:** `Router(cfg-call-home)# profile profile1`	Enters the Call Home destination profile configuration mode for the specified destination profile. If the specified destination profile does not exist, it is created.
Step 5	**anonymous-reporting-only** **Example:** `Router(cfg-call-home-profile)# anonymous-reporting-only`	Sets the profile to anonymous mode. **Note** By default, the profile sends a full report of all types of events subscribed in the profile. When **anonymous-reporting-only** is set, only crash, inventory, and test messages are sent.

Subscribing to Alert Groups

An alert group is a predefined subset of Call Home alerts that are supported. You can select one or more alert groups to be received by a destination profile.

- Configuration
- Crash
- Diagnostic
- Environment
- Inventory
- Snapshot
- Syslog

The triggering events for each alert group are listed in the Alert Groups Trigger Events and Commands, and the contents of the alert group messages are listed in the Message Contents.

You can select one or more alert groups to be received by a destination profile.

Note Call Home alerts are only sent to destination profiles that have subscribed to the alert group containing that Call Home alert. The alert group must be enabled. The Call Home event severity must be at or above the message severity set in the destination profile.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password, if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**call-home** **Example:** `Router(config)# call-home`	Enters Call Home configuration mode.
Step 4	**alert-group** { **all** | **configuration** | **crash** | **diagnostic** | **environment** | **inventory** | **snapshot** | **syslog** } **Example:** `Router(cfg-call-home)# alert-group all`	Enables the specified alert group. Use the keyword **all** to enable all alert groups. By default, all alert groups are enabled.
Step 5	**profile** *name* **Example:** `Router(cfg-call-home)# profile profile1`	Enters the Call Home destination profile configuration mode for the specified destination profile. If the specified destination profile does not exist, it is created.
Step 6	**subscribe-to-alert-group configuration** [**periodic** { **daily** *hh:mm* | **monthly** *date hh:mm* | **weekly** *day hh:mm* }] **Example:** `Router(cfg-call-home-profile)# subscribe-to-alert-group configuration periodic daily 12:00`	Subscribes this destination profile to the Configuration alert group. The Configuration alert group can be configured for periodic notification.
Step 7	**subscribe-to-alert-group crash** **Example:** `Router(cfg-call-home-profile)# subscribe-to-alert-group crash`	Subscribes to the Crash alert group in user profile. By default, Cisco TAC profile subscribes to the Crash alert group and cannot be unsubscribed.
Step 8	**subscribe-to-alert-group diagnostic** [**severity** { **catastrophic** | **disaster** | **fatal** | **critical** | **major** | **minor** | **warning** | **notification** | **normal** | **debugging** }] **Example:** `Router(cfg-call-home-profile)# subscribe-to-alert-group syslog severity major`	Subscribes this destination profile to the Diagnostic alert group. The Diagnostic alert group can be configured to filter messages based on severity.
Step 9	**subscribe-to-alert-group environment** [**severity** { **catastrophic** | **disaster** |	Subscribes this destination profile to the Environment alert group. The Environment

	Command or Action	Purpose
	fatal | **critical** | **major** | **minor** | **warning** | **notification** | **normal** | **debugging** }] **Example:** `Router(cfg-call-home-profile)# subscribe-to-alert-group environment severity major`	alert group can be configured to filter messages based on severity.
Step 10	**subscribe-to-alert-group inventory** [**periodic** { **daily** *hh:mm* | **monthly** *date hh:mm* | **weekly** *day hh:mm* }] **Example:** `Router(cfg-call-home-profile)# subscribe-to-alert-group inventory periodic daily 12:00`	Subscribes this destination profile to the Inventory alert group. The Inventory alert group can be configured for periodic notification
Step 11	**subscribe-to-alert-group snapshot** [**periodic** { **daily** *hh:mm* | **monthly** *date hh:mm* | **weekly** *day hh:mm* | **hourly** *mm* | **interval** *mm* }] **Example:** `Router(cfg-call-home-profile)# subscribe-to-alert-group snapshot periodic daily 12:00`	Subscribes this destination profile to the Snapshot alert group. The Snapshot alert group can be configured for periodic notification. By default, the Snapshot alert group has no command to run. You can add commands into the alert group. In doing so, the output of the commands added in the Snapshot alert group will be included in the snapshot message.
Step 12	**subscribe-to-alert-group syslog** [**severity** { **catastrophic** | **disaster** | **fatal** | **critical** | **major** | **minor** | **warning** | **notification** | **normal** | **debugging** }] [**pattern** *string*] **Example:** `Router(cfg-call-home-profile)# subscribe-to-alert-group syslog severity major`	Subscribes this destination profile to the Syslog alert group. The Syslog alert group can be configured to filter messages based on severity. You can specify a pattern to be matched in the syslog message. If the pattern contains spaces, you must enclose it in quotes (""). You can specify a text pattern to be matched within each syslog message. If you configure a pattern, a Syslog alert group message is sent only if it contains the specified pattern and meets the severity threshold. If the pattern contains spaces, you must enclose it in quotes (“”). You can specify up to five patterns for each destination profile.
Step 13	**subscribe-to-alert-group all** **Example:**	(Optional) Subscribes to all available alert groups.

	Command or Action	Purpose
	`Router(cfg-call-home-profile)# subscribe-to-alert-group all`	**Important** Entering this command causes the generation of a large number of syslog messages. We recommend that you subscribe to alert groups individually, using appropriate severity levels and patterns when possible.

Periodic Notification

When you subscribe a destination profile to either the Configuration, Inventory, or Snapshot alert group, you can choose to receive the alert group messages asynchronously or periodically at a specified time. The following time intervals are available:

- Daily—Specify the time of day to send, using an hour:minute format *hh:mm*, with a 24-hour clock (for example, 14:30).
- Weekly—Specify the day of the week and time of day in the format day *hh:mm*, where the day of the week is spelled out (for example, monday).
- Monthly—Specify the numeric date, from 1 to 31, and the time of day, in the format date *hh:mm*.
- Interval—Specifies the interval at which the periodic message is sent, from 1 to 60 minutes.
- Hourly—Specifies the minute of the hour at which the periodic message is sent, from 0 to 59 minutes.

Note Hourly and by interval periodic notifications are available for the Snapshot alert group only.

Message Severity Threshold

Call Home allows you to filter messages based on severity. You can associate each predefined or user-defined destination profile with a Call Home threshold from 0 (least urgent) to 9 (most urgent). The default is 0 (all messages are sent).

When you subscribe a destination profile to the Environment or Syslog alert group, you can set a threshold for relay of alert group messages based on the message severity level. Any message with a value lower than the destination profile threshold is not sent to the destination.

When you subscribe to an alert group in a destination profile with a specified severity, you subscribe to getting messages triggered by the events that have same or higher severity in that alert group.

Note Subscribing to syslog message at low severity level is not recommended, as it would trigger too many syslog messages that might lower the system performance.

Note Call Home severity levels are not the same as system message logging severity levels.

Table 205: Severity and Syslog Level Mapping

Smart Call Home Level	Keyword	syslog Level	Description
9	**catastrophic**	—	Network-wide catastrophic failure.
8	**disaster**	—	Significant network impact.
7	**fatal**	Emergency (0)	System is unusable.
6	**critical**	Alert (1)	Critical conditions that indicate that immediate attention is needed.
5	**major**	Critical (2)	Major conditions.
4	**minor**	Error (3)	Minor conditions.
3	**warning**	Warning (4)	Warning conditions.
2	**notification**	Notice (5)	Basic notification and informational messages. Possibly independently insignificant.
1	**normal**	Information (6)	Normal event signifying return to normal state.
0	**debugging**	Debug (7)	Debugging messages.

Syslog Pattern Matching

When you subscribe a destination profile to the Syslog alert group, you can optionally specify a text pattern to be matched within each syslog message. If you configure a pattern, a Syslog alert group message will be sent only if it contains the specified pattern and meets the severity threshold. If the pattern contains spaces, you must enclose it in quotes (" ") when configuring it. You can specify up to five patterns for each destination profile.

Configuring Snapshot Command List

To configure the snapshot command list, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#` **`configure terminal`**	Enters global configuration mode.
Step 2	**call-home** **Example:**	Enters Call Home configuration mode.

	Command or Action	Purpose
	`Device(config)#` **`call-home`**	
Step 3	[**no** | **default**] **alert-group-config snapshot** **Example:** `Device(cfg-call-home)#` **`alert-group-config snapshot`**	Enters snapshot configuration mode. The **no** or **default** command will remove all snapshot command.
Step 4	[**no** | **default**] **add-command** *command string* **Example:** `Device(cfg-call-home-snapshot)#` **`add-command`** *`"show version"`*	Adds the command to the Snapshot alert group. The **no** or **default** command will remove the corresponding command. • *command string* —IOS command. Maximum length is 128.
Step 5	**end** **Example:** `Device(cfg-call-home-snapshot)#` **`exit`**	Exits and saves the configuration.

Configuring General email Options

Configuring the Mail Server

To use the e-mail message transport, you must configure at least one Simple Mail Transfer Protocol (SMTP) e-mail server address. You can specify up to four backup e-mail servers, for a maximum of five total mail-server definitions.

Consider the following guidelines when configuring the mail server:

- Backup e-mail servers can be defined by repeating the **mail-server** command using different priority numbers.
- The **mail-server priority** *number* parameter can be configured from 1 to 100. The server with the highest priority (lowest priority number) is tried first.

To configure general email options, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#` **`configure terminal`**	Enters global configuration mode.
Step 2	**call-home** **Example:** `Device(config)#` **`call-home`**	Enters call home configuration mode.
Step 3	**mail-server** { *ipv4-address* | *name* } **priority** *number* **Example:** `Device(cfg-call-home)#` **`mail-server stmp.example.com priority 1`**	Assigns an email server address and its relative priority among configured email servers. Provide either of these: • The email server's IP address or • The email server's fully qualified domain name (FQDN) of 64 characters or less. Assign a priority number between 1 (highest priority) and 100 (lowest priority).
Step 4	**sender from** *email-address* **Example:** `Device(cfg-call-home)#` **`sender from username@example.com`**	(Optional) Assigns the e-mail address that will appear in the from field in Call Home e-mail messages. If no address is specified, the contact e-mail address is used.
Step 5	**sender reply-to** *email-address* **Example:** `Device(cfg-call-home)#` **`sender reply-to username@example.com`**	(Optional) Assigns the e-mail address that will appear in the reply-to field in Call Home e-mail messages.
Step 6	**source-interface** *interface-name* **Example:** `Device(cfg-call-home)#` **`source-interface loopback1`**	Assigns the source interface name to send call-home messages. *interface-name* —Source interface name. Maximum length is 64.

	Command or Action	Purpose
		Note For HTTP messages, use the **ip http client source-interface** *interface-name* command in global configuration mode to configure the source interface name. This allows all HTTP clients on the device to use the same source interface.
Step 7	**source-ip-address** *ipv4/ipv6 address* **Example:** `Device(cfg-call-home)#` **`ip-address 209.165.200.226`**	Assigns source IP address to send call-home messages. • *ipv4/ipv6 address* —Source IP (ipv4 or ipv6) address. Maximum length is 64.
Step 8	**vrf** *vrf-name* **Example:** `Device(cfg-call-home)#` **`vrf`** *`vpn1`*	(Optional) Specifies the VRF instance to send call-home email messages. If no vrf is specified, the global routing table is used. **Note** For HTTP messages, if the source interface is associated with a VRF, use the **ip http client source-interface** *interface-name* command in global configuration mode to specify the VRF instance that will be used for all HTTP clients on the device.

Specifying Rate Limit for Sending Call Home Messages

To specify the rate limit for sending Call Home messages, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#` **`configure terminal`**	Enters global configuration mode.
Step 2	**call-home** **Example:** `Device(config)#` **`call-home`**	Enters call home configuration mode.

	Command or Action	Purpose
Step 3	**rate-limit** *number* **Example:** `Device(cfg-call-home)# rate-limit 40`	Specifies a limit on the number of messages sent per minute. • *number* —Range 1 to 60. The default is 20.

Specifying HTTP Proxy Server

To specify an HTTP proxy server for sending Call Home HTTP(S) messages to a destination, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**call-home** **Example:** `Device(config)# call-home`	Enters call home configuration mode.
Step 3	**http-proxy** {*ipv4-address* | *ipv6-address name*} *name* **Example:** `Device(config)# http-proxy 10.1.1.1 port 1`	Specifies the proxy server for the HTTP request.

Enabling AAA Authorization to Run IOS Commands for Call Home Messages

To enable AAA authorization to run IOS commands that enable the collection of output for a Call Home message, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**call-home** **Example:** `Device(config)#` **`call-home`**	Enters call home configuration mode.
Step 3	**aaa-authorization** **Example:** `Device(cfg-call-home)#` **`aaa-authorization`**	Enables AAA authorization. **Note** By default, AAA authorization is disabled for Call Home.
Step 4	**aaa-authorization** [**username** *username*] **Example:** `Device(cfg-call-home)#` **`aaa-authorization username`** *`username`*	Specifies the username for authorization. • **username** *user* —Default username is callhome. Maximum length is 64.

Configuring Syslog Throttling

To enable or disable Call Home syslog message throttling and avoid sending repetitive Call Home syslog messages, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#` **`configure terminal`**	Enters global configuration mode.
Step 2	**call-home** **Example:** `Device(config)#` **`call-home`**	Enters call home configuration mode.
Step 3	[**no**] **syslog-throttling** **Example:** `Device(cfg-call-home)#`	Enables or disables Call Home syslog message throttling and avoids sending repetitive Call Home syslog messages. By default, syslog message throttling is enabled.

	Command or Action	Purpose
	`syslog-throttling`	

Configuring Call Home Data Privacy

The **data-privacy** command scrubs data, such as passwords and IP addresses, from running configuration files to protect the privacy of customers. Enabling the **data-privacy** command can affect CPU utilization when scrubbing a large amount of data. Currently, **show** command output is not being scrubbed except for configuration messages in the **show running-config** all and show startup-config data.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#` **`configure terminal`**	Enters global configuration mode.
Step 2	**call-home** **Example:** `Device(config)#` **`call-home`**	Enters call home configuration mode.
Step 3	**data-privacy** { **level** {**normal** | **high** } | **hostname** } **Example:** `Device(cfg-call-home)#` **`data-privacy level high`**	Scrubs data from running configuration file to protect the privacy of the user. The default data-privacy level is normal. **Note** Enabling the data-privacy command can affect CPU utilization when scrubbing a large amount of data. • **normal** —Scrubs sensitive data such as passwords. • **high** —Scrubs all normal-level commands plus the IP domain name and IP address commands. • **hostname** —Scrubs all high-level commands plus the hostname command. **Note** Scrubbing the hostname from configuration messages can cause Smart Call Home processing failure on some platforms.

Sending Call Home Messages Manually

Sending a Call Home Test Message Manually

You can use the **call-home test** command to send a user-defined Call Home test message.

Procedure

	Command or Action	Purpose
Step 1	**call-home test** [**"** *test-message* **"**] **profile** *name* **Example:** `Router#` **`call-home test profile profile1`**	Sends a test message to the specified destination profile. The user-defined test message text is optional, but must be enclosed in quotes (" ") if it contains spaces. If no user-defined message is configured, a default message is sent.

Sending Call Home Alert Group Messages Manually

Before you begin

- Only the snapshot, crash, configuration, and inventory alert groups can be sent manually. Syslog alert groups cannot be sent manually.
- When you manually trigger a snapshot, configuration, or inventory alert group message and you specify a destination profile name, a message is sent to the destination profile regardless of the profile's active status, subscription status, or severity setting.
- When you manually trigger a snapshot, configuration, or inventory alert group message and do not specify a destination profile name, a message is sent to all active profiles that have either a normal or periodic subscription to the specified alert group.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router>` **`enable`**	Enables privileged EXEC mode. • Enter your password, if prompted.
Step 2	**call-home send alert-group snapshot** [**profile** *name*] **Example:** `Router#` **`call-home send alert-group snapshot profile profile1`**	Sends a snapshot alert group message to one destination profile if specified, or to all subscribed destination profiles.
Step 3	**call-home send alert-group crash** [**profile** *name*] **Example:**	Sends a crash alert group message to one destination profile if specified, or to all subscribed destination profiles.

	Command or Action	Purpose
	`Router# call-home send alert-group configuration profile profile1`	
Step 4	**call-home send alert-group configuration** [**profile** *name*] **Example:** `Router# call-home send alert-group configuration profile profile1`	Sends a configuration alert group message to one destination profile if specified, or to all subscribed destination profiles.
Step 5	**call-home send alert-group inventory** [**profile** *name*] **Example:** `Router# call-home send alert-group inventory`	Sends an inventory alert group message to one destination profile if specified, or to all subscribed destination profiles.

Submitting Call Home Analysis and Report Requests

You can use the **call-home request** command to submit information about your system to Cisco Systems to receive helpful analysis and report information specific to your system. You can request a variety of reports, including security alerts, known bugs, best practices, and command references.

Note the following guidelines when manually sending Call Home analysis and report requests:

- If a **profile** *name* is specified, the request is sent to the profile. If no profile is specified, the request is sent to the Cisco TAC profile. The recipient profile does not need to be enabled for the call-home request. The profile should specify the e-mail address where the transport gateway is configured so that the request message can be forwarded to the Cisco TAC and the user can receive the reply from the Smart Call Home service.
- The **ccoid** *user-id* is the registered identifier of the Smart Call Home user. If the *user-id* is specified, the response is sent to the e-mail address of the registered user. If no *user-id* is specified, the response is sent to the contact e-mail address of the device.
- Based on the keyword specifying the type of report requested, the following information is returned:
  - **config-sanity** —Information on best practices as related to the current running configuration.
  - **bugs-list** —Known bugs in the running version and in the currently applied features.
  - **command-reference** —Reference links to all commands in the running configuration.
  - **product-advisory** —Product Security Incident Response Team (PSIRT) notices, End of Life (EOL) or End of Sales (EOS) notices, or field notices (FN) that may affect devices in your network.

To submit a request for analysis and report information from the Cisco Output Interpreter tool, complete the following steps:

Procedure

	Command or Action	Purpose
Step 1	**call-home request output-analysis** " *show-command* " **Example:**	Sends the output of the specified **show** command for analysis. The **show** command must be contained in quotes ("").

	Command or Action	Purpose
	[**profile** *name*] [**ccoid** *user-id*] **Example:** Device# **call-home request output-analysis "show diag" profile TG**	
Step 2	**call-home request** { **config-sanity** | **bugs-list** | **command-reference** | **product-advisory** } **Example:** [**profile** *name*] [**ccoid** *user-id*] **Example:** Device# **call-home request config-sanity profile TG**	Sends the output of a predetermined set of commands, such as the **show running-config all** and **show version** commands, for analysis. In addition, the **call home request product-advisory** subcommand includes all inventory alert group commands. The keyword specified after the **call-home request** command specifies the type of report requested.

Manually Sending Command Output Message for One Command or a Command List

You can use the **call-home send** command to execute a CLI command and e-mail the command output to Cisco or to an e-mail address that you specify.

Note the following guidelines when sending the output of a command:

- The specified IOS command or list of IOS commands can be any run command, including commands for all modules. The command must be contained in quotes (“”).

- If the email option is selected using the "email" keyword and an email address is specified, the command output is sent to that address. If neither the email nor the HTTP option is specified, the output is sent in long-text format with the specified service request number to the Cisco TAC (attach@cisco.com).
- If neither the "email" nor the "http" keyword is specified, the service request number is required for both long-text and XML message formats and is provided in the subject line of the email.
- If the HTTP option is specified without a profile name or destination URL, the CiscoTac-1 profile destination HTTP or HTTPS URL is used as the destination. The destination email address can be specified so that Smart Call Home can forward the message to the email address. The user must specify either the destination email address or an SR number but they can also specify both.
- If a profile is specified and the profile has callhome@cisco.com as one of its email destination, you must use XML as the message format. If you use long-text format, an error message is displayed.

To execute a command and send the command output, complete the following step:

Procedure

	Command or Action	Purpose
Step 1	Use one of the following commands: • **call-home send** {*cli command* | *cli list*} [**email** [**profile** *profile-name* | *email*] [**msg-format** {**long-text** | **xml**}]] [**tac-sevice-request** *SR#*] • **call-home send** {*cli command* | *cli list*} [**http** [**profile** *profile-name* | *URL-dest*] [**destination-email-address** *email*]] [**tac-sevice-request** *SR#*] **Example:** `Router# call-home send "show version;show running-config show inventory" email support@example.com msg-format xml`	Executes the CLI or CLI list and sends output via email or HTTP. • {*cli command* | *cli list*}—Specifies the IOS command or list of IOS commands (separated by ';'). It can be any run command, including commands for all modules. The commands must be contained in quotes (""). • **email** [**profile** *profile-name* | *email*] [**msg-format** {**long-text** | **xml**}]—If the email option is selected and a profile name is specified, the command output will be sent to the email address configured in the profile. If an email address is specified, the command output will be sent to the specified email address. The message is in long-text or XML format with the service request number in the subject. The profile name or email address, the service request number, or both must be specified. The service request number is required if the profile name or email address is not specified (default is attach@cisco.com for long-text format and callhome@cisco.com for XML format). • **http** [**profile** *profile-name* | *URL-dest*] [**destination-email-address** *email*] —If the http option is selected without a profile name or destination URL, the command output will be sent to Smart Call Home backend server (URL specified

	Command or Action	Purpose
		in TAC profile) in XML format. If a profile name or destination URL is specified, the command output will be sent to the destination URLs configured in the profile (profile-name case) or the destination URL specified in the command. **destination-email-address** *email* can be specified so that the backend server can forward the message to the email address. The email address, the service request number, or both must be specified. • **tac-service-request** *SR#*—Specifies the service request number. The service request number is required if the email address is not specified.

Configuring Diagnostic Signatures

The Diagnostic Signatures feature downloads digitally signed signatures to devices. Diagnostic Signatures (DS) files are formatted files that collate knowledge of diagnostic events and provide methods to troubleshoot them without a need to upgrade the Cisco software. The aim of DS is to deliver flexible intelligence that can detect and collect troubleshooting information that can be used to resolve known problems in customer networks.

Prerequisites for Diagnostic Signatures

Before you download and configure diagnostic signatures (DSes) on a device, you must ensure that the following conditions are met:

- You must assign a DS to the device. Refer to the “Diagnostic Signature Downloading” section for more information on how to assign DSes to devices.
- HTTP/Secure HTTP (HTTPS) transport is required for downloading DS files. You must install the certification authority (CA) certificate to enable the authentication of the destination HTTPS server.

Note If you configure the trustpool feature, the CA certificate is not required.

Information About Diagnostic Signatures

Diagnostic Signature Overview

Diagnostic signatures (DS) for the Call Home system provides a flexible framework that allows the defining of new events and corresponding CLIs that can analyze these events without upgrading the Cisco software.

DSs provide the ability to define more types of events and trigger types than the standard Call Home feature supports. The DS subsystem downloads and processes files on a device as well as handles callbacks for diagnostic signature events.

The Diagnostic Signature feature downloads digitally signed signatures that are in the form of files to devices. DS files are formatted files that collate the knowledge of diagnostic events and provide methods to troubleshoot these events.

DS files contain XML data to specify the event description, and these files include CLI commands or scripts to perform required actions. These files are digitally signed by Cisco or a third party to certify their integrity, reliability, and security.

The structure of a DS file can be one of the following formats

- Metadata-based simple signature that specifies the event type and contains other information that can be used to match the event and perform actions such as collecting information by using the CLI. The signature can also change configurations on the device as a workaround for certain bugs.
- Embedded Event Manager (EEM) Tool Command Language (Tcl) script-based signature that specifies new events in the event register line and additional action in the Tcl script.
- Combination of both the formats above.

The following basic information is contained in a DS file:

- ID (unique number)—unique key that represents a DS file that can be used to search a DS.
- Name (ShortDescription)—unique description of the DS file that can be used in lists for selection.
- Description—long description about the signature.
- Revision—version number, which increments when the DS content is updated.
- Event & Action—defines the event to be detected and the action to be performed after the event happens.

Diagnostic Signature Downloading

To download the diagnostic signature (DS) file, you require the secure HTTP (HTTPS) protocol. If you have already configured an email transport method to download files on your device, you must change your assigned profile transport method to HTTPS to download and use DS.

Cisco software uses a PKI Trustpool Management feature, which is enabled by default on devices, to create a scheme to provision, store, and manage a pool of certificates from known certification authorities (CAs). The trustpool feature installs the CA certificate automatically. The CA certificate is required for the authentication of the destination HTTPS servers.

There are two types of DS update requests to download DS files: regular and forced-download.

Regular download requests DS files that were recently updated. You can trigger a regular download request either by using a periodic configuration or by initiating an on-demand CLI. The regular download update

happens only when the version of the requested DS is different from the version of the DS on the device. Periodic download is only started after there is any DS assigned to the device from DS web portal. After the assignment happens, the response to the periodic inventory message from the same device will include a field to notify device to start its periodic DS download/update. In a DS update request message, the status and revision number of the DS is included such that only a DS with the latest revision number is downloaded.

Forced-download downloads a specific DS or a set of DSes. You can trigger the forced-download update request only by initiating an on-demand CLI. In a force-download update request, the latest version of the DS file is downloaded irrespective of the current DS file version on the device.

The DS file is digitally signed, and signature verification is performed on every downloaded DS file to make sure it is from a trusted source.

Diagnostic Signature Signing

The diagnostic signature (DS) files are digitally signed before they are made available for downloading. The following methods are used for digitally signing DS files:

- Signing algorithm (Rivest Shamir and Adleman [RSA] 2048 bits)
- Request keypairs to Abraxas system, which is the digital signing client
- DS signed via secure socket layer (SSL) through a code signing client, where the signature is embedded using XML tags
- Public keys are embedded in the DS subsystem (Cisco signed, partner signed, third-party signed) in the Cisco software. The digitally signed DS file contains the product name such as Diagnostic_Signatures (Cisco signed), Diagnostic_Signatures_Partner, Diagnostic_Signatures_3rd_Party. The product names are only used to sign the DS files.

The digital signing client can be found at https://abraxas.cisco.com/SignEngine/submit.jsp

These conditions that must be met to verify the digital signature in a DS file:

- Code sign component support must be available in Cisco software.
- Various public keys that verify the different kinds of diagnostic signatures must be included in platforms where DS is supported.
- After parsing and retrieving the DS, the DS must execute the verification application program interface (API) to verify that the DS is valid.

Diagnostic Signature Workflow

The diagnostic signature feature is enabled by default in Cisco software. The following is the workflow for creating diagnostic signatures:

1. Find the DSs you want to download and assign them to the device. This step is mandatory for regular periodic download, but not required for forced download.
2. The device downloads all assigned DS(es) or a specific DS by regular periodic download or by on-demand forced download.
3. The device verifies the digital signature of every single DS. If verification passes, the device stores the DS file into a non-removable disk, such as bootflash or hard disk, so that DS files can be read after the device is reloaded. On the routers, the DS file is stored in the bootflash:/call home directory.

4. The device continues sending periodic regular DS download requests to get the latest revision of DS and replace the older one in device.
5. The device monitors the event and executes the actions defined in the DS when the event happens.

Diagnostic Signature Events and Actions

The events and actions sections are the key areas used in diagnostic signatures. The event section defines all event attributes that are used for event detection. The action section lists all actions which should be performed after the event happens, such as collecting **show** command outputs and sending them to Smart Call Home to parse.

Diagnostic Signature Event Detection

Event detection in a DS is defined in two ways: single event detection and multiple event detection.

Single Event Detection

In single event detection, only one event detector is defined within a DS. The event specification format is one of the following two types:

- DS event specification type: syslog, periodic, configuration, Online Insertion Removal (OIR) immediate, and callhome are the supported event types, where “immediate” indicates that this type of DS does not detect any events, its actions are performed once it is downloaded, and the call-home type modifies the current CLI commands defined for existing alert-group.
- The Embedded Event Manager (EEM) specification type: supports any new EEM event detector without having to modify the Cisco software.

  Other than using EEM to detect events, a DS is triggered when a Tool Command Language (Tcl) script is used to specify event detection types.

Multiple Event Detection

Multiple event detection involves defining two or more event detectors, two ore more corresponding tracked object states, and a time period for the events to occur. The specification format for multiple event detection can include complex event correlation for tracked event detectors. For example, three event detectors (syslog, OIR, and IPSLA) are defined during the creation of a DS file. The correlation that is specified for these event detectors is that the DS will execute its action if both syslog and OIR events are triggered simultaneously, or if IPSLA is triggered alone.

Diagnostic Signature Actions

The diagnostic signature (DS) file consists of various actions that must be initiated when an evnt occurs. The action type indicates the kind of action that will be initiated in response to a certain event.

Variables are elements within a DS file that are used to customize the files.

DS actions are categorized into the following five types:

- call-home
- command
- emailto
- script
- message

DS action types call-home and emailto collect event data and send a message to call-home servers or to the defined email addresses. The message uses "diagnostic-signature" as its message type and DS ID as the message sub-type.

The commands defined for the DS action type initiate CLI commands that can change configuration of the device, collect show command outputs, or run any EXEC command on the device. The DS action type script executes Tcl scripts.

DS action type message defines action to generate message to notify or remind user certain important information. The message could be broadcasted to all TTY lines or generated as a syslog entry.

Action Types

DS actions are categorized into the following four types:

- Call-home
- Command
- Emailto
- Script

DS action types call-home and emailto collect event data and send a message to call-home servers or to the defined email addresses. The message includes the following elements:

- Message type—diagnostic-signature
- Message subtype—ds-id
- Message description—event-id : ds name

The commands defined for the DS action type initiates CLI commands that can change configuration of the device. The DS action type script executes Tcl scripts.

Diagnostic Signature Variables

Variables are referenced within a DS and are used to customize the DS file. All DS variable names have the prefix ds_ to separate them from other variables. The following are the supported DS variable types:

- System variable: variables assigned automatically by the device without any configuration changes. The Diagnostic Signatures feature supports two system variables: ds_hostname and ds_signature_id.
- Environment variable: values assigned manually by using the **environment** *variable-name variable-value* command in call-home diagnostic-signature configuration mode. Use the **show call-home diagnostic-signature** command to display the name and value of all DS environment variables. If the DS file contains unresolved environment variables, this DS will stay in pending status until the variable gets resolved.
- Prompt variable: values assigned manually by using the **call-home diagnostic-signature install** *ds-id* command in privileged EXEC mode. If you do not set this value, the status of the DS indicates pending.
- Regular expression variable: values assigned from a regular expression pattern match with predefined CLI command outputs. The value is assigned during the DS run.
- Syslog event variable: values assigned during a syslog event detection in the DS file. This variable is valid only for syslog event detection.

How to Configure Diagnostic Signatures

Configuring the Service Call Home for Diagnostic Signatures

Configure the Service Call Home feature to set attributes such as the contact email address where notifications related with diagnostic signatures (DS) are sent and destination HTTP/secure HTTP (HTTPS) URL to download the DS files from.

You can also create a new user profile, configure correct attributes and assign it as the DS profile. For periodic downloads, the request is sent out just following full inventory message. By changing the inventory periodic configuration, the DS periodic download also gets rescheduled.

Note The predefined CiscoTAC-1 profile is enabled as a DS profile by default and Cisco recommends using it. If used, you only need to change the destination transport-method to the **http** setting.

Before you begin

Before you download and configure diagnostic signatures (DSs) on a device, you must ensure that the following conditions are met:

- You must assign one or more DSs to the device.
- HTTP/Secure HTTP (HTTPS) transport is required for downloading DS files. You must install the certification authority (CA) certificate to enable the authentication of the destination HTTPS server.

Note If you configure the trustpool feature, the CA certificate is not required.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**service call-home** **Example:** `Router(config)# service call-home`	Enables Call Home service on a device.
Step 4	**call-home** **Example:**	Enters Call Home configuration mode.

	Command or Action	Purpose
	`Router(config)# call-home`	
Step 5	**contact-email-addr** *email-address* **Example:** `Router(cfg-call-home)# contact-email-addr username@example.com`	Assigns customer's e-mail address. You can enter a maximum of 200 characters in e-mail address format with no spaces. **Note** You can use any valid e-mail address. You cannot use spaces.
Step 6	**mail-server** { *ipv4-address* | *name* } **priority** *number* **Example:** `Router(cfg-call-home)# mail-server 10.1.1.1 priority 4`	(Optional) Configures a Simple Mail Transfer Protocol (SMTP) email server address for Call Home. This command is only used when sending email is part of the actions defined in any DS.
Step 7	**profile** *name* **Example:** `Router(cfg-call-home)# profile profile1`	Enters the Call Home destination profile configuration mode for the specified destination profile. If the specified destination profile does not exist, it is created.
Step 8	**destination transport-method** { **email** | **http** } **Example:** `Router(cfg-call-home-profile)# destination transport-method email`	(Optional) Enables the message transport method. • **email** —Sets the e-mail message transport method. • **http** —Sets the HTTP message transport method. **Note** To configure diagnostic signatures, you must use the **http** option.
Step 9	**destination address** { **email** *email-address* | **http** *url* } **Example:** `Router(cfg-call-home-profile)# destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService`	Configures the destination e-mail address or URL to which Call Home messages are sent. **Note** To configure diagnostic signatures, you must use the **http** option.
Step 10	**subscribe-to-alert-group inventory** [**periodic** { **daily** *hh:mm* | **monthly** *date hh:mm* | **weekly** *day hh:mm* }] **Example:** `Router(cfg-call-home-profile)# subscribe-to-alert-group inventory periodic daily 12:00`	Subscribes this destination profile to the Inventory alert group. The Inventory alert group can be configured for periodic notification **Note** This command is used only for the periodic downloading of DS files.

What to do next

Set the profile configured in the previous procedure as the DS profile and configure other DS parameters.

Configuring Diagnostic Signatures

Before you begin

Configure the Service Call Home feature to set attributes for the Call Home profile. You can either use the default CiscoTAC-1 profile or use the newly-created user profile.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**call-home** **Example:** `Router(config)# call-home`	Enters Call Home configuration mode.
Step 4	**diagnostic-signature** **Example:** `Router(cfg-call-home)# diagnostic-signature`	Enters call-home diagnostic signature mode.
Step 5	**profile** *ds-profile-name* **Example:** `Router(cfg-call-home-diag-sign)# profile user1`	Specifies the destination profile on a device that DS uses.
Step 6	**environment** *ds_env-var-name ds-env-var-value* **Example:** `Router(cfg-call-home-diag-sign)# environment ds_env1 envarval`	Sets the environment variable value for DS on a device.
Step 7	**end** **Example:** `Router(cfg-call-home-diag-sign)# end`	Exits call-home diagnostic signature mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 8	**call-home diagnostic-signature** { { **deinstall** | **download** } { *ds-id* | **all** } | **install** *ds-id* } **Example:** `Router# call-home diagnostic-signature download 6030`	Downloads, installs, and uninstalls diagnostic signature files on a device.
Step 9	**show call-home diagnostic-signature** [*ds-id* [**actions** | **events** | **prerequisite** | **prompt** | **variables**] | **failure** | **statistics** [download]] **Example:** `Router# show call-home diagnostic-signature actions`	Displays the call-home diagnostic signature information.

Verifying the Call Home Configuration

- **show call-home** —Displays the Call Home configuration summary.

  Following is a sample output of the command:

```
Router# show call-home

Current call home settings:
    call home feature : enable
    call home message's from address: Not yet set up
    call home message's reply-to address: Not yet set up

    vrf for call-home messages: Not yet set up

    contact person's email address: sch-smart-licensing@cisco.com (default)

    contact person's phone number: Not yet set up
    street address: Not yet set up
    customer ID: Not yet set up
    contract ID: Not yet set up
    site ID: Not yet set up

    source ip address: Not yet set up
    source interface: TenGigabitEthernet4/1/1
    Mail-server[1]: Address: 173.36.13.143 Priority: 60
    http proxy: Not yet set up

    Diagnostic signature: enabled
    Profile: CiscoTAC-1 (status: ACTIVE)

    Smart licensing messages: enabled
    Profile: CiscoTAC-1 (status: ACTIVE)

    aaa-authorization: disable
    aaa-authorization username: callhome (default)
    data-privacy: normal
    syslog throttling: enable
```

```
    Rate-limit: 20 message(s) per minute

    Snapshot command[0]: show cable modem summary totalb
    Snapshot command[1]: show cable modem summary total

Available alert groups:
    Keyword                  State   Description
    ------------------------ ------- --------------------------------
    configuration            Enable  configuration info
    crash                    Enable  crash and traceback info
    diagnostic               Enable  diagnostic info
    environment              Enable  environmental info
    inventory                Enable  inventory info
    snapshot                 Enable  snapshot info
    syslog                   Enable  syslog info

Profiles:
    Profile Name: CiscoTAC-1
    Profile Name: test
```

- **show call-home detail**—Displays the Call Home configuration in detail.

  Following is a sample output of the command:

```
Router# show call-home detail

Current call home settings:
    call home feature : enable
    call home message's from address: Not yet set up
    call home message's reply-to address: Not yet set up

    vrf for call-home messages: Not yet set up

    contact person's email address: sch-smart-licensing@cisco.com (default)

    contact person's phone number: Not yet set up
    street address: Not yet set up
    customer ID: Not yet set up
    contract ID: Not yet set up
    site ID: Not yet set up

    source ip address: Not yet set up
    source interface: TenGigabitEthernet4/1/1
    Mail-server[1]: Address: 173.36.13.143 Priority: 60
    http proxy: Not yet set up

    Diagnostic signature: enabled
    Profile: CiscoTAC-1 (status: ACTIVE)

    Smart licensing messages: enabled
    Profile: CiscoTAC-1 (status: ACTIVE)

    aaa-authorization: disable
    aaa-authorization username: callhome (default)
    data-privacy: normal
    syslog throttling: enable

    Rate-limit: 20 message(s) per minute

    Snapshot command[0]: show cable modem summary totalb
    Snapshot command[1]: show cable modem summary total

Available alert groups:
    Keyword                  State   Description
```

```
    ------------------------ ------- --------------------------------
    configuration            Enable  configuration info
    crash                    Enable  crash and traceback info
    diagnostic               Enable  diagnostic info
    environment              Enable  environmental info
    inventory                Enable  inventory info
    snapshot                 Enable  snapshot info
    syslog                   Enable  syslog info

Profiles:

Profile Name: CiscoTAC-1
    Profile status: ACTIVE
    Profile mode: Anonymous Reporting Only
    Reporting Data: Smart Call Home, Smart Licensing
    Preferred Message Format: xml
    Message Size Limit: 3145728 Bytes
    Transport Method: http
    Email address(es): callhome@cisco.com
    HTTP  address(es): https://tools.cisco.com/its/service/oddce/services/DDCEService

    Periodic configuration info message is scheduled every 17 day of the month at 09:39


    Periodic inventory info message is scheduled every 17 day of the month at 09:24

    Alert-group              Severity
    -----------------------  ------------
    crash                    debug
    diagnostic               minor
    environment              minor
    inventory                normal

    Syslog-Pattern           Severity
    -----------------------  ------------
    .*                       major
```

- **show call-home alert-group** —Displays the available alert groups and their status.

  Following is a sample output of the command:

```
Router# show call-home alert-group

Available alert groups:
    Keyword                  State   Description
    ------------------------ ------- --------------------------------
    configuration            Enable  configuration info
    crash                    Enable  crash and traceback info
    diagnostic               Enable  diagnostic info
    environment              Enable  environmental info
    inventory                Enable  inventory info
    snapshot                 Enable  snapshot info
    syslog                   Enable  syslog info
```

- **show call-home mail-server status**—Checks and displays the availability of the configured email servers.

  Following is a sample output of the command:

```
Router# show call-home mail-server status

    Mail-server[1]: Address: 173.36.13.143 Priority: 60
```

- **show call-home profile** {**all** | *name*} —Displays the configuration of the specified destination profile. Use the keyword **all** to display the configuration of all destination profiles.

  Following is a sample output of the command:

```
Router# show call-home profile CiscoTac-1

Profile Name: CiscoTAC-1
    Profile status: ACTIVE
    Profile mode: Full Reporting
    Reporting Data: Smart Call Home, Smart Licensing
    Preferred Message Format: xml
    Message Size Limit: 3145728 Bytes
    Transport Method: email
    Email address(es): callhome@cisco.com
    HTTP  address(es): http://10.22.183.117:8080/ddce/services/DDCEService

    Periodic configuration info message is scheduled every 17 day of the month at 09:39


    Periodic inventory info message is scheduled every 17 day of the month at 09:24

    Alert-group               Severity
    ------------------------  ------------
    crash                     debug
    diagnostic                minor
    environment               minor
    inventory                 normal

    Syslog-Pattern            Severity
    ------------------------  ------------
    .*                        major
```

- **show call-home statistics** [**detail** | **profile** *profile-name*]—Displays the statistics of Call Home events.

  Following is a sample output of the command:

```
Router# show call-home statistics

Message Types    Total                Email                HTTP
-------------   -------------------- -------------------- ------------------
Total Success   4                    3                    1
    Config      1                    1                    0
    Crash       0                    0                    0
    Diagnostic  0                    0                    0
    Environment 0                    0                    0
    Inventory   1                    0                    1
    Snapshot    0                    0                    0
    SysLog      2                    2                    0
    Test        0                    0                    0
    Request     0                    0                    0
    Send-CLI    0                    0                    0
    SCH         0                    0                    0

Total In-Queue  0                    0                    0
    Config      0                    0                    0
    Crash       0                    0                    0
    Diagnostic  0                    0                    0
    Environment 0                    0                    0
    Inventory   0                    0                    0
    Snapshot    0                    0                    0
    SysLog      0                    0                    0
```

```
    Test        0                   0                   0
    Request     0                   0                   0
    Send-CLI    0                   0                   0
    SCH         0                   0                   0

Total Failed    0                   0                   0
    Config      0                   0                   0
    Crash       0                   0                   0
    Diagnostic  0                   0                   0
    Environment 0                   0                   0
    Inventory   0                   0                   0
    Snapshot    0                   0                   0
    SysLog      0                   0                   0
    Test        0                   0                   0
    Request     0                   0                   0
    Send-CLI    0                   0                   0
    SCH         0                   0                   0

Total Ratelimit
      -dropped  0                   0                   0
    Config      0                   0                   0
    Crash       0                   0                   0
    Diagnostic  0                   0                   0
    Environment 0                   0                   0
    Inventory   0                   0                   0
    Snapshot    0                   0                   0
    SysLog      0                   0                   0
    Test        0                   0                   0
    Request     0                   0                   0
    Send-CLI    0                   0                   0
    SCH         0                   0                   0

Last call-home message sent time: 2015-03-06 18:21:49 GMT+00:00
```

- **show call-home diagnostic-signature**—Displays the configuration of diagnostic signature information.

  Following is a sample output of the command:

```
Router# show call-home diagnostic-signature

Current diagnostic-signature settings:
 Diagnostic-signature: enabled
 Profile: CiscoTAC-1 (status: ACTIVE)
 Environment variable:
           Not yet set up

Downloaded DSes:
                                                                  Last Update
DS ID    DS Name                          Revision Status     (GMT-05:00)
-------- -------------------------------- -------- ---------- -------------------
```

- **show call-home version**—Displays the Call Home version information.

  Following is a sample output of the command:

```
Router# show call-home version

Call-Home Version 3.0
Component Version:
 call-home: (rel4)1.0.15
 eem-call-home: (rel2)1.0.5
```

Configuration Example for Call Home

Example: Call Home Configuration

Following is a configuration example for configuring the HTTPS transport :

```
ip host tools.cisco.com 72.163.4.38
vrf definition smart-vrf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
interface TenGigabitEthernet4/1/1
 vrf forwarding smart-vrf
 ip address 172.22.11.25 255.255.255.128
 no ip proxy-arp
!
ip route vrf smart-vrf 72.163.4.38 255.255.255.255 172.22.11.1
!
ip http client source-interface TenGigabitEthernet4/1/1
!
```

Following is a configuration example for configuring email options:

```
call-home
 mail-server 173.36.13.143 priority 60
 source-interface TenGigabitEthernet4/1/1
 vrf smart-vrf
 alert-group-config snapshot
  add-command "show cable modem summary total"
 profile "test"
  active
  destination transport-method email
  destination address email call-home@cisco.com
  subscribe-to-alert-group configuration
  subscribe-to-alert-group crash
  subscribe-to-alert-group diagnostic severity debug
  subscribe-to-alert-group environment severity debug
  subscribe-to-alert-group inventory
  subscribe-to-alert-group syslog severity major pattern .*
  subscribe-to-alert-group syslog severity notification pattern "^.+UPDOWN.+changed state
to (down|up)$"
  subscribe-to-alert-group snapshot periodic daily 12:00
!
ip route vrf smart-vrf 173.36.13.143 255.255.255.255 172.22.11.1
!
```

Example: Configuring HTTP Transport for Call Home on the Cisco cBR Series Router

Procedure

Step 1 Back up the current running configuration file.

Step 2 Verify the built-in router certificates.

Example:

```
Router# show crypto pki trustpool | include Class 3 Public

    ou=Class 3 Public Primary Certification Authority
    ou=Class 3 Public Primary Certification Authority
```

Step 3 (Optional) Configure VRF.

Example:

```
Router(config)# vrf def smart-vrf
Router(config-vrf)# address-family ipv4
Router(config-vrf-af)# exit-address-family
Router(config-vrf)# address-family ipv6
Router(config-vrf-af)# exit-address-family
```

Step 4 Set up the network interface.

Example:

```
Router(config)# interface TenGigabitEthernet4/1/1
Router(config)# vrf forward smart-vrf
Router(config-if)# ip address 172.22.11.25 255.255.255.128
Router(config-if)# no ip proxy-arp
Router(config-if)# no shut
```

Note If IPv6 is enabled, you must configure IPv6 address.

Step 5 Set up the Cisco portal.

Example:

```
Router(config)# ip host tools.cisco.com 72.163.4.38
Router(config)# ip route vrf smart-vrf 72.163.4.38 255.255.255.255 172.22.11.1
```

Step 6 Verify the data path.

Example:

```
!Verify the connectivity to TenGigabitEthernet4/1/1 interface
Router# ping vrf smart-vrf 172.22.11.25

!Verify the connectivity to TenGigabitEthernet4//1/1 gateway
Router# ping vrf smart-vrf 172.22.11.1

!Verify the connectivity to tools.cisco.com
Router# ping vrf smart-vrf 72.163.4.38
```

Step 7 Configure the HTTP client interface.

Example:

```
Router(config)# ip http client source-interface TenGigabitEthernet4/1/1
```

Step 8 Send the Call Home alert group message manually and verify the configuration.

Example:

```
Router# call-home send alert inventory profile CiscoTAC-1

Sending inventory info call-home message ...
Please wait. This may take some time ...

Router# show call-home statistics | include Total
Message Types    Total               Email               HTTP
Total Success   0                   0                   0
Total In-Queue  1                   0                   1
Total Failed    0                   0                   0
Total Ratelimit

Router# show call-home statistics | include Total
Message Types    Total               Email               HTTP
Total Success   1                   0                   1
Total In-Queue  0                   0                   0
Total Failed    0                   0                   0
Total Ratelimit
```

Step 9 Display the Call Home configuration.

Example:

```
Router# show call-home profile CiscoTAC-1

Profile Name: CiscoTAC-1
    Profile status: ACTIVE
    Profile mode: Full Reporting
    Reporting Data: Smart Call Home, Smart Licensing
    Preferred Message Format: xml
    Message Size Limit: 3145728 Bytes
    Transport Method: http
    Email address(es): callhome@cisco.com
    HTTP  address(es): https://tools.cisco.com/its/service/oddce/services/DDCEService

    Periodic configuration info message is scheduled every 15 day of the month at 15:37

    Periodic inventory info message is scheduled every 15 day of the month at 15:22
Alert-group              Severity
    ------------------------  ------------
    crash                     debug
    diagnostic                minor
    environment               minor
    inventory                 normal

    Syslog-Pattern            Severity
    ------------------------  ------------
    .*                        major
```

Example: Configuring Email Transport for Call Home on the Cisco cBR Series Router

Procedure

Step 1 Back up the current running configuration file.

Step 2 (Optional) Configure VRF.

Example:

```
Router(config)# vrf def smart-vrf
Router(config-vrf)# address-family ipv4
Router(config-vrf-af)# exit-address-family
Router(config-vrf)# address-family ipv6
Router(config-vrf-af)# exit-address-family
```

Step 3 Set up the network interface.

Example:

```
Router(config)# interface TenGigabitEthernet4/1/1
Router(config)# vrf forward smart-vrf
Router(config-if)# ip address 172.22.11.25 255.255.255.128
Router(config-if)# no ip proxy-arp
Router(config-if)# no shut
```

Note If IPv6 is enabled, you must configure IPv6 address.

Step 4 Verify the data path.

Example:

```
!Verify the connectivity to TenGigabitEthernet4/1/1 interface
Router# ping vrf smart-vrf 172.22.11.25

!Verify the connectivity to TenGigabitEthernet4//1/1 gateway
Router# ping vrf smart-vrf 172.22.11.1

!Verify the connectivity to tools.cisco.com
Router# ping vrf smart-vrf 72.163.4.38
```

Step 5 (Optional) Configure Call Home.

Example:

```
Router(config)# call-home

!Configure the TenGigabitEthernet 4/1/1
Router(cfg-call-home)# source-ip-address 172.22.11.25
```

Step 6 Configure the mail server and verify the configuration.

Example:

```
Router(config)# call-home
Router(cfg-call-home)# mail-server 173.36.13.143 priority 60
Router(cfg-call-home)# vrf smart-vrf
```

```
Router(cfg-call-home)# exit
Router(config)# ip route vrf smart-vrf 173.36.13.143 255.255.255.255 172.22.11.1
Router(config)# end

Router# ping vrf smart-vrf 173.36.13.143
…

Router# show call-home mail status

Please wait. Checking for mail server status ...

Mail-server[1]: Address: 173.36.13.143 Priority: 60 [Available]
```

Note The VRF configuration is optional.

Step 7 Create a new destination profile and subscribe to alert group.

Example:

```
Router(config)# call-home
Router(cfg-call-home)# alert-group-config snapshot
Router(cfg-call-home-snapshot)# add-command "show cable modem summary total”
Router(cfg-call-home-snapshot)# exit
Router(cfg-call-home)# profile test
Router(cfg-call-home-profile)# active
Router(cfg-call-home-profile)# destination transport-method email
Router(cfg-call-home-profile)# destination address email xyz@company.com
Router(cfg-call-home-profile)# subscribe syslog severity notification pattern
"^.+UPDOWN.+changed state to (down|up)$”
Router(cfg-call-home-profile)# subscribe-to-alert-group snapshot periodic daily 12:00
Router(cfg-call-home-profile)# end
```

Step 8 Send the Call Home alert group message manually and verify the configuration.

Example:

```
Router# call-home send alert-group inventory profile test
Sending inventory info call-home message ...
Please wait. This may take some time ...

Router# show call-home statistics | include Total
Message Types    Total               Email               HTTP
Total Success   1                   0                   1
Total In-Queue  2                   2                   0
Total Failed    0                   0                   0
Total Ratelimit

Router# show call-home statistics | include Total
Message Types    Total               Email               HTTP
Total Success   3                   2                   1
Total In-Queue  0                   0                   0
Total Failed    0                   0                   0
Total Ratelimit
```

Step 9 Display the Call Home configuration.

Example:

```
Router# show call-home profile test

Profile Name: test
    Profile status: ACTIVE
```

```
    Profile mode: Full Reporting
    Reporting Data: Smart Call Home
    Preferred Message Format: xml
    Message Size Limit: 3145728 Bytes
    Transport Method: email
    Email address(es): abcd@company.com
    HTTP  address(es): Not yet set up

    Periodic snapshot info message is scheduled daily at 12:00

    Alert-group               Severity
    ------------------------  ------------
    configuration             normal
    crash                     debug
    diagnostic                debug
    environment               debug
    inventory                 normal
Syslog-Pattern            Severity
    ------------------------  ------------
    ^.+UPDOWN.+changed state
     to (down|up)$            notification
```

Default Settings

Table 206: Default Call Home Parameters

Parameters	Default
Call Home feature status	Enabled
User-defined profile status	Active
Predefined CiscoTAC-1 profile status	Active
Transport method	HTTP
Message format type	XML
Alert group status	Enabled
Call Home message severity threshold	Debug
Message rate limit for messages per minute	20
AAA authorization	Disabled
Call Home syslog message throttling	Enabled
Data privacy level	Normal

Alert Groups Trigger Events and Commands

The table below lists the supported alert groups and the default command output included in Call Home messages generated for the alert group.

Table 207: Call Home Alert Groups, Events, and Actions

Alert Group	Call Home Trigger Event	Syslog Event	Severity	Description and Executed Commands
Configuration	—	—	**normal** **periodic**	Periodic events related to configuration sent monthly. Commands executed: • **show platform** • **show version** • **show inventory** • **show running-config all** • **show startup-config**

Alert Group	Call Home Trigger Event	Syslog Event	Severity	Description and Executed Commands
Crash	—	—	**debug**	Events generated by a crash on the router, such as Supervisor or line card crash. Commands executed: Crash traceback • **show version** • **show logging** • **show region** • **show stack** Crash system • **show version** • **show inventory** • **show logging** • **show region** • **show stack** • **more crashinfo-file** Crash module • **show version** • **show inventory** • **show platform** • **show logging** • **show region** • **show stack** • **more crashinfo-file**

Alert Group	Call Home Trigger Event	Syslog Event	Severity	Description and Executed Commands
Diagnostic	—	—	**minor**	Events generated by diagnostics. Commands executed: • **show platform** • **show version** • **show diagnostic event slot detail** • **show inventory** • **show buffers** • **show logging** • **show diagnostic events slot all**
Environmental	FAN_FAILURE	CBR_PEM-6-FANOK CBR_PEM-3-FANFAIL	**minor**	Events related to power, fan, and environment sensing elements, such as temperature alarms. Commands executed: • **show platform** • **show environment** • **show inventory** • **show logging**
	TEMPERATURE_ALARM	ENVIRONMENTAL-1-ALERT		
	POWER_SUPPLY_FAILURE	CBR_PEM-6-PEMOK CBR_PEM-3-PEMFAIL		

Alert Group	Call Home Trigger Event	Syslog Event	Severity	Description and Executed Commands
Inventory	OIR_REMOVE OIR_INSERTION	—	**normal**	Inventory status that is provided whenever a unit is cold booted, or when FRUs are inserted or removed. This alert is considered a non-critical event, and the information is used for status and entitlement. Command executed: • **show platform** • **show version** • **show inventory oid** • **show diag all eeprom detail** • **show interfaces** • **show file systems** • **show bootflash: all** • **show data-corruption** • **show memory statistics** • **show process memory** • **show process cpu** • **show process cpu history** • **show license udi** • **show license detail** • **show buffers** • **show platform software proc slot monitor cycle**
Snapshot	—	—	**normal**	User-generated CLI commands.
Syslog	—	—	**major**	Events generated by Syslog messages. Commands executed: • **show inventory** • **show logging**

Alert Group	Call Home Trigger Event	Syslog Event	Severity	Description and Executed Commands
Test	—	—	**normal**	User-generated test message sent to the destination profile. Commands executed: • **show inventory** • **show platform** • **show version**

Message Contents

Smart Call Home supports the following message formats:

- Short Text Message Format
- Common Fields for Full Text and XML Messages
- Fields Specific to Alert Group Messages for Full Text and XML Messages
- Inserted Fields for a Reactive and Proactive Event Message
- Inserted Fields for an Inventory Event Message
- Inserted Fields for a User-Generated Test Message

The table below describes the short text formatting option for all message types.

Table 208: Short Text Message Format

Data Item	Description
Device identification	Configured device name
Date/time stamp	Time stamp of the triggering event
Error isolation message	Plain English description of triggering event
Alarm urgency level	Error level such as that applied to system message

The table below describes the first set of common event message fields for full text or XML messages.

Table 209: Common Fields for Full Text and XML Messages

Data Item (Plain Text and XML)	Description (Plain Text and XML)	Call-Home Message Tag (XML Only)
Time stamp	Date and time stamp of event in ISO time notation: *YYYY-MM-DD HH:MM:SS GMT+HH:MM.*	CallHome/EventTime
Message name	Name of message.	For short text message only
Message type	Name of message type, specifically "Call Home".	CallHome/Event/Type
Message subtype	Specific type of message: full, delta, test	CallHome/Event/SubType
Message group	Name of alert group, specifically "reactive". Optional, because default is "reactive". .	For long-text message only
Severity level	Severity level of message	Body/Block/Severity
Source ID	Product type for routing through the workflow engine. This is typically the product family name.	For long-text message only
Device ID	Unique device identifier (UDI) for the end device that generated the message. This field should be empty if the message is nonspecific to a device. The format is *type@Sid@serial*. • *type* is the product model number from the backplane IDPROM. • @ is a separator character. • *Sid* is C, identifying the serial ID as a chassis serial number· • *serial* is the number identified by the Sid field. An example is WS-C6509@C@12345678	CallHome/CustomerData/ContractData/DeviceId
Customer ID	Optional user-configurable field used for contract information or other ID by any support service.	CallHome/CustomerData/ContractData/CustomerId
Contract ID	Optional user-configurable field used for contract information or other ID by any support service.	CallHome/CustomerData/ContractData/ContractId

Data Item (Plain Text and XML)	Description (Plain Text and XML)	Call-Home Message Tag (XML Only)
Site ID	Optional user-configurable field used for Cisco-supplied site ID or other data meaningful to alternate support service.	CallHome/CustomerData/ContractData/SiteId
Server ID	If the message is generated from the device, this is the unique device identifier (UDI) of the device. The format is *type@Sid@serial*. • *type* is the product model number from the backplane IDPROM. • @ is a separator character. • *Sid* is C, identifying the serial ID as a chassis serial number. • *serial* is the number identified by the Sid field. An example is WS-C6509@C@12345678.	For long text message only
Message description	Short text that describes the error.	CallHome/MessageDescription
Device name	Node that experienced the event (hostname of the device).	CallHome/CustomerData/SystemInfo/NameName
Contact name	Name of person to contact for issues associated with the node that experienced the event.	CallHome/CustomerData/SystemInfo/Contact
Contact e-mail	E-mail address of person identified as the contact for this unit.	CallHome/CustomerData/SystemInfo/ContactEmail
Contact phone number	Phone number of the person identified as the contact for this unit.	CallHome/CustomerData/SystemInfo/ContactPhoneNumber
Street address	Optional field that contains the street address for RMA part shipments associated with this unit.	CallHome/CustomerData/SystemInfo/StreetAddress
Model name	Model name of the device (the specific model as part of a product family name).	CallHome/Device/Cisco_Chassis/Model
Serial number	Chassis serial number of the unit.	CallHome/Device/Cisco_Chassis/SerialNumber
Chassis part number	Top assembly number of the chassis.	/aml/body/chassis/partNo
System object ID	System Object ID that uniquely identifies the system.	CallHome/Device/Cisco_Chassis/ AdditionalInformation/AD@name=“sysObjectID”
System description	System description for the managed element.	CallHome/Device/Cisco_Chassis/ AdditionalInformation/AD@name=“sysDescr”

The table below describes the fields specific to alert group messages for full text and XML. These fields may be repeated if multiple commands are executed for an alert group.

Table 210: Fields Specific to Alert Group Messages for Full Text and XML Messages

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Command output name	Exact name of the issued command.	/aml/attachments/attachment/name
Attachment type	Specific command output.	/aml/attachments/attachment/type
MIME type	Either plain text or encoding type.	/aml/attachments/attachment/mime
Command output text	Output of command automatically executed.	/mml/attachments/attachment/atdata

The table below describes the reactive and proactive event message format for full text or XML messages.

Table 211: Inserted Fields for a Reactive and Proactive Event Message

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Chassis hardware version	Hardware version of chassis.	/aml/body/chassis/hwVersion
Supervisor module software version	Top-level software version.	/aml/body/chassis/swVersion
Affected FRU name	Name of the affected FRU that is generating the event message.	/aml/body/fru/name
Affected FRU serial number	Serial number of the affected FRU.	/aml/body/fru/serialNo
Affected FRU part number	Part number of the affected FRU.	/aml/body/fru/partNo
FRU slot	Slot number of the FRU that is generating the event message.	/aml/body/fru/slot
FRU hardware version	Hardware version of the affected FRU.	/aml/body/fru/hwVersion
FRU software version	Software version(s) that is running on the affected FRU.	/aml/body/fru/swVersion

The table below describes the inventory event message format for full text or XML messages.

Table 212: Inserted Fields for an Inventory Event Message

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Chassis hardware version	Hardware version of the chassis.	/aml/body/chassis/hwVersion
Supervisor module software version	Top-level software version.	/aml/body/chassis/swVersion
FRU name	Name of the affected FRU that is generating the event message.	/aml/body/fru/name
FRU s/n	Serial number of the FRU.	/aml/body/fru/serialNo
FRU part number	Part number of the FRU.	/aml/body/fru/partNo
FRU slot	Slot number of the FRU.	/aml/body/fru/slot

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
FRU hardware version	Hardware version of the FRU.	/aml/body/fru/hwVersion
FRU software version	Software version(s) that is running on the FRU.	/aml/body/fru/swVersion

The table below describes the user-generated test message format for full text or XML.

Table 213: Inserted Fields for a User-Generated Test Message

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Process ID	Unique process ID.	/aml/body/process/id
Process state	State of process (for example, running or halted).	/aml/body/process/processState
Process exception	Exception or reason code.	/aml/body/process/exception

Sample syslog Alert Notification in XML Format

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
xmlns:soap-env="http://www.w3.org/2003/05/soap-envelope">
<soap-env:Header>
<aml-session:Session
xmlns:aml-session="http://www.cisco.com/2004/01/aml-session"
soap-env:mustUnderstand="true"
soap-env:role="http://www.w3.org/2003/05/soap-envelope/role/next">
<aml-session:To>http://tools.cisco.com/neddce/services/DDCEService</aml
-session:To>
<aml-session:Path>
<aml-session:Via>http://www.cisco.com/appliance/uri</aml-session:Via>
</aml-session:Path>
<aml-session:From>http://www.cisco.com/appliance/uri</aml-session:From>
<aml-session:MessageId>MA:FXS1739Q0NR:548F4417</aml-session:MessageId>
</aml-session:Session>
</soap-env:Header>
<soap-env:Body>
<aml-block:Block
xmlns:aml-block="http://www.cisco.com/2004/01/aml-block">
<aml-block:Header>
<aml-block:Type>http://www.cisco.com/2005/05/callhome/syslog</aml-block
:Type>
<aml-block:CreationDate>2014-12-16 04:27:03
GMT+08:00</aml-block:CreationDate>
<aml-block:Builder>
<aml-block:Name>CBR8</aml-block:Name>
<aml-block:Version>2.0</aml-block:Version>
</aml-block:Builder>
<aml-block:BlockGroup>
<aml-block:GroupId>GB:FXS1739Q0NR:548F4417</aml-block:GroupId>
<aml-block:Number>0</aml-block:Number>
<aml-block:IsLast>true</aml-block:IsLast>
<aml-block:IsPrimary>true</aml-block:IsPrimary>
<aml-block:WaitForPrimary>false</aml-block:WaitForPrimary>
</aml-block:BlockGroup>
<aml-block:Severity>6</aml-block:Severity>
</aml-block:Header>
<aml-block:Content>
```

```
<ch:CallHome xmlns:ch="http://www.cisco.com/2005/05/callhome"
version="1.0">
<ch:EventTime>2014-12-16 04:26:59 GMT+08:00</ch:EventTime>
<ch:MessageDescription>Dec 16 04:26:59.885 CST: %ENVIRONMENTAL-1-ALERT:
Temp: INLET, Location: 6, State: Critical, Reading: 53
Celsius</ch:MessageDescription> <ch:Event> <ch:Type>syslog</ch:Type>
<ch:SubType></ch:SubType> <ch:Brand>Cisco Systems</ch:Brand>
<ch:Series>CBR8 Series Routers</ch:Series> </ch:Event>
<ch:CustomerData> <ch:UserData> <ch:Email>xxxx@company.com</ch:Email>
</ch:UserData>
<ch:ContractData>
<ch:CustomerId></ch:CustomerId>
<ch:SiteId></ch:SiteId>
<ch:ContractId></ch:ContractId>
<ch:DeviceId>CBR-8-CCAP-CHASS@C@FXS1739Q0NR</ch:DeviceId>
</ch:ContractData>
<ch:SystemInfo>
<ch:Name>sig-cbr</ch:Name>
<ch:Contact></ch:Contact>
<ch:ContactEmail>xxxx@company.com</ch:ContactEmail>
<ch:ContactPhoneNumber></ch:ContactPhoneNumber>
<ch:StreetAddress></ch:StreetAddress>
</ch:SystemInfo>
<ch:CCOID></ch:CCOID>
</ch:CustomerData>
<ch:Device>
<rme:Chassis xmlns:rme="http://www.cisco.com/rme/4.0">
<rme:Model>CBR-8-CCAP-CHASS</rme:Model>
<rme:HardwareVersion>0.1</rme:HardwareVersion>
<rme:SerialNumber>FXS1739Q0NR</rme:SerialNumber>
<rme:AdditionalInformation>
<rme:AD name="PartNumber" value="000-00000-00" /> <rme:AD
name="SoftwareVersion" value="15.5(20141214:005145) " /> <rme:AD
name="SystemObjectId" value="1.3.6.1.4.1.9.1.2141" /> <rme:AD
name="SystemDescription" value="Cisco IOS Software, IOS-XE Software
(X86_64_LINUX_IOSD-ADVENTERPRISEK9-M), Experimental Version
15.5(20141214:005145) [ece5_throttle_ios-ram-ece5-bk 105] Copyright (c)
1986-2014 by Cisco Systems, Inc.
Compiled Sun 14-Dec-14 00:20 by ram" /> <rme:AD name="ServiceNumber"
value="" /> <rme:AD name="ForwardAddress" value="" />
</rme:AdditionalInformation> </rme:Chassis> </ch:Device> </ch:CallHome>
</aml-block:Content> <aml-block:Attachments> <aml-block:Attachment
type="inline"> <aml-block:Name>show inventory</aml-block:Name>
<aml-block:Data encoding="plain"> <![CDATA[show inventory Load for five
secs: 2%/0%; one minute: 2%; five minutes: 2% Time source is NTP,
04:27:02.278 CST Tue Dec 16 2014
NAME: "Chassis", DESCR: "Cisco cBR-8 CCAP Chassis"
PID: CBR-8-CCAP-CHASS  , VID: V01, SN: FXS1739Q0NR

NAME: "sup 0", DESCR: "Cisco cBR CCAP Supervisor Card"
PID: CBR-CCAP-SUP-160G , VID: V01, SN: CAT1736E05L

NAME: "harddisk 4/1", DESCR: "Hard Disk"
PID: UGB88RTB100HE3-BCU-DID, VID:    , SN: 11000072780

NAME: "sup-pic 4/1", DESCR: "Cisco cBR CCAP Supervisor Card PIC"
PID: CBR-SUPPIC-8X10G  , VID: V01, SN: CAT1735E004

NAME: "SFP+ module 4/1/0", DESCR: "iNSI xcvr"
PID: SFP+ 10GBASE-SR   , VID: A   , SN: FNS1727294V

NAME: "SFP+ module 4/1/1", DESCR: "iNSI xcvr"
PID: SFP+ 10GBASE-SR   , VID: A   , SN: FNS172727WZ
```

```
NAME: "SFP+ module 4/1/4", DESCR: "iNSI xcvr"
PID: 10GE ZR           , VID:     , SN: AGM120525EW

NAME: "sup 1", DESCR: "Cisco cBR CCAP Supervisor Card"
PID: CBR-CCAP-SUP-160G , VID: V01, SN: CAT1736E05L

NAME: "clc 6", DESCR: "Cisco cBR CCAP Line Card"
PID: CBR-CCAP-LC-40G   , VID: V01, SN: CAT1736E0EN

NAME: "Cable PHY Module", DESCR: "CLC Downstream PHY Module 6/0"
PID: cBR-8-GEMINI      , VID: V01 , SN: CSJ13152101

NAME: "Cable PHY Module", DESCR: "CLC Upstream PHY Module 6/2"
PID: cBR-8-LEOBEN      , VID: V01 , SN: TST98765432

NAME: "Power Supply Module 0", DESCR: "Cisco cBR CCAP AC Power Supply"
PID: PWR-3KW-AC-V2     , VID: V02, SN: DTM173702KQ

NAME: "Power Supply Module 2", DESCR: "Cisco cBR CCAP AC Power Supply"
PID: PWR-3KW-AC-V2     , VID: V02, SN: DTM173702GD


sig-cbr#]]></aml-block:Data>
</aml-block:Attachment>
<aml-block:Attachment type="inline">
<aml-block:Name>show logging</aml-block:Name> <aml-block:Data
encoding="plain"> <![CDATA[show logging Load for five secs: 2%/0%; one
minute: 2%; five minutes: 2% Time source is NTP, 04:27:02.886 CST Tue
Dec 16 2014

Syslog logging: enabled (0 messages dropped, 51 messages rate-limited,
0 flushes, 0 overruns, xml disabled, filtering disabled)

No Active Message Discriminator.



No Inactive Message Discriminator.


    Console logging: level debugging, 213 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level debugging, 262 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled

No active filter modules.

    Trap logging: level informational, 209 message lines logged
        Logging Source-Interface:       VRF Name:

Log Buffer (1000000 bytes):

*Dec 15 20:20:16.188: Rommon debug: debugFlagsStr[7], flags[0x7] *Dec
15 20:20:16.188: TRACE - Debug flag set 0x7 *Dec 15 20:20:16.188: TRACE
- Register NV N:systemInitByEvent V:True with no CallBack *Dec 15
20:20:16.188: TRACE - Register NV N:routingReadyByEvent V:True with no
CallBack *Dec 15 20:20:16.188: TRACE - Smart agent init started.
Version=1.2.0_dev/22
*Dec 15 20:20:16.188: ERROR - PD init failed: The requested operation
```

```
is not supported *Dec 15 20:20:16.188: ERROR - Pre Role Init Failed:
The requested operation is not supported *Dec 15 20:20:16.188: TRACE -
Smart agent init Done. status 10, state 4294967295, init 0 enable 0
Current Role Invalid *Dec 15 20:20:16.188: TRACE - Shutdown Started
*Dec 15 20:20:16.188: DEBUG - Scheduler shutdown start *Dec 15
20:20:16.188: ERROR - Failed to set shutdown watched boolean (code
Invalid argument (22)). Going the hard way!!!
*Dec 15 20:20:16.188: DEBUG - Destroying XOS stuff to exit dispatch
loop *Dec 15 20:20:16.188: DEBUG - XDM dispatch loop about to exit *Dec
15 20:20:16.188: DEBUG - Scheduler shutdown end *Dec 15 20:20:16.188:
ERROR - SmartAgent not initialized.
*Dec 15 20:20:16.188: ERROR - Smart Agent not a RF client *Dec 15
20:20:16.188: ERROR - Smart Agent not a CF client *Dec 15 20:20:16.188:
TRACE - Setting Ha Mgmt Init FALSE *Dec 15 20:20:16.188: TRACE -
Shutting down Any Role *Dec 15 20:20:17.432: (DBMS RPHA) Client
initialization; status=success *Dec 15 20:20:17.432: CABLE Parser
Trace:  cable_parser_init:82 *Dec 15 20:20:17.774: ****
mcprp_ubr_punt_init: Initialized*****
-->RF_STATUS_SEND_RF_STATE received-->RF_PROG_INITIALIZATION received
*Dec 15 20:20:20.790: CWAN OIR debugging enabled (ROMMON variable
DEBUG_CWAN_OIR set)-->RF_PROG_ACTIVE_FAST
received-->RF_PROG_ACTIVE_DRAIN
received-->RF_PROG_ACTIVE_PRECONFIG
received-->received-->RF_PROG_ACTIVE_POSTCONFIG
received-->RF_PROG_ACTIVE received
*Dec 15 20:20:20.841: **** IPC port 0x1000E created!
*Dec 15 20:20:20.841:  **** CIPC RP Server created UBRCCE_CIPC_14/0 !
*Dec 15 20:20:28.294: %SPANTREE-5-EXTENDED_SYSID: Extended SysId
enabled for type vlan *Dec 15 20:20:31.944: %VOICE_HA-7-STATUS: CUBE
HA-supported platform detected.
*Dec 15 20:20:33.391:  instant_msg_handle_proc_sup started!!
*Dec 15 20:20:33.391:  queue_msg_handle_proc_sup started!!
*Dec 15 20:20:35.603: %IOSXE_MGMTVRF-6-CREATE_SUCCESS_INFO: Management
vrf Mgmt-intf created with ID 1, ipv4 table-id 0x1, ipv6 table-id
0x1E000001
*Dec 15 20:20:34.513: %IOSXE-6-PLATFORM: CLC4: cpp_cp: Process
CPP_PFILTER_EA_EVENT__API_CALL__REGISTER
*Dec 15 20:20:03.806: %HW_PFU-3-PFU_IDPROM_CORRUPT: R0/0: cmand:  The
PEM/FM idprom could be read, but is corrupt in slot P11 The system will
run without environmental monitoring for this component *Dec 15
20:20:09.012: %SYSTEM-3-SYSTEM_SHELL_LOG: R0/0:  2014/12/15
20:20:08 : <anon>
*Dec 15 20:20:13.919: %IOSXE-4-PLATFORM: R0/0: kernel: astro: FD open
*Dec 15 20:20:13.919: %IOSXE-4-PLATFORM: R0/0: kernel: astro: astro:
mmio_start=d0000000 mmio_len=2000000
*Dec 15 20:20:13.919: %IOSXE-4-PLATFORM: R0/0: kernel: astro: Done
astro Memory map base_ptr ffffc90016600000, astro_reg_ptr ffffc90016600000...
*Dec 15 20:20:16.259: %IOSXE-4-PLATFORM: R0/0: kernel: astro: FD open
*Dec 15 20:20:16.553: %CPPHA-7-START: F0: cpp_ha:  CPP 0 preparing
ucode *Dec 15 20:20:17.220: %CPPHA-7-START: F0: cpp_ha:  CPP 0 startup
init *Dec 15 20:20:18.549: %PMAN-3-PROC_EMPTY_EXEC_FILE: F0: pvp.sh:
Empty executable used for process iosdb *Dec 15 20:20:20.003:
%PMAN-3-PROC_EMPTY_EXEC_FILE: CLC4: pvp.sh:  Empty executable used for
process iosdb *Dec 15 20:20:20.783: %PMAN-3-PROC_EMPTY_EXEC_FILE: CLC4:
pvp.sh:  Empty executable used for process iosdb *Dec 15 20:20:24.061:
%HW_PFU-3-PFU_IDPROM_CORRUPT: R0/0: cmand:  The PEM/FM idprom could be
read, but is corrupt in slot P11 The system will run without
environmental monitoring for this component *Dec 15 20:20:31.722:
%CPPHA-7-START: F0: cpp_ha:  CPP 0 running init *Dec 15 20:20:32.070:
%CPPHA-7-READY: F0: cpp_ha:  CPP 0 loading and initialization complete
*Dec 15 20:20:36.528 UTC: TRACE - Platform EventCB invoked. EventType:
8 *Dec 15 20:20:36.528 UTC: DEBUG - Hostname changed. Old:sig-cbr
New:sig-cbr *Dec 15 20:20:36.528 UTC: %CNS IQ:0.1 ID:0
Changed:[sig-cbr] *Dec 15 20:20:36.528 UTC: %CNS IQ:0.2 ID:1
```

```
Changed:[sig-cbr] *Dec 15 20:20:36.528 UTC: %CNS IQ:0.3 ID:2
Changed:[sig-cbr] *Dec 15 20:20:36.594 UTC: %SYS-5-LOG_CONFIG_CHANGE:
Buffer logging: level debugging, xml disabled, filtering disabled, size
(1000000) *Dec 16 04:20:36.597 CST: %SYS-6-CLOCKUPDATE: System clock
has been updated from 20:20:36 UTC Mon Dec 15 2014 to 04:20:36 CST Tue
Dec 16 2014, configured from console by console.
*Dec 16 04:20:36.607 CST: spa_type 2946 ports 8 *Dec 16 04:20:36.622
CST: spa_type 2946 ports 8 *Dec 16 04:20:37.350 CST:
cmts_set_int_us_qos_flags: move US-QOS flags 0 to CDMAN *Dec 16
04:20:37.350 CST: cmts_set_int_us_default_weights: move US-QOS weights
to CDMAN *Dec 16 04:20:36.625 CST: %IOSXE-4-PLATFORM: R0/0: kernel:
astro: FD open *Dec 16 04:20:43.221 CST: %LINEPROTO-5-UPDOWN: Line
protocol on Interface Video6/0/0, changed state to up *Dec 16
04:20:43.223 CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Video6/0/1, changed state to up *Dec 16 04:20:43.502 CST: % Redundancy
mode change to SSO

*Dec 16 04:20:43.502 CST: %VOICE_HA-7-STATUS: NONE->SSO; SSO mode will
not take effect until after a platform
reload.-->RF_STATUS_REDUNDANCY_MODE_CHANGE received *Dec 16
04:20:44.220 CST: %SYS-5-CONFIG_I: Configured from memory by console
*Dec 16 04:20:44.228 CST: %IOSXE_OIR-6-INSCARD: Card (rp) inserted in
slot R1 *Dec 16 04:20:44.229 CST: %IOSXE_OIR-6-INSCARD: Card (fp)
inserted in slot F0 *Dec 16 04:20:44.229 CST: %IOSXE_OIR-6-ONLINECARD:
Card (fp) online in slot F0 *Dec 16 04:20:44.263 CST:
%IOSXE_OIR-6-INSCARD: Card (fp) inserted in slot F1 *Dec 16
04:20:44.263 CST: %IOSXE_OIR-6-INSCARD: Card (cc) inserted in slot 4
*Dec 16 04:20:44.263 CST: %IOSXE_OIR-6-ONLINECARD: Card (cc) online in
slot 4 *Dec 16 04:20:44.264 CST: %IOSXE_OIR-6-INSCARD: Card (cc)
inserted in slot 5 *Dec 16 04:20:44.264 CST: %IOSXE_OIR-6-INSCARD: Card
(cc) inserted in slot 6 *Dec 16 04:20:44.330 CST: %IOSXE_OIR-6-INSSPA:
SPA inserted in subslot 4/1 *Dec 16 04:20:44.751 CST: %SYS-5-RESTART:
System restarted -- Cisco IOS Software, IOS-XE Software
(X86_64_LINUX_IOSD-ADVENTERPRISEK9-M), Experimental Version
15.5(20141214:005145) [ece5_throttle_ios-ram-ece5-bk 105] Copyright (c)
1986-2014 by Cisco Systems, Inc.
Compiled Sun 14-Dec-14 00:20 by ram
*Dec 16 04:20:44.775 CST: %XML-SRVC: Security Enforcement XML
Service(111) OK. PID=574
*Dec 16 04:20:44.775 CST: %SSH-5-ENABLED: SSH 1.99 has been enabled
*Dec 16 04:20:45.453 CST: %LINK-3-UPDOWN: Interface GigabitEthernet0,
changed state to up *Dec 16 04:20:45.543 CST: %LINK-5-CHANGED:
Interface TenGigabitEthernet4/1/2, changed state to administratively
down *Dec 16 04:20:45.546 CST: %LINK-5-CHANGED: Interface
TenGigabitEthernet4/1/3, changed state to administratively down *Dec 16
04:20:45.548 CST: %LINK-5-CHANGED: Interface TenGigabitEthernet4/1/4,
changed state to administratively down *Dec 16 04:20:45.551 CST:
%LINK-5-CHANGED: Interface TenGigabitEthernet4/1/5, changed state to
administratively down *Dec 16 04:20:45.571 CST: %LINK-5-CHANGED:
Interface TenGigabitEthernet4/1/6, changed state to administratively
down *Dec 16 04:20:45.574 CST: %LINK-5-CHANGED: Interface
TenGigabitEthernet4/1/7, changed state to administratively down *Dec 16
04:20:45.576 CST: %LINK-5-CHANGED: Interface TenGigabitEthernet5/1/0,
changed state to administratively down *Dec 16 04:20:45.578 CST:
%LINK-5-CHANGED: Interface TenGigabitEthernet5/1/1, changed state to
administratively down *Dec 16 04:20:45.580 CST: %LINK-5-CHANGED:
Interface TenGigabitEthernet5/1/2, changed state to administratively
down *Dec 16 04:20:45.582 CST: %LINK-5-CHANGED: Interface
TenGigabitEthernet5/1/3, changed state to administratively down *Dec 16
04:20:45.584 CST: %LINK-5-CHANGED: Interface TenGigabitEthernet5/1/4,
changed state to administratively down *Dec 16 04:20:45.586 CST:
%LINK-5-CHANGED: Interface TenGigabitEthernet5/1/5, changed state to
administratively down *Dec 16 04:20:45.588 CST: %LINK-5-CHANGED:
Interface TenGigabitEthernet5/1/6, changed state to administratively
```

```
down *Dec 16 04:20:45.590 CST: %LINK-5-CHANGED: Interface
TenGigabitEthernet5/1/7, changed state to administratively down *Dec 16
04:20:45.596 CST: %LINK-3-UPDOWN: Interface Integrated-Cable6/0/0:0,
changed state to down *Dec 16 04:20:45.602 CST: %LINK-3-UPDOWN:
Interface Integrated-Cable6/0/0:1, changed state to down *Dec 16
04:20:45.603 CST: %LINK-3-UPDOWN: Interface Integrated-Cable6/0/0:2,
changed state to down *Dec 16 04:20:45.604 CST: %LINK-3-UPDOWN:
Interface Integrated-Cable6/0/0:3, changed state to down *Dec 16
04:20:45.606 CST: %LINK-3-UPDOWN: Interface Integrated-Cable6/0/0:4,
changed state to down *Dec 16 04:20:45.607 CST: %LINK-3-UPDOWN:
Interface Integrated-Cable6/0/0:5, changed state to down *Dec 16
04:20:45.608 CST: %LINK-3-UPDOWN: Interface Integrated-Cable6/0/0:6,
changed state to down *Dec 16 04:20:45.610 CST: %LINK-3-UPDOWN:
Interface Integrated-Cable6/0/0:7, changed state to down *Dec 16
04:20:45.648 CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Bundle1, changed state to up *Dec 16 04:20:45.649 CST: %LINK-3-UPDOWN:
Interface Bundle1, changed state to up *Dec 16 04:20:45.649 CST:
%LINK-3-UPDOWN: Interface Cable6/0/0, changed state to down *Dec 16
04:20:45.649 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
Cable6/0/0 changed state to down
*Dec 16 04:20:45.666 CST: %LINK-3-UPDOWN: Interface
Wideband-Cable6/0/0:0, changed state to down *Dec 16 04:20:45.666 CST:
%LINK-3-UPDOWN: Interface Wideband-Cable6/0/0:1, changed state to down
*Dec 16 04:20:45.681 CST: %LINK-3-UPDOWN: Interface
Wideband-Cable6/0/0:2, changed state to down *Dec 16 04:20:45.681 CST:
%LINK-3-UPDOWN: Interface Wideband-Cable6/0/0:3, changed state to down
*Dec 16 04:20:45.681 CST: %LINK-3-UPDOWN: Interface
Wideband-Cable6/0/0:4, changed state to down *Dec 16 04:20:45.681 CST:
%LINK-3-UPDOWN: Interface Wideband-Cable6/0/0:5, changed state to down
*Dec 16 04:20:45.682 CST: %LINK-3-UPDOWN: Interface
Wideband-Cable6/0/0:6, changed state to down *Dec 16 04:20:45.682 CST:
%LINK-3-UPDOWN: Interface Wideband-Cable6/0/0:7, changed state to down
*Dec 16 04:20:45.685 CST: %LINK-3-UPDOWN: Interface
Integrated-Cable6/0/1:0, changed state to down *Dec 16 04:20:45.694
CST: %LINK-3-UPDOWN: Interface Integrated-Cable6/0/1:1, changed state
to down *Dec 16 04:20:45.694 CST: %LINK-3-UPDOWN: Interface Cable6/0/1,
changed state to down *Dec 16 04:20:45.694 CST: %SNMP-5-LINK_DOWN:
LinkDown:Interface
Cable6/0/1 changed state to down
*Dec 16 04:20:45.699 CST: %LINK-3-UPDOWN: Interface
Wideband-Cable6/0/1:0, changed state to down *Dec 16 04:20:45.703 CST:
%LINK-3-UPDOWN: Interface Wideband-Cable6/0/1:1, changed state to down
*Dec 16 04:20:45.706 CST: %LINK-3-UPDOWN: Interface
Integrated-Cable6/0/1:2, changed state to down *Dec 16 04:20:45.707
CST: %LINK-3-UPDOWN: Interface Integrated-Cable6/0/1:3, changed state
to down *Dec 16 04:20:45.709 CST: %LINK-3-UPDOWN: Interface
Wideband-Cable6/0/2:0, changed state to down *Dec 16 04:20:46.469 CST:
%SNMP-5-COLDSTART: SNMP agent on host sig-cbr is undergoing a cold
start *Dec 16 04:20:46.472 CST: %LINEPROTO-5-UPDOWN: Line protocol on
Interface GigabitEthernet0, changed state to up *Dec 16 04:20:46.543
CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet4/1/2, changed state to down *Dec 16 04:20:46.546
CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet4/1/3, changed state to down *Dec 16 04:20:46.548
CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet4/1/4, changed state to down *Dec 16 04:20:46.551
CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet4/1/5, changed state to down *Dec 16 04:20:46.571
CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet4/1/6, changed state to down *Dec 16 04:20:46.574
CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet4/1/7, changed state to down *Dec 16 04:20:46.576
CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet5/1/0, changed state to down *Dec 16 04:20:46.578
```

```
CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet5/1/1, changed state to down *Dec 16 04:20:46.580
CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet5/1/2, changed state to down *Dec 16 04:20:46.582
CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet5/1/3, changed state to down *Dec 16 04:20:46.584
CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet5/1/4, changed state to down *Dec 16 04:20:46.586
CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet5/1/5, changed state to down *Dec 16 04:20:46.588
CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet5/1/6, changed state to down *Dec 16 04:20:46.590
CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet5/1/7, changed state to down *Dec 16 04:20:46.641
CST: %SYS-6-BOOTTIME: Time taken to reboot after reload =  374 seconds
*Dec 16 04:20:53.697 CST: %IOSXE-1-PLATFORM: R0/0: kernel: Raptor MAC
image download wrote 55917152 bytes *Dec 16 04:21:23.432 CST:
%TRANSCEIVER-6-INSERTED: CLC4: iomd:
transceiver module inserted in TenGigabitEthernet4/1/0 *Dec 16
04:21:23.435 CST: %TRANSCEIVER-6-INSERTED: CLC4: iomd:
transceiver module inserted in TenGigabitEthernet4/1/1 *Dec 16
04:21:23.440 CST: %TRANSCEIVER-6-INSERTED: CLC4: iomd:
transceiver module inserted in TenGigabitEthernet4/1/4 *Dec 16
04:21:29.430 CST: %CBRDTI-5-DTISLOT: DTI slot 4/1: card role changed to
Active

*Dec 16 04:21:29.454 CST: %SPA_OIR-6-ONLINECARD: SPA (CBR-SUPPIC-8X10G)
online in subslot 4/1 *Dec 16 04:21:31.403 CST: %LINK-3-UPDOWN:
Interface TenGigabitEthernet4/1/0, changed state to up *Dec 16
04:21:31.405 CST: %CBR_SPA-7-RAPTOR_ESI_EGRESS_HDR_LO_INTERRUPT:
CLC4: iomd:  LOCAL RAPTOR, DP 0, channel_not_found_err *Dec 16
04:21:31.412 CST: %LINK-3-UPDOWN: Interface TenGigabitEthernet4/1/1,
changed state to up *Dec 16 04:21:32.403 CST: %LINEPROTO-5-UPDOWN: Line
protocol on Interface TenGigabitEthernet4/1/0, changed state to up *Dec
16 04:21:32.412 CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet4/1/1, changed state to up *Dec 16 04:21:41.171 CST:
%IOSXE-3-PLATFORM: R0/0: kernel: i801_smbus
0000:00:1f.3: Transaction timeout
*Dec 16 04:21:41.174 CST: %IOSXE-3-PLATFORM: R0/0: kernel:
/nobackup/ram/ece5-bk/binos/os/linux/drivers/binos/i2c/max3674/max3674_
mai n.c:show_reg_pll (line 88): show_reg_pll failed *Dec 16
04:21:58.237 CST: %IOSXE-5-PLATFORM: CLC6: cdman: Basestar FPGA rev_id
0x00000002, fpga_rev_id 0x00000032 *Dec 16 04:21:59.074 CST:
%CMRP-3-BAD_ID_HW: R0/0: cmand:  Failed Identification Test in CBR
linecard. The module linecard slot 6 in this router may not be a
genuine Cisco product. Cisco warranties and support programs only apply
to genuine Cisco products. If Cisco determines that your insertion of
non-Cisco memory, WIC cards, AIM cards,  Network Modules, SPA cards,
GBICs or other modules into a  Cisco product is the cause of a support
issue, Cisco may  deny support under your warranty or under a Cisco
support  pro *Dec 16 04:21:59.075 CST: %IOSXE_OIR-6-ONLINECARD: Card
(cc) online in slot 6 *Dec 16 04:22:08.825 CST:
%ASR1000_INFRA-3-EOBC_SOCK: CLC6:
ubrclc-k9lc-ms:  Socket event for  EO6/0/1, fd 11, failed to bind;
Address already in use success *Dec 16 04:22:09.605 CST:  SNMP IPC
session up(RP <-> slot 6)!
*Dec 16 04:22:09.605 CST:  CMTS IPC session up!
*Dec 16 04:22:14.564 CST: %SNMP-5-LINK_UP: LinkUp:Interface
Cable6/0/0-upstream0 changed state to up *Dec 16 04:22:14.565 CST:
%SNMP-5-LINK_UP: LinkUp:Interface
Cable6/0/0-upstream1 changed state to up *Dec 16 04:22:14.566 CST:
%SNMP-5-LINK_UP: LinkUp:Interface
Cable6/0/2-upstream0 changed state to up *Dec 16 04:22:14.566 CST:
%SNMP-5-LINK_UP: LinkUp:Interface
```

```
Cable6/0/2-upstream1 changed state to up *Dec 16 04:22:15.051 CST:
%SNMP-5-LINK_UP: LinkUp:Interface Cable6/0/0 changed state to up *Dec
16 04:22:15.258 CST: %SNMP-5-LINK_UP: LinkUp:Interface Cable6/0/1
changed state to up *Dec 16 04:22:15.258 CST: %SNMP-5-LINK_UP:
LinkUp:Interface Cable6/0/2 changed state to up *Dec 16 04:22:15.259
CST: %SNMP-5-LINK_UP: LinkUp:Interface Cable6/0/3 changed state to up
*Dec 16 04:22:15.259 CST: %SNMP-5-LINK_UP: LinkUp:Interface Cable6/0/4
changed state to up *Dec 16 04:22:15.411 CST: %SNMP-5-LINK_UP:
LinkUp:Interface Cable6/0/5 changed state to up *Dec 16 04:22:15.411
CST: %SNMP-5-LINK_UP: LinkUp:Interface Cable6/0/6 changed state to up
*Dec 16 04:22:15.411 CST: %SNMP-5-LINK_UP: LinkUp:Interface Cable6/0/7
changed state to up *Dec 16 04:22:15.411 CST: %SNMP-5-LINK_UP:
LinkUp:Interface Cable6/0/8 changed state to up *Dec 16 04:22:15.432
CST: %SNMP-5-LINK_UP: LinkUp:Interface Cable6/0/9 changed state to up
*Dec 16 04:22:15.432 CST: %SNMP-5-LINK_UP: LinkUp:Interface Cable6/0/10
changed state to up *Dec 16 04:22:15.433 CST: %SNMP-5-LINK_UP:
LinkUp:Interface Cable6/0/11 changed state to up *Dec 16 04:22:15.433
CST: %SNMP-5-LINK_UP: LinkUp:Interface Cable6/0/12 changed state to up
*Dec 16 04:22:15.433 CST: %SNMP-5-LINK_UP: LinkUp:Interface Cable6/0/13
changed state to up *Dec 16 04:22:15.433 CST: %SNMP-5-LINK_UP:
LinkUp:Interface Cable6/0/14 changed state to up *Dec 16 04:22:15.433
CST: %SNMP-5-LINK_UP: LinkUp:Interface Cable6/0/15 changed state to up
*Dec 16 04:22:15.677 CST: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Cable6/0/8, changed state to up *Dec 16 04:22:15.678 CST:
%LINEPROTO-5-UPDOWN: Line protocol on Interface Cable6/0/9, changed
state to up *Dec 16 04:22:15.901 CST: %LINEPROTO-5-UPDOWN: Line
protocol on Interface Cable6/0/10, changed state to up *Dec 16
04:22:15.902 CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Cable6/0/11, changed state to up *Dec 16 04:22:15.902 CST:
%LINEPROTO-5-UPDOWN: Line protocol on Interface Cable6/0/12, changed
state to up *Dec 16 04:22:15.903 CST: %LINEPROTO-5-UPDOWN: Line
protocol on Interface Cable6/0/13, changed state to up *Dec 16
04:22:15.903 CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Cable6/0/14, changed state to up *Dec 16 04:22:15.904 CST:
%LINEPROTO-5-UPDOWN: Line protocol on Interface Cable6/0/15, changed
state to up *Dec 16 04:22:17.046 CST: %LINEPROTO-5-UPDOWN: Line
protocol on Interface Cable6/0/0, changed state to up *Dec 16
04:22:17.047 CST: %LINK-3-UPDOWN: Interface Cable6/0/0, changed state
to up *Dec 16 04:22:17.256 CST: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Cable6/0/1, changed state to up *Dec 16 04:22:17.257 CST:
%LINK-3-UPDOWN: Interface Cable6/0/1, changed state to up *Dec 16
04:22:17.259 CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Cable6/0/2, changed state to up *Dec 16 04:22:17.260 CST:
%LINK-3-UPDOWN: Interface Cable6/0/2, changed state to up *Dec 16
04:22:17.260 CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Cable6/0/3, changed state to up *Dec 16 04:22:17.260 CST:
%LINK-3-UPDOWN: Interface Cable6/0/3, changed state to up *Dec 16
04:22:17.260 CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Cable6/0/4, changed state to up *Dec 16 04:22:17.260 CST:
%LINK-3-UPDOWN: Interface Cable6/0/4, changed state to up *Dec 16
04:22:17.411 CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Cable6/0/5, changed state to up *Dec 16 04:22:17.411 CST:
%LINK-3-UPDOWN: Interface Cable6/0/5, changed state to up *Dec 16
04:22:17.411 CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Cable6/0/6, changed state to up *Dec 16 04:22:17.411 CST:
%LINK-3-UPDOWN: Interface Cable6/0/6, changed state to up *Dec 16
04:22:17.411 CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Cable6/0/7, changed state to up *Dec 16 04:22:17.412 CST:
%LINK-3-UPDOWN: Interface Cable6/0/7, changed state to up *Dec 16
04:22:16.714 CST: %IOSXE-5-PLATFORM: CLC6: cdman: DS-JIB:ILK Interrupts
Enabled. (Init:20539, Check:9566 1stPKO:8942) *Dec 16 04:22:17.809 CST:
%CMRP-3-IDPROM_SENSOR: R0/0: cmand:  One or more sensor fields from the
idprom failed to parse properly because Invalid argument.
Dec 16 04:22:57.161 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
```

```
Integrated-Cable6/0/0-downstream0 changed state to down Dec 16
04:22:57.161 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
Integrated-Cable6/0/0-downstream1 changed state to down Dec 16
04:22:57.161 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
Integrated-Cable6/0/0-downstream2 changed state to down Dec 16
04:22:57.162 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
Integrated-Cable6/0/0-downstream3 changed state to down Dec 16
04:22:57.162 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
Integrated-Cable6/0/0-downstream4 changed state to down Dec 16
04:22:57.162 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
Integrated-Cable6/0/0-downstream5 changed state to down Dec 16
04:22:57.162 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
Integrated-Cable6/0/0-downstream6 changed state to down Dec 16
04:22:57.162 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
Integrated-Cable6/0/0-downstream7 changed state to down Dec 16
04:22:57.163 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
Integrated-Cable6/0/0-downstream8 changed state to down Dec 16
04:22:57.163 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
Integrated-Cable6/0/0-downstream9 changed state to down Dec 16
04:22:57.163 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
Integrated-Cable6/0/0-downstream10 changed state to down Dec 16
04:22:57.163 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
Integrated-Cable6/0/1-downstream0 changed state to down Dec 16
04:22:57.164 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
Integrated-Cable6/0/1-downstream1 changed state to down Dec 16
04:22:57.164 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
Integrated-Cable6/0/1-downstream2 changed state to down Dec 16
04:22:57.164 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
Integrated-Cable6/0/1-downstream3 changed state to down Dec 16
04:22:57.164 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
Integrated-Cable6/0/1-downstream4 changed state to down Dec 16
04:22:57.164 CST: %SNMP-5-LINK_DOWN: LinkDown:Interface
Integrated-Cable6/0/1-downstream5 changed state to down Dec 16
04:22:57.183 CST: %SNMP-5-LINK_UP: LinkUp:Interface
Integrated-Cable6/0/0-downstream0 changed state to up Dec 16
04:22:57.184 CST: %SNMP-5-LINK_UP: LinkUp:Interface
Integrated-Cable6/0/0-downstream1 changed state to up Dec 16
04:22:57.189 CST: %SNMP-5-LINK_UP: LinkUp:Interface
Integrated-Cable6/0/0-downstream2 changed state to up Dec 16
04:22:57.211 CST: %SNMP-5-LINK_UP: LinkUp:Interface
Integrated-Cable6/0/0-downstream3 changed state to up Dec 16
04:22:57.212 CST: %SNMP-5-LINK_UP: LinkUp:Interface
Integrated-Cable6/0/0-downstream4 changed state to up Dec 16
04:22:57.212 CST: %SNMP-5-LINK_UP: LinkUp:Interface
Integrated-Cable6/0/0-downstream6 changed state to up Dec 16
04:22:57.213 CST: %SNMP-5-LINK_UP: LinkUp:Interface
Integrated-Cable6/0/0-downstream7 changed state to up Dec 16
04:22:57.213 CST: %SNMP-5-LINK_UP: LinkUp:Interface
Integrated-Cable6/0/0-downstream8 changed state to up Dec 16
04:22:57.213 CST: %SNMP-5-LINK_UP: LinkUp:Interface
Integrated-Cable6/0/0-downstream9 changed state to up Dec 16
04:22:57.213 CST: %SNMP-5-LINK_UP: LinkUp:Interface
Integrated-Cable6/0/0-downstream10 changed state to up Dec 16
04:22:57.214 CST: %SNMP-5-LINK_UP: LinkUp:Interface
Integrated-Cable6/0/1-downstream0 changed state to up Dec 16
04:22:57.424 CST: %SNMP-5-LINK_UP: LinkUp:Interface
Integrated-Cable6/0/1-downstream1 changed state to up Dec 16
04:22:57.426 CST: %SNMP-5-LINK_UP: LinkUp:Interface
Integrated-Cable6/0/1-downstream2 changed state to up Dec 16
04:22:57.435 CST: %SNMP-5-LINK_UP: LinkUp:Interface
Integrated-Cable6/0/1-downstream3 changed state to up Dec 16
04:22:57.437 CST: %SNMP-5-LINK_UP: LinkUp:Interface
Integrated-Cable6/0/1-downstream4 changed state to up Dec 16
04:22:57.449 CST: %SNMP-5-LINK_UP: LinkUp:Interface
```

```
Integrated-Cable6/0/1-downstream5 changed state to up Dec 16
04:22:59.219 CST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Integrated-Cable6/0/1:0, changed state to up Dec 16 04:22:59.219 CST:
%LINK-3-UPDOWN: Interface Integrated-Cable6/0/1:0, changed state to up
Dec 16 04:22:59.427 CST: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Integrated-Cable6/0/1:1, changed state to up Dec 16
04:22:59.427 CST: %LINK-3-UPDOWN: Interface Integrated-Cable6/0/1:1,
changed state to up Dec 16 04:22:59.449 CST: %LINEPROTO-5-UPDOWN: Line
protocol on Interface Wideband-Cable6/0/0:0, changed state to up Dec 16
04:22:59.450 CST: %LINK-3-UPDOWN: Interface Wideband-Cable6/0/0:0,
changed state to up Dec 16 04:22:59.450 CST: %LINK-3-UPDOWN: Interface
Wideband-Cable6/0/0:1, changed state to up Dec 16 04:22:59.450 CST:
%LINK-3-UPDOWN: Interface Wideband-Cable6/0/0:2, changed state to up
Dec 16 04:22:59.450 CST: %LINK-3-UPDOWN: Interface
Wideband-Cable6/0/0:3, changed state to up Dec 16 04:22:59.450 CST:
%LINK-3-UPDOWN: Interface Wideband-Cable6/0/0:4, changed state to up
Dec 16 04:22:59.450 CST: %LINK-3-UPDOWN: Interface
Wideband-Cable6/0/0:5, changed state to up Dec 16 04:22:59.451 CST:
%LINK-3-UPDOWN: Interface Wideband-Cable6/0/0:6, changed state to up
Dec 16 04:22:59.451 CST: %LINK-3-UPDOWN: Interface
Wideband-Cable6/0/0:7, changed state to up Dec 16 04:22:59.451 CST:
%LINEPROTO-5-UPDOWN: Line protocol on Interface Wideband-Cable6/0/1:0,
changed state to up Dec 16 04:22:59.451 CST: %LINK-3-UPDOWN: Interface
Wideband-Cable6/0/1:0, changed state to up Dec 16 04:22:59.451 CST:
%LINEPROTO-5-UPDOWN: Line protocol on Interface Wideband-Cable6/0/1:1,
changed state to up Dec 16 04:22:59.452 CST: %LINK-3-UPDOWN: Interface
Wideband-Cable6/0/1:1, changed state to up Dec 16 04:22:59.452 CST:
%LINK-3-UPDOWN: Interface Wideband-Cable6/0/2:0, changed state to up
Dec 16 04:23:27.352 CST: %IOSXE-5-PLATFORM: CLC6: cdman: DSPHY Gemini
module  1 was not present Dec 16 04:26:59.885 CST:
%ENVIRONMENTAL-1-ALERT: Temp: INLET, Location:
6, State: Critical, Reading: 53 Celsius sig-cbr#]]></aml-block:Data>
</aml-block:Attachment> </aml-block:Attachments> </aml-block:Block>
</soap-env:Body> </soap-env:Envelope>
```

Additional References

Related Documents

Related Topic	Document Title
Smart Call Home site page on Cisco.com for access to all related product information.	Cisco Smart Call Home site
The User Guide explains how the Smart Call Home service offers web-based access to important information on select Cisco devices. The User Guide also describes the higher network availability and increased operational efficiency by providing real-time alerts.	Smart Call Home User Guide
Call Home Quick Start Guide	Smart Call Home Quick Start Configuration Guide for Cisco cBR Series Routers

MIBs

MIB	MIBs Link
CISCO-CALLHOME-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. You can subscribe to various services to receive security and technical information about your products. The services include the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

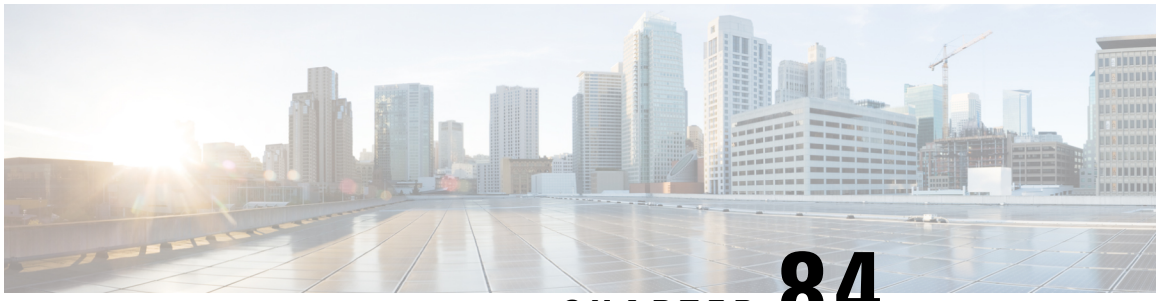
Feature Information for Call Home

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 214: Feature Information for Call Home

Feature Name	Releases	Feature Information
Smart Call Home	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 84

SNMP Support over VPNs—Context-Based Access Control

The SNMP Support over VPNs--Context-Based Access Control feature provides the infrastructure for multiple Simple Network Management Protocol (SNMP) context support in Cisco software and VPN-aware MIB infrastructure using the multiple SNMP context support infrastructure.

- Finding Feature Information, on page 1319
- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1319
- Restrictions for SNMP Support over VPNs—Context-Based Access Control, on page 1320
- Information About SNMP Support over VPNs—Context-Based Access Control, on page 1320
- How to Configure SNMP Support over VPNs—Context-Based Access Control, on page 1323
- Configuration Examples for SNMP Support over VPNs—Context-Based Access Control, on page 1327
- Additional References, on page 1328
- Feature Information for SNMP Support over VPNs—Context-Based Access Control, on page 1329

Finding Feature Information

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 215: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for SNMP Support over VPNs—Context-Based Access Control

- If you delete an SNMP context using the **no snmp-server context** command, all SNMP instances in that context are deleted.
- Not all MIBs are VPN-aware.

Information About SNMP Support over VPNs—Context-Based Access Control

SNMP Versions and Security

Cisco software supports the following versions of SNMP:

- SNMPv1—Simple Network Management Protocol: a full Internet standard, defined in RFC 1157. (RFC 1157 replaces the earlier versions that were published as RFC 1067 and RFC 1098.) Security is based on community strings.

- SNMPv2c—The community string-based Administrative Framework for SNMPv2. SNMPv2c (the "c" is for "community") is an experimental Internet protocol defined in RFC 1901, RFC 1905, and RFC 1906. SNMPv2c is an update of the protocol operations and data types of SNMPv2p (SNMPv2 Classic) and uses the community-based security model of SNMPv1.

For more information about SNMP Versions, see the " Configuring SNMP Support " module in the *Cisco Network Management Configuration Guide*.

SNMPv1 or SNMPv2 Security

Cisco IOS software supports the following versions of SNMP:

- SNMPv1—Simple Network Management Protocol: a full Internet standard, defined in RFC 1157. (RFC 1157 replaces the earlier versions that were published as RFC 1067 and RFC 1098.) Security is based on community strings.
- SNMPv2c—The community string-based Administrative Framework for SNMPv2. SNMPv2c (the "c" is for "community") is an experimental Internet protocol defined in RFC 1901, RFC 1905, and RFC 1906. SNMPv2c is an update of the protocol operations and data types of SNMPv2p (SNMPv2 Classic) and uses the community-based security model of SNMPv1.

SNMPv1 and SNMPv2 are not as secure as SNMPv3. SNMP version 1 and 2 use plain text communities and do not perform the authentication or security checks that SNMP version 3 performs. To configure the SNMP Support over VPNs—Context-Based Access Control feature when using SNMP version 1 or SNMP version 2, you need to associate a community name with a VPN. This association causes SNMP to process requests coming in for a particular community string only if it comes in from the configured VRF. If the community string contained in the incoming packet does not have an associated VRF, it is processed only if it came in through a non-VRF interface. This process prevents users outside the VPN from snooping a clear text community string to query the VPN's data. These methods of source address validation are not as secure as using SNMPv3.

SNMPv3 Security

If you are using SNMPv3, the security name should always be associated with authentication or privileged passwords. Source address validation is not performed on SNMPv3 users. To ensure that a VPN's user has access only to context associated to the VPN and cannot see the MIB data of other VPNs, you must configure a minimum security level of AuthNoPriv.

On a provider edge (PE) router, a community can be associated with a VRF to provide source address validation. However, on a customer edge (CE) router, if source address validation is to be provided, you must associate a source address with the community list by using an access control list.

If you are using SNMPv3, the security name or security password of the users of a VPN should be unknown to users of other VPNs. Cisco recommends not to use SNMPv3 nonauthorized users if you need security of management information.

SNMP Notification Support over VPNs

The SNMP Notification Support over VPNs feature allows the sending and receiving of SNMP notifications (traps and informs) using VPN routing and forwarding (VRF) instance tables. In particular, this feature adds support to Cisco software for the sending and receiving of SNMP notifications (traps and informs) specific to individual VPNs.

SNMP is an application-layer protocol that provides a message format for communication between SNMP managers and agents.

A VPN is a network that provides high-connectivity transfers on a shared system with the same usage guidelines as a private network. A VPN can be built on the Internet over IP, Frame Relay, or ATM networks.

A VRF stores per-VPN routing data. It defines the VPN membership of a customer site attached to the network access server (NAS). A VRF consists of an IP routing table, a derived Cisco Express Forwarding (formerly known as CEF) table, and guidelines and routing protocol parameters that control the information that is included in the routing table.

The SNMP Support for VPNs—Context-Based Access Control feature provides configuration commands that allow users to associate SNMP agents and managers with specific VRFs. The associated VRF is used for the sending of SNMP notifications (traps and informs) and responses between agents and managers. If a VRF is not specified, the default routing table for the VPN is used.

VPN-Aware SNMP

The SNMP Support for VPNs—Context-Based Access Control feature extends the capabilities of the SNMP Notification Support for VPNs feature and enables SNMP to differentiate between incoming packets from different VPNs.

When the SNMP Support for VPNs—Context-Based Access Control feature is configured, SNMP accepts requests on any configured VRF and returns responses to the same VRF. A trap host also can be associated with a specific VRF. The configured VRF is then used for sending out traps; otherwise, the default routing table is used. You also can associate a remote user with a specific VRF. You also can configure the VRFs from which SNMP should accept requests. Any requests coming from VRFs that are not specified are dropped.

IP access lists can be configured and associated with SNMP community strings. This feature enables you to configure an association between VRF instances with SNMP community strings. When a VRF instance is associated with an SNMP community string, SNMP processes the requests coming in for a particular community string only if the requests are received from the configured VRF. If the community string contained in the incoming packet does not have a VRF associated with it, the community string will be processed only if it came in through a non-VRF interface.

You also can enable or disable authentication traps for SNMP packets dropped due to VRF mismatches. By default if SNMP authentication traps are enabled, VRF authentication traps are also enabled.

VPN Route Distinguishers

A route distinguisher (RD) creates routing and forwarding tables and specifies the default route distinguisher for a VPN. The RD is added to the beginning of the customer's IPv4 prefixes to change them into globally unique VPN-IPv4 prefixes.

The RD is either an autonomous system number (ASN)-relative RD, in which case it comprises an autonomous system number and an arbitrary number, or it is an IP-address-relative RD, in which case it comprises an IP address and an arbitrary number.

You can enter an RD in either of these formats:

- 16-bit ASN: your 16-bit number: For example, 101:3.
- 32-bit IP address: your 32-bit number: For example, 192.168.122.15:1.

SNMP Contexts

SNMP contexts provide VPN users with a secure way of accessing MIB data. When a VPN is associated with a context, that VPN's specific MIB data exists in that context. Associating a VPN with a context enables service providers to manage networks with multiple VPNs. Creating and associating a context with a VPN enables a provider to prevent the users of one VPN from accessing information about other VPN userss on the same networking device.

VPN-aware SNMP requires an agreement between SNMP manager and agent entities operating in a VPN environment on a mapping between the SNMP security name and the VPN ID. This mapping is created by using multiple contexts for the SNMP data of different VPNs through the configuration of the SNMP-VACM-MIB. The SNMP-VACM-MIB is configured with views so that a user on a VPN with a security name is allowed access to the restricted object space associated with a user's access type in the context associated with the user of that VPN.

SNMP request messages undergo three phases of security and access control before a response message is sent back with the object values in the context of a VPN:

- In the first phase, the username is authenticated. This phase ensures that the user is authenticated and authorized for SNMP access.
- In the second phase, the user is authorized for the SNMP access requested to the group objects under consideration of the configured SNMP context. This phase is called the access control phase.
- In the third phase, access is made to a particular instance of a table entry. With this third phase, complete retrieval can be based on the SNMP context name.

How to Configure SNMP Support over VPNs—Context-Based Access Control

Configuring an SNMP Context and Associating the SNMP Context with a VPN

Perform this task to configure an SNMP context and to associate the SNMP context with a VPN.

Note

- Only the following MIBs are context-aware. All the tables in these MIBs can be polled:
  - CISCO-IPSEC-FLOW-MONITOR-MIB
  - CISCO-IPSEC-MIB
  - CISCO-PING-MIB
  - IP-FORWARD-MIB
  - MPLS-LDP-MIB
- Only two SNMP variables in the IP-FORWARD-MIB can be polled: 1.3.6.1.2.1.4.24.3 (ipCidrRouteNumber - Scalar) and 1.3.6.1.2.1.4.24.4.1 (ipCidrRouteEntry - Table).

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server context** *context-name* **Example:** `Device(config)# snmp-server context context1`	Creates and names an SNMP context.
Step 4	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vrf1`	Configures a VRF routing table and enters VRF configuration mode.
Step 5	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:120`	Creates a VPN route distinguisher.
Step 6	**context** *context-name* **Example:** `Device(config-vrf)# context context1`	Associates an SNMP context with a particular VRF. **Note** Depending on your release, the **context** command is replaced by the **snmp context** command. See the *Cisco IOS Network Management Command Reference* for more information.
Step 7	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Device(config-vrf)# route-target export 100:1000`	(Optional) Creates a route-target extended community for a VRF.
Step 8	**end** **Example:** `Device(config-vrf)# end`	Exits interface mode and enters global configuration mode.

	Command or Action	Purpose
Step 9	**end** **Example:** `Device(config)# end`	Exits global configuration mode.

Configuring SNMP Support and Associating an SNMP Context

Perform this task to configure SNMP support and associate it with an SNMP context.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server user** *username group-name* [**remote** *host* [**udp-port** *port*] [**vrf** *vrf-name*]] { **v1** | **v2c** | **v3** [**encrypted**] [**auth** { **md5** | **sha** } *auth-password*]} [**access** [**ipv6** *nacl*] [**priv** { **des** | **3des** | **aes** { **128** | **192** | **256** }} *privpassword*] { *acl-number* | *acl-name* }] **Example:** `Device(config)# snmp-server user customer1 group1 v1`	Configures a new user to an SNMP group.
Step 4	**snmp-server group** *group-name* { **v1** | **v2c** | **v3** { **auth** | **noauth** | **priv** }} [**context** *context-name*] [**read** *read-view*] [**write** *write-view*] [**notify** *notify-view*] [**access** [**ipv6** *named-access-list*] [*acl-number* | *acl-name*]] **Example:** `Device(config)# snmp-server group group1 v1 context context1 read view1 write view1 notify view1`	Configures a new SNMP group or a table that maps SNMP users to SNMP views. • Use the **context** *context-name* keyword argument pair to associate the specified SNMP group with a configured SNMP context.
Step 5	**snmp-server view** *view-name oid-tree* { **included** | **excluded** }	Creates or updates a view entry.

	Command or Action	Purpose
	Example: `Device(config)# snmp-server view view1 ipForward included`	
Step 6	**snmp-server enable traps** [*notification-type*] [**vrrp**] **Example:** `Device(config)# snmp-server enable traps`	Enables all SNMP notifications (traps or informs) available on your system.
Step 7	**snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number* | *extended-access-list-number* | *access-list-name*] **Example:** `Device(config)# snmp-server community public view view1 rw`	Sets up the community access string to permit access to the SNMP.
Step 8	**snmp-server host** { *hostname* | *ip-address* } [**vrf** *vrf-name*] [**traps** | **informs**] [**version** { **1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] **Example:** `Device(config)# snmp-server host 10.0.0.1 vrf vrf1 public udp-port 7002`	Specifies the recipient of an SNMP notification operation.
Step 9	**snmp mib community-map** *community-name* [**context** *context-name*] [**engineid** *engine-id*] [**security-name** *security-name*] [**target-list** *upn-list-name*] **Example:** `Device(config)# snmp mib community-map community1 context context1 target-list commAVpn`	Associates an SNMP community with an SNMP context, Engine ID, or security name.
Step 10	**snmp mib target list** *vpn-list-name* { **vrf** *vrf-name* | **host** *ip-address* } **Example:** `Device(config)# snmp mib target list commAVpn vrf vrf1`	Creates a list of target VRFs and hosts to associate with an SNMP community.

	Command or Action	Purpose
Step 11	**no snmp-server trap authentication vrf** **Example:** `Device(config)# no snmp-server trap authentication vrf`	(Optional) Disables all SNMP authentication notifications (traps and informs) generated for packets received on VRF interfaces. • Use this command to disable authentication traps only for those packets on VRF interfaces with incorrect community associations.

Configuration Examples for SNMP Support over VPNs—Context-Based Access Control

Example: Configuring Context-Based Access Control

The following configuration example shows how to configure the SNMP Support over VPNs—Context-Based Access Control feature for SNMPv1 or SNMPv2:

Note Depending on your releases, the **context** command is replaced by the **snmp context** command. See the *Cisco IOS Network Management Command Reference* for more information.

```
snmp-server context A
snmp-server context B
ip vrf Customer_A
 rd 100:110
 context A
 route-target export 100:1000
 route-target import 100:1000
!
ip vrf Customer_B
 rd 100:120
 context B
 route-target export 100:2000
 route-target import 100:2000
!
interface TenGigabitEthernet4/1/0
 description Belongs to VPN A
 ip vrf forwarding CustomerA
 ip address 192.168.2.1 255.255.255.0

interface TenGigabitEthernet4/1/1
 description Belongs to VPN B
 ip vrf forwarding CustomerB
 ip address 192.168.2.2 255.255.255.0
snmp-server user commA grp1A v1
snmp-server user commA grp2A v2c
snmp-server user commB grp1B v1
snmp-server user commB grp2B v2c
snmp-server group grp1A v1 context A read viewA write viewA notify viewA
```

```
snmp-server group grp1B v1 context B read viewB write viewB notify viewB
snmp-server view viewA ipForward included
snmp-server view viewA ciscoPingMIB included
snmp-server view viewB ipForward included
snmp-server view viewB ciscoPingMIB included
snmp-server enable traps
snmp-server host 192.168.2.3 vrf CustomerA commA udp-port 7002
snmp-server host 192.168.2.4 vrf CustomerB commB udp-port 7002
snmp mib community-map commA context A target-list commAvpn
! Configures source address validation
snmp mib community-map commB context B target-list commBvpn
! Configures source address validation
snmp mib target list commAvpn vrf CustomerA
! Configures a list of VRFs or from which community commA is valid
snmp mib target list commBvpn vrf CustomerB
! Configures a list of VRFs or from which community commB is valid
```

Additional References

Related Documents

Related Topic	Document Title
Cisco Network Management commands	*Cisco IOS Network Management Command Reference*
SNMP configuration	"Configuring SNMP Support" chapter in the *Cisco Network Management Configuration Guide*
SNMP Support for VPNs	SNMP Notification Support for VPNs

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
• CISCO-PING-MIB • IP-FORWARD-MIB • SNMP-VACM-MIB, *The View-based Access Control Model (ACM) MIB for SNMP*	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1441	*Introduction to version 2 of the Internet-standard Network Management Framework*
RFC 1442	*Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)*
RFC 1443	*Textual Conventions for version 2 of the Simple Network Management Protocol (SNMPv2)*
RFC 1444	*Conformance Statements for version 2 of the Simple Network Management Protocol (SNMPv2)*
RFC 1445	*Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)*
RFC 1446	*Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2)*
RFC 1447	*Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2)*
RFC 1448	*Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)*
RFC 1449	*Transport Mappings for version 2 of the Simple Network Management Protocol (SNMPv2)*
RFC 1450	*Management Information Base for version 2 of the Simple Network Management Protocol (SNMPv2)*
RFC 2571	*An Architecture for Describing SNMP Management Frameworks*
RFC 2576	*Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provide online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

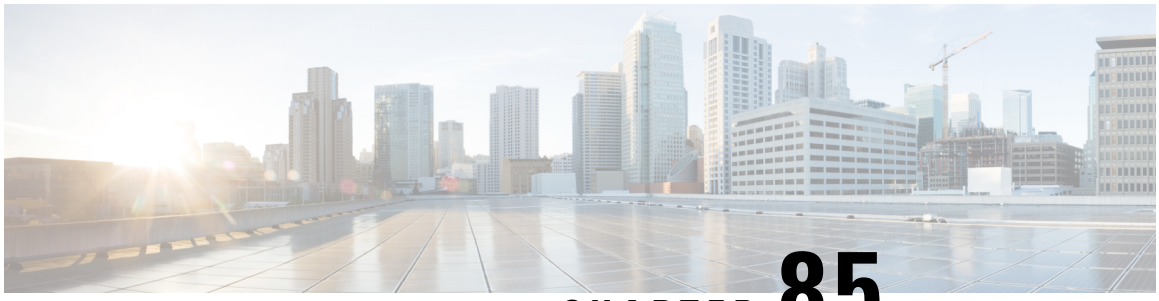
Feature Information for SNMP Support over VPNs—Context-Based Access Control

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 216: Feature Information for SNMP Support over VPNs—Context-Based Access Control

Feature Name	Releases	Feature Information
SNMP Support over VPNs—Context-Based Access Control	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 85

SNMP Engine Enhancement

The SNMP Cache Engine Enhancement feature caches the SNMP information on the Supervisor .

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1331
- Restrictions for SNMP Cache Engine Enhancement, on page 1332
- Information About SNMP Cache Engine Enhancement, on page 1332
- How to Configure SNMP Cache Engine Enhancement, on page 1333
- Verifying the SNMP Cache Engine Status, on page 1334
- Additional References, on page 1335
- Feature Information for SNMP Cache Engine Enhancement, on page 1335

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 217: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Restrictions for SNMP Cache Engine Enhancement

The time interval for which the cached information is available on the Supervisor is 5 seconds.

Information About SNMP Cache Engine Enhancement

The SNMP Cache Engine Enhancement feature caches the information on the Supervisor for the MIB tables, which need to retrieve the data from the interface cards. When a MIB table item is queried from the interface card, the next *N* items are retrieved and cached on the Supervisor .

For example, if SNMP client queries the docsIf3CmtsCmRegStatusMacAddr.1, the interface card bundles docsIf3CmtsCmRegStatusMacAddr.1, docsIf3CmtsCmRegStatusMacAddr.2, docsIf3CmtsCmRegStatusMacAddr.3, to docsIf3CmtsCmRegStatusMacAddr.N together in one IPC response, and sends it to the Supervisor . The Supervisor caches all the items locally. When the SNMP client queries the docsIf3CmtsCmRegStatusMacAddr.2 later, the information is available in the Supervisor cache directly instead of sending another IPC message to interface card. The number *N* depends on the single MIB item size and maximum IPC message buffer size.

The MIB table information for following MIBs are retrieved and cached on the Supervisor :

- DOCS-IF-MIB

- DOCS-IFEXT2-MIB
- DOCS-QOS-MIB
- DOCS-IF3-MIB
- DOCS-IF31-MIB
- DOCS-QOS3-MIB
- DOCS-IETF-QOS-MIB
- DOCS-BPI-PLUS-MIB
- DOCS-LOADBALANCING-MIB
- DOCS-LOADBAL3-MIB
- DOCS-DSG-IF-MIB
- CISCO-DOCS-EXT-MIB
- CISCO-CABLE-WIDEBAND-MIB
- CISCO-CABLE-SPECTRUM-MIB

This feature is enabled by default on the Cisco cBR routers. The time interval for which the SNMP cache information is stored on the Supervisor is known as *age* and set to 5 seconds.

How to Configure SNMP Cache Engine Enhancement

Before you begin

You must configure the **service internal** command in global configuration mode to enable or disable SNMP Cache Engine Enhancement.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable snmp cache active** **Example:** `Router(config)# cable snmp cache active`	Sets the SNMP cache status to active. **Note** Use the **no** form of the command to disable the SNMP cache status.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Router(config)# exit`	Exits the global configuration mode and enters the privileged EXEC mode.

Verifying the SNMP Cache Engine Status

Use the **show cable snmp cache-status** command to display the current SNMP cache engine status.

Important You must configure the **service internal** command in global configuration mode to verify the SNMP cache engine status.

Following is a sample output of the command.

```
Router# show cable snmp cache-status

    Cache engine is ON, age: 5 seconds
```

Use the **test cable snmp counter-show** command to display the cache counters information.

```
Router# test cable snmp counter-show
========== cache counters ==========
ubrcce_snmp_cache_hit_counter:0.
ubrcce_snmp_cache_get_from_lc_counter:1.
ubrcce_snmp_cache_miss_counter:0.
ubrcce_snmp_cache_ipc_fail_counter:0.
ubrcce_snmp_cache_buffer_full_counter:0.
```

hit and *mis* are the historic information for the SNMP cache after the system bootup. *hit* indicates the number of times the SNMP queries are hit in the cache and *mis* indicates the number of times the SNMP queries are missed in the SNMP cache.

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

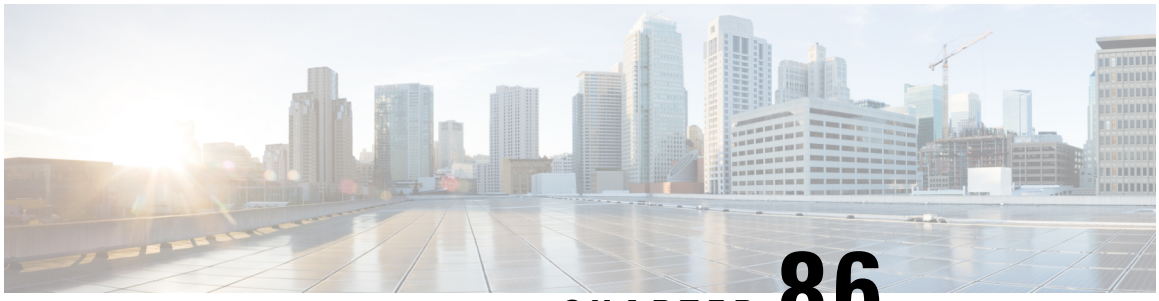
Feature Information for SNMP Cache Engine Enhancement

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 218: Feature Information for SNMP Cache Engine Enhancement

Feature Name	Releases	Feature Information
SNMP Cache Engine Enhancement	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 86

Onboard Failure Logging

Onboard Failure Logging (OBFL) captures and stores hardware failure and environmental information into nonvolatile memory. OBFL permits improved accuracy in hardware troubleshooting and root cause isolation analysis. Stored OBFL data can be retrieved in the event of a router crash or failure.

- Finding Feature Information, on page 1337
- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1337
- Understanding OBFL, on page 1338
- Configuring OBFL, on page 1339
- Displaying OBFL Logging Information, on page 1339
- Clearing OBFL Logging, on page 1339
- Configuration and Verification Examples, on page 1340
- Feature Information for Onboard Failure Logging, on page 1342

Finding Feature Information

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 219: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Understanding OBFL

OBFL provides a mechanism to store hardware, software, and environment related critical data in a non-volatile memory, such as flash EPROM or EEPROM on routers. The logging information is used by the TAC team to troubleshoot and fix hardware issues.

OBFL collects data like temperatures and voltages. It stores the data in a dedicated area of the flash memory of the router. This data is retrieved by TAC personnel to troubleshoot routers. It can also be analyzed by back-end software to detect failure patterns, and possibly to recommend specific quality improvements.

Retrieval of the OBFL message

If the hardware is defective and the system cannot boot up, any data in flash is inaccessible. In that case, use any one of the following methods to recover OBFL data:

- Read the flash through JTAG: this requires provisions in hardware design and back-end hardware and software support tools.
- Repair the system; boot it; use the OBFL CLI commands.

Configuring OBFL

Use the **hw-module** {**all**|**slot**|**module**} {*slotnumber*/*subslotnumber*|*modulenumber*} **logging onboard** {**disable** | **enable**} command to enable or disable OBFL on a specified hardware module.

Note OBFL is enabled by default.

```
Router# hw-module slot R0 logging onboard enable
```

Displaying OBFL Logging Information

Use the **show logging onboard** {**slot**|**module**|**bay**} {*slotnumber*/*subslotnumber*|*modulenumber*} {**dram** | **message** | **serdes** | **status** | **temperature** | **uptime** | **voltage**] command to view the OBFL log information.

Note OBFL is enabled by default on the Cisco cBR series router.

For the card PICs, use the **show logging onboard bay** *slotnumber*/*subslotnumber* {**dram** | **message** | **serdes** | **status** | **temperature** | **uptime** | **voltage**] command to view its OBFL information.

Clearing OBFL Logging

Use the **clear logging onboard** {**slot**|**module**} {*slotnumber*/*subslotnumber*|*modulenumber*} {**dram** | **message** | **serdes** | **temperature** | **voltage**} command to clear OBFL logging.

Following example shows how to clear DRAM ECC error log:

```
Router# clear logging onboard slot R0 dram
```

Following example shows how to clear OBFL error message:

```
Router# clear logging onboard slot R0 message
```

Following example shows how to clear onboard serdes log:

```
Router# clear logging onboard slot R0 serdes
```

Following example shows how to clear onboard temperature log:

```
Router# clear logging onboard slot R0 temperature
```

Following example shows how to clear onboard voltage log:

```
Router# clear logging onboard slot R0 voltage
```

Configuration and Verification Examples

Example—Verifying OBFL Configuration Status

```
Router#show logging onboard slot R1 status
Status: Enabled

Router#show logging onboard slot 5 status
Status: Disabled
```

Example—Displaying OBFL Logs

```
Router#show logging onboard slot R1 message
timestamp            module    sev  message

------------------------------------------------------------------------------------------------------------------------

01/01/12 12:00:23    SUP_PSOC  3    SUP MB PSOC alert interrupt

01/01/12 12:00:23    SUP_PSOC  3    SUP MB PSOC alert interrupt

01/01/12 12:00:23    SUP_PSOC  3    SUP MB PSOC alert interrupt

01/01/12 12:00:23    SUP_PSOC  3    SUP MB PSOC alert interrupt

01/01/12 12:01:15    SUP_PSOC  3    SUP MB PSOC alert interrupt
```

```
Router#show logging onboard slot R1 voltage
Name              Id    Data (mV)  Poll      Last Update
----------------------------------------------------------------------
PSOC-MB2_20: VO   40         1791  1         01/01/12 17:03:03
PSOC-MB2_21: VO   41         3290  1         01/01/12 17:03:03
PSOC-MB2_22: VO   42         3293  1         01/01/12 17:03:03
PSOC-MB2_23: VO   43         3299  1         01/01/12 17:03:03
PSOC-MB2_24: VO   44         4958  1         01/01/12 17:03:03
PSOC-MB2_25: VO   45         4508  1         01/01/12 17:03:03
PSOC-MB3_0: VOU   46         4999  1         01/01/12 17:03:03
PSOC-MB3_1: VOU   47         4982  1         01/01/12 17:03:03
PSOC-MB3_2: VOU   48         1499  1         01/01/12 17:03:03
PSOC-MB3_3: VOU   49         1193  1         01/01/12 17:03:03
PSOC-MB3_4: VOU   50          708  1         01/01/12 17:03:03
PSOC-MB3_5: VOU   51          757  1         01/01/12 17:03:03
PSOC-MB3_6: VOU   52          585  1         01/01/12 17:03:03
PSOC-MB3_7: VOU   53         1501  1         01/01/12 17:03:03
```

```
Router#show logging onboard slot R1 temperature
Name              Id     Data (C)  Poll      Last Update
----------------------------------------------------------------------
Temp: BB_DIE      159          25  1         01/02/12 23:04:19
Temp: VP_DIE      160          21  1         01/02/12 23:04:19
Temp: RT-E_DIE    161          29  1         01/02/12 23:04:19
Temp: INLET_1     162          20  1         01/02/12 23:04:19
Temp: INLET_2     163          18  1         01/02/12 23:04:19
Temp: OUTLET_1    164          22  1         01/02/12 23:04:19
Temp: 3882_1      165          44  1         01/02/12 23:04:19
```

```
Temp: 3882_1A     166          38  1         01/02/12 23:04:19
Temp: 3882_1B     167          36  1         01/02/12 23:04:19
Temp: 3882_2      168          38  1         01/02/12 23:04:19
Temp: 3882_2A     169          37  1         01/02/12 23:04:19
Temp: 3882_2B     170          35  1         01/02/12 23:04:19
Temp: 3882_3      171          38  1         01/02/12 23:04:19
```

```
Router#show logging onboard slot R1 uptime latest
 Slot              Reset reason  Power On
-------------------------------------------------------
    1      reset local software  01/02/12 23:02:46
```

```
Router#show logging onboard slot R1 uptime
 Slot              Reset reason  Power On
-------------------------------------------------------
    0      reset local software  01/06/12 01:52:26
    4      reset local software  01/06/12 01:52:42
    0      reset local software  01/06/12 01:52:45
    0      reset local software  01/06/12 02:20:27
    4      reset local software  01/06/12 02:20:43
    0      reset local software  01/06/12 02:20:46
    0      reset local software  01/06/12 05:12:02
    4      reset local software  01/06/12 05:12:19
    0      reset local software  01/06/12 05:12:22
    0      reset local software  01/06/12 05:17:31
    4      reset local software  01/06/12 05:17:48
    0      reset local software  01/06/12 05:17:51
    0            reset power on  01/01/12 08:56:44
    4            reset power on  01/01/12 08:57:00
```

```
Router#show logging onboard bay 4/3 message
timestamp            module    sev  message

-------------------------------------------------------------------------------------------------------------------

01/02/12 08:14:22    RFSW-PIC  6    CAT1836E07Q:7.13:Initialize:3/1

01/02/12 08:20:42    RFSW-PIC  6    CAT1836E07Q:7.13:Initialize:3/1

01/02/12 09:13:23    RFSW-PIC  6    CAT1836E07Q:7.13:Initialize:3/1

01/02/12 09:42:33    RFSW-PIC  6    CAT1836E07Q:7.13:Initialize:3/1

01/02/12 11:56:09    RFSW-PIC  6    CAT1836E07Q:7.13:Initialize:3/1

01/02/12 12:27:23    RFSW-PIC  6    CAT1836E07Q:7.13:Initialize:3/1
```

```
Router#show logging onboard bay 5/3 message
timestamp            module    sev  message

-------------------------------------------------------------------------------------------------------------------

01/22/15 01:06:05    RFSW-PIC  6    JAB092709EL:7.35:Init--stby:3/1

01/22/15 01:19:01    RFSW-PIC  6    JAB092709EL:7.35:Init--stby:3/1

01/22/15 01:31:47    RFSW-PIC  6    JAB092709EL:7.35:Init--stby:3/1

01/22/15 01:44:38    RFSW-PIC  6    JAB092709EL:7.35:Init--stby:3/1

01/22/15 01:59:04    RFSW-PIC  6    JAB092709EL:7.35:Init--stby:3/1
```

```
01/22/15 02:12:07    RFSW-PIC  6    JAB092709EL:7.35:Init--stby:3/1


Router#show logging onboard bay 4/4 message
timestamp            module    sev  message

------------------------------------------------------------------------------------------------------------------------

01/01/12 10:01:44    SUP-PIC   0    TEST1122334:0.130:PLL-LOS:1[04]

01/01/12 10:01:45    SUP-PIC   0    TEST1122334:0.130:PLL-LOS:2[04]

01/01/12 10:01:46    SUP-PIC   0    TEST1122334:0.130:PLL-LOS:3[04]

01/01/12 10:01:49    SUP-PIC   0    TEST1122334:0.130:PLL-LOS:4[04]

01/01/12 10:01:50    SUP-PIC   0    TEST1122334:0.130:PLL-LOS:5[04]

01/01/12 10:01:51    SUP-PIC   0    TEST1122334:0.130:PLL-LOS:6[04]


Router#show logging onboard bay 5/5 message
timestamp            module    sev  message

------------------------------------------------------------------------------------------------------------------------

01/03/12 13:52:55    SUP-PIC   0    TEST8877665:0.130:PLL-LOS:1[04]

01/03/12 13:52:56    SUP-PIC   0    TEST8877665:0.130:PLL-LOS:2[04]

01/03/12 13:52:57    SUP-PIC   0    TEST8877665:0.130:PLL-LOS:3[04]

01/03/12 13:53:00    SUP-PIC   0    TEST8877665:0.130:PLL-LOS:4[04]

01/03/12 13:53:01    SUP-PIC   0    TEST8877665:0.130:PLL-LOS:5[04]
```

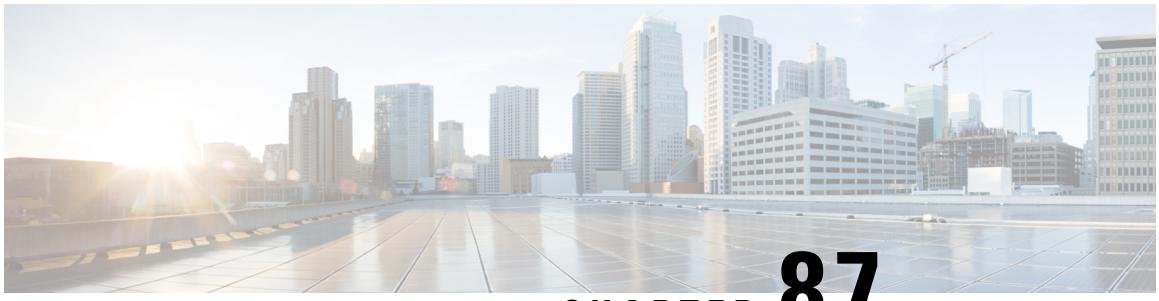
Feature Information for Onboard Failure Logging

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 220: Feature Information for Onboard Failure Logging

Feature Name	Releases	Feature Information
Onboard Failure Logging	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 87

Control Point Discovery

This document describes the Control Point Discovery (CPD) feature. This feature, along with Network Layer Signaling (NLS), enables automatic discovery of any control point associated with an end point.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1343
- Prerequisites for Control Point Discovery, on page 1344
- Restrictions for Control Point Discovery, on page 1344
- Information About Control Point Discovery, on page 1345
- How to Configure CPD, on page 1347
- Additional References, on page 1351
- Feature Information for Control Point Discovery, on page 1352

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 221: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Control Point Discovery

No special equipment or software is needed to use the Control Point Discovery feature.

Restrictions for Control Point Discovery

- The CPD feature does not sync any dynamic CPD/NLS related data between the route processors (RPs). After sending a NLS challenge to the controller, the new active PRE will ignore the NLS response as a result of any RP switchover.
- The CPEs become inaccessible for a small duration during line card switchovers. During this interval, any CPD request received on CMTS will be responded to as if the endpoint is not connected or as if the control relationship is not supported.
- The CPD functionality is restricted to default VPN table id (0).
- Only manual configuration of NLS authentication pass phrase would be supported for CPD/NLS security.
- For NLS authentication, HMAC SHA1 (no configuration option) is used with MAC length truncated to 96 bits.

Information About Control Point Discovery

To configure the Control Point Discovery feature, you should understand the following concepts:

Control Points

Control points are points in a network that can be used to apply certain functions and controls for a media stream. In a cable environment, the control points are Cable Modem Termination Systems (CMTS) and devices that utilizes these control points are referred to as CPD Requestors (or controllers).

Cable CPD Requestors include the following:

- Call Management Server (CMS)
- Policy Server (PS)
- Mediation Device for Lawful Intercept (MD)

Network Layer Signaling (NLS)

Network Layer Signaling (NSL) is an on-path request protocol used to carry topology discovery and other requests in support of various applications. In the CPD feature, NLS is used to transport CPD messages.

NLS for CPD

NLS is used to transport CPD messages. The CPD data is carried under an application payload of the NLS and contains a NLS header with flow id. The NLS flow id is used during NLS authentication to uniquely identify the CPD requests and responses for an end point of interest.

NLS Flags

All NLS headers contain bitwise flags. The CMTS expects the following NLS flag settings for CPD applications:

- HOP-BY-HOP = 0
- BUILD-ROUTE = 0
- TEARDOWN = 0
- BIDIRECTOINAL = 0
- AX_CHALLANGE = 0/1
- AX_RESPONSE = 0/1

Note Any requests with flags other then AX flags, set to one will be rejected with an error indicating a poorly formed message.

NLS TLVs

The following NLS TLVs are supported for all CPD applications:

- APPLICATION_PAYLOAD
- IPV4_ERROR_CODE
- IPV6_ERROR_CODE

- AGID
- A_CHALLENGE
- A_RESPONSE
- B_CHALLENGE
- B_RESPONSE
- AUTHENTICATION
- ECHO

The following NLS TLVs are not supported for CPD applications:

- NAT_ADDRESS
- TIMEOUT
- IPV4_HOP
- IPV6_HOP

Control Point Discovery

The control point discovery feature allows CPD Requestors to determine the control point IP address between the CPD Requestor and the media endpoint.

Using Networking Layer Signaling (NLS), the control point discovery feature sends a CPD message towards the end point (MTA). The edge/aggregation device (CMTS), located between the requestor and the endpoint, will respond to the message with its IP address.

Note For Lawful Intercept, it is important that the endpoint does not receive the CPD message. In this instance, the CMTS responds to the message without forwarding it to its destination.

CPD Protocol Hierarchy

CPD messages are sent over the NLS.

The CPD Protocol Hierarchy is as follows:

1. CPD
2. NLS
3. UDP
4. IP

Note Since NLS is implemented on the UDP protocol, there is a potential of message loss. If messages are lost, the controller will re-send the CPD request in any such event.

Control Relationship

A control relationship between a control point and a controller is identified as a function on a media flow that passes through a control point. A control relationship is uniquely defined by a control relationship type (CR TYPE) and control relationship ID (CR ID). The CR ID is provisioned on CMTS as well as the controller.

The table lists the supported CR TYPEs and corresponding pre-defined CR IDs

Table 222: Supported Control Relationship Types and Corresponding Control Relationship IDs

Control Relationship Type	Pre-Defined Corresponding Control Relationship ID
CR TYPE = 1 (Lawful Intercept)	CR ID = 1: CMTS
	CR ID = 2: Aggregation router or switch in front of CMTS
	CR ID = 3: Aggregation router or switch in front of Media Services
	CR ID = 4: Media Gateway
	CR ID = 5: Conference Server
	CR ID = 6: Other
CR TYPE = 2 (DQoS)	CR ID = 1: CMTS
CR TYPE = 3 (PCMM)	CR ID = 1: CMTS

How to Configure CPD

Enabling CPD Functionality

To enable the CPD functionality, use the cpd command in global configuration mode. The CPD message authentication is determined by NLS configuration.

Before you begin

The CPD message authentication is determined by NLS configuration.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cpd** **Example:**	Enables CPD functionality • Us the “no” form of this command to disable CPD functionality.

	Command or Action	Purpose
	`Router (config)# cpd`	
Step 4	**end** **Example:** `Router# end`	Exits global configuration mode and enters privileged EXEC mode.

Examples for CPD Enable

The following example shows the cpd enabled on a router:

```
Router (config)# cpd
```

Debugging CPD Functionality

To debug the CPD feature, use the **debug cpd** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

Configuring Control Relationship Identifier

To configure a Control relationship identifier (CR ID) for CMTS, use the cpd cr-id command. When CPD request comes with a wild-card CR ID, the CMTS will respond with this configured value.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cpd cr-id** **Example:** `Router (config)# cpd cr-id 100`	Configures a control relationship identifier (CR ID) for CMTS.

	Command or Action	Purpose
Step 4	**end** **Example:** `Router# end`	Exits global configuration mode and enters privileged EXEC mode.

Examples

The following example shows the cpd cr-id command configured with a cr-id number of 100 on a router.

```
Router (config)# cpd cr-id 100
```

Enabling NLS Functionality

To enable the NLS functionality, use the nls command in global configuration mode. It is recommended that NLS message authentication be enabled at all times.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**nls** **Example:** `Router (config)# nls`	Enables NLS functionality. • NLS authentication is optional. • It is recommended that NLS message authentication be enabled at all times.
Step 4	**end** **Example:** `Router# end`	Exits global configuration mode and enters privileged EXEC mode.

Examples

The following example shows the nls command enbaled on a router.

```
Router (config)# nls
```

Debugging NLS Functionality

To debug the NLS feature, use the **debug nls** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

Configuring Authorization Group Identifier and Authentication Key

The Authorization Group Identifier (AG ID) and corresponding authorization key are provisioned on CMTS, as well as on controller/CPD requester.

To configure the Authorization Group Identifier and Authentication Key, use the nls ag-id command in global configuration mode. It is recommended that NLS message authentication be enabled at all times.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**nls ag-id** **Example:** `Router (config)# nls ag-id 100 auth-key 20`	Configures the Authorization Group Identifier and Authentication Key.
Step 4	**end** **Example:** `Router# end`	Exits global configuration mode and enters privileged EXEC mode.

Examples

The following example shows the nls ag-id command with an Authorization Group ID of 100 and Authentication Key of 20.

```
Router (config)# nls ag-id 100 auth-key 20
```

Configuring NLS Response Timeout

The NLS response timeout governs the time CMTS will wait for getting a response for a NLS authentication request.

To configure the NLS response timeout, use the nls ag-id command in global configuration mode. It is recommended that NLS message authentication be enabled at all times.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**nls resp-timeout** **Example:** `Router (config)# nls resp-timeout 60`	Configures the NLS response time.
Step 4	**end** **Example:** `Router# end`	Exits global configuration mode and enters privileged EXEC mode.

Examples

The following example shows the nls resp-timeout command with a response timeout setting of 60 seconds.

```
Router (config)# nls resp-timeout 60
```

Additional References

The following sections provide references related to the CPD feature.

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

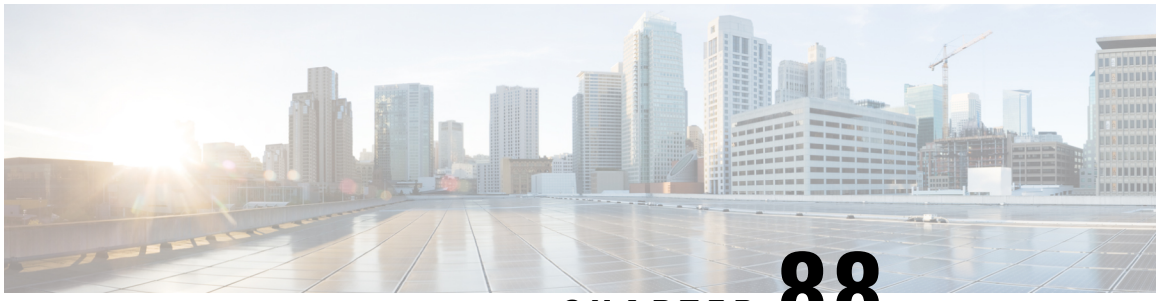
Feature Information for Control Point Discovery

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 223: Feature Information for Control Point Discovery

Feature Name	Releases	Feature Information
Control Point Discovery	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 88

IPDR Streaming Protocol

The Cisco cBR Series Converged Broadband Router supports the Internet Protocol Detail Record (IPDR) streaming protocol feature that provides high volume data exported from the network equipment to mediation systems such as the Operations Support Systems (OSS) or Business Support Systems (BSS). IPDR provides information about IP-based service usage and other activities that are used by OSS and BSS. This protocol provides a mechanism to collect data from various network elements or equipment using a push model as opposed to the conventional Simple Network Management Protocol (SNMP) polling mechanism.

Based on the DOCSIS 3.0 specifications, the IPDR feature optimizes time and resource efficiency in the transfer of large amounts of performance metrics to the management systems. DOCSIS 3.0 introduces five management features or the FCAPS model. FCAPS represents Fault, Configuration, Accounting, Performance and Security.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Restrictions for Configuring IPDR Streaming Protocol, on page 1353
- Information About IPDR Streaming Protocol, on page 1354
- How to Configure IPDR Streaming Protocol, on page 1355
- Configuration Examples for IPDR Streaming Protocol, on page 1360
- Verifying IPDR Streaming Protocol, on page 1361
- Additional References, on page 1363
- Feature Information for IPDR Streaming Protocol , on page 1363

Restrictions for Configuring IPDR Streaming Protocol

- An IPDR exporter can be connected to many collectors, but it will only send data to the highest priority operating collector at any given time.

- Each IPDR session can be associated to one active (zero) or more standby collector with priority.

Information About IPDR Streaming Protocol

IPDR Streaming Protocol is designed to address the need for a reliable, fast, efficient, and flexible export process of high volume data records such as billing, performance and diagnostic data.

The IPDR/SP process communicates with IPDR collectors. The IPDR streaming protocol supports multiple IPDR sessions. The architecture supports primary and secondary collectors for failover purposes. At any time, data is sent to only one collector. If the exporter to primary collector connection fails due to any reason, the data is sent to the secondary collector. Depending on the network configuration, you can have only one primary collector for each session, while for different sessions, you can have different primary collectors. For example, there may be a billing collector, a diagnostic collector, and so on.

Note IPDR exporter refers to the Cable Modem Termination System (CMTS) and the IPDR collector refers to the network equipment.

Data Collection Methodologies

IPDR is the data generated or collected for various performance related metrics such as billing information, diagnostics, network topology, signal quality monitoring, and other management data. These data are based on the FCAPS model (Fault, Configuration, Accounting, Performance and Security.)

The IPDR client application communicates with the IPDR exporter using the IPDR_GET_SESSIONS message to identify the streams provided by the exporter, and the exporter sends responses to the client using the IPDR_GET_SESSIONS_RESPONSE message. This data collection method is based on the *Operations Support System Interface Specification* (CM-SP-OSSIv3.0-I13-101008).

The IPDR_GET_SESSIONS_RESPONSE message includes the SessionBlock.reserved attribute to identify the IPDR session ID. This attribute helps the Cisco CMTS router define an IPDR session ID for each data collection mechanism supported for each IPDR service definition. This attribute was not used in Cisco IOS Releases earlier to Cisco IOS Release 12.2(33)SCE.

The IPDR feature defines methods for the collectors or network elements to collect data from the CMTS. Below is the list of collection methodologies:

Time Interval Session: In this method, the CMTS follows a schedule-based session to stream data at a periodic time interval. A time interval is the time gap between two adjacent sessions' start messages. This method is managed by the CMTS in controlling the start and stop operation of a session. The time interval session terminates after the CMTS exports the records.

Note During the course of a one-time interval when the CMTS is streaming records, if another time interval is expected, the CMTS will ignore the new time interval and continue exporting the data until the previous time interval ends.

Event-based Session: In this method, the CMTS can export records at any time, when the session is open. In other words, this method works on an open-ended session.

Ad-hoc Session: In this method, the CMTS creates a session, allows data streaming, and closes the session when the data export is complete or when a closing command is generated.

A new session is created by issuing the **ipdr session** command. After, the CMTS receives the FLOW_START message from the collector, the CMTS exporter sends a SESSION_START message to start exporting the IPDR data from the collector. After all data is transported, the exporter receives a ACK message from the collector, and then sends a SESSION_STOP message to the collector. This method is known as the Ad-hoc session.

How to Configure IPDR Streaming Protocol

This section describes the configuration tasks that are performed when using the IPDR streaming protocol feature on the Cisco CMTS platforms.

Note Use no ipdr command to remove the IPDR configuration.

Configuring the IPDR Session

To enable the CMTS application to add a session to the IPDR exporter, use the ipdr session command in global configuration mode.

Use the no form of the command to remove the IPDR session.

Note
- The session ID must be unique.
- To remove an active session, you must deactivate it before removing it.

>

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	ipdr session session_id session_name session_descr **Example:**	Enables the CMTS application to add a session to the IPDR exporter.

	Command or Action	Purpose
	`Router(config)# ipdr session 1 samis_sxn test`	

Configuring the IPDR Type

To configure the IPDR session type, use the ipdr type command in global configuration mode. The IPDR session types that can be defined using this command are event type, time-interval type, and the ad hoc type.

Use the no form of the command to reset the session type to the default "event" type.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	ipdr type session_id [ad-hoc | event | time-interval value] **Example:** `Router(config)# ipdr type 1 time-interval 15`	Enables the CMTS application to configure an IPDR session type.

What to do next

Note Once the IPDR session type is configured, only the templates supported by this IPDR type are allowed be associated with it. Also, the console provides information about those templates that are not supported by this IPDR session type when the type is changed.

Configuring the IPDR Collector

To configure the IPDR collector details, use the ipdr collector command in global configuration mode. The port number is used when an exporter creates an active connection.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipdr collector** **Example:** `Router(config)# ipdr collector federal 192.168.6.5`	Enables the CMTS application to configure an IPDR collector and authenticate the IPDR protocol. **Note** Configure the NAT address in case of an IPDR collector that is operating in a NAT enabled network.

Configuring the IPDR Associate

To associate the collector with a session, use the ipdr associate command in global configuration mode.

Before you begin

- You must deactivate the session before configuring the associate.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipdr associate session_id collector_name priority** **Example:**	Associates the collector with a session.

	Command or Action	Purpose
	`Router(config)# ipdr associate 1 federal 1`	

Configuring the IPDR Template

To add an IPDR template to the IPDR session, use the ipdr template command in global configuration mode. The template list can be viewed by entering a "?" at the command prompt.

Note
- You can add only the system-supported templates.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipdr template session_id template_name** **Example:** `Router(config)# ipdr template 1 SAMIS`	Adds an IPDR template to the IPDR session.

Configuring the IPDR Exporter

IPDR exporter parameters such as keepalive timer count, the maximum number of unacknowledged records, and unacknowledged timeout interval value can be configured using the following commands.

- **ipdr exporter keepalive**—Sets the keepalive timer count value on the IPDR Exporter.
- **ipdr exporter max-unacked**—Sets the maximum number of unacknowledged records on the IPDR Exporter.
- **ipdr exporter ack-timeout**—Sets the time interval for acknowledged records on the IPDR Exporter.

Note The default value for DataAckTimeInterval is 60 seconds and the default value for DataAckSequenceInterval is 200 seconds.

You can set the values for the IPDR parameters to customize exporter for the collectors used in the facility. However, these commands are optional, so if not configured, the default values of the commands are used when **ipdr exporter start** command is executed.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipdr exporter keepalive** *time_interval* **Example:** `Router(config)# ipdr exporter keepalive 300`	(Optional) Sets the keepalive timer count for the IPDR Exporter. The valid range is from 5 to 300 seconds. The default value is 300.
Step 4	**ipdr exporter max-unacked** *records* **Example:** `Router(config)# ipdr exporter max-unacked 200`	(Optional) Sets the number of maximum unacknowledged records on the IPDR Exporter. The valid range is from 5 to 200 records. The default value is 200.
Step 5	**ipdr exporter ack-timeout** *time_interval* **Example:** `Router(config)# ipdr exporter ack-timeout 60`	(Optional) Sets the acknowledged records timeout interval on the IPDR Exporter. The valid range is from 5 to 60 seconds. The default value is 60.
Step 6	ipdr exporter start **Example:** `Router(config)# ipdr exporter start`	Enables the CMTS application to start the IPDR exporter process to connect the exporter and the collector.

Configuration Examples for IPDR Streaming Protocol

Example: Configuring the IPDR Session

The following example shows how to configure the IPDR session.

```
Router> enable
Router# configure terminal
Router(config)# ipdr session 1 test no_descr
```

Example: Configuring the IPDR Type

The following example shows how to configure the IPDR “time-interval” session type for a time interval of 15 minutes.

```
Router> enable
Router# configure terminal
Router(config)# ipdr type 1 time-interval 15
```

Example: Configuring the IPDR Collector

The following example shows how to configure the IPDR collector.

```
Router> enable
Router# configure terminal
Router(config)# ipdr collector federal 209.165.200.225
```

Example for Configuring the IPDR Collector with NAT Address

This example shows the **nat-address** keyword used to configure the NAT address for an IPDR collector:

```
Router(config)#ipdr collector federal 192.0.2.225 nat-address 192.0.2.51
```

Example: Configuring the IPDR Associate

The following example shows how to associate the collector with a session.

```
Router> enable
Router# configure terminal
Router(config)# ipdr associate 1 federal 1
```

Example: Configuring the IPDR Template

The following example shows how to add an IPDR template to the IPDR session.

```
Router> enable
Router# configure terminal
Router(config)# ipdr template 1 SAMIS-TYPE1
```

Example: Configuring the IPDR Exporter

The following example shows how to configure the IPDR exporter process to connect the exporter and the collector.

```
Router> enable
Router# configure terminal
Router(config)# ipdr exporter keepalive 300
Router(config)# ipdr exporter max-unacked 200
Router(config)# ipdr exporter ack_timeout 60
Router(config)# ipdr exporter start
```

Verifying IPDR Streaming Protocol

This section describes the commands used for verification of the IPDR streaming protocol feature on the Cisco CMTS platforms.

Verifying the IPDR Collector

The **show ipdr collector** command displays the collector information, message statistics, and event for all the sessions that are associated with the collector.

The following example shows the sample output for the **show ipdr collector** command.

```
Router# show ipdr collector federal
Collector Name: federal, IP: 192.0.2.0, Port: 0
2001-07-05T19:28:22 Collector in session 1 Statistics:
 Transmitted 12658 Acknowledged 12658 Enqueued 12658 Lost 0
 Last Event: Event Id 1 IPDR_EVENT_SERVER_CONNECTED - INCOMING
Router(config)#
```

Verifying IPDR exporter

The **show ipdr exporter** command displays information about the IPDR Exporter state as listed below.

- started
- not started
- not initialized

The following example shows the sample output for the **show ipdr exporter** command:

```
Router# show ipdr exporter
IPDR exporter is started.
Current parameters:
        KeepAliveInterval   :300
        AckTimeInterval     :60
        AckSequenceInterval :200
Router#
```

Verifying IPDR session

The **show ipdr session** command displays the session details such as the session ID, description, and the session state for all sessions as well as for a specific session.

The following example shows the sample output for the **all** keyword for the **show ipdr session** command.

```
Router# show ipdr session all
Session ID: 1, Name: utilsta, Descr: test, Started: False
```

The following example shows the sample output for the **session_id** keyword for the **show ipdr session** command.

```
Router# show ipdr session 1
Session ID: 1, Name: utilsta, Descr: test, Started: False
2001-07-05T19:36:28 Statistics:
Transmitted 0 Acknowledged 0 Enqueued 0 Lost 0
queuedOutstanding 0 queuedUnacknowledged 0
1 Collectors in the session:
Name: federal, IPAddr: 192.0.2.0, Port: 0, Priority: 1
```

Verifying IPDR Session Collector

The **show ipdr session collector** command displays the details of a collector that is associated with a specific session. Because there can be multiple collectors associated to a session, this command is used to show a specific session-collector pair.

The following example shows the sample output for the **show ipdr session collector** command.

```
Router# show ipdr session 1 collector federal
Session ID: 1, Name: utilsta, Descr: test, Started: False
Collecotr Name: federal, IP: 192.0.2.0, Port: 0
2001-07-05T19:38:02 Collector in session 1 Statistics:
 Transmitted 0 Acknowledged 0 Enqueued 0 Lost 0
 Last Event: Event Id 0 WRONG_EVENT_ID
```

Verifying IPDR Session Template

The **show ipdr session template** command displays the list of all active templates supported by a specific session.

The following example shows the sample output for the **show ipdr session template** command.

```
Router# show ipdr session 1 template
Template ID: 2, Name:
http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-CMSERVICE-FLOW-TYPE,
Type: DOCSIS-Type, KeyNumber: 22
Session 1 has totally 1 templates.
```

Additional References

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

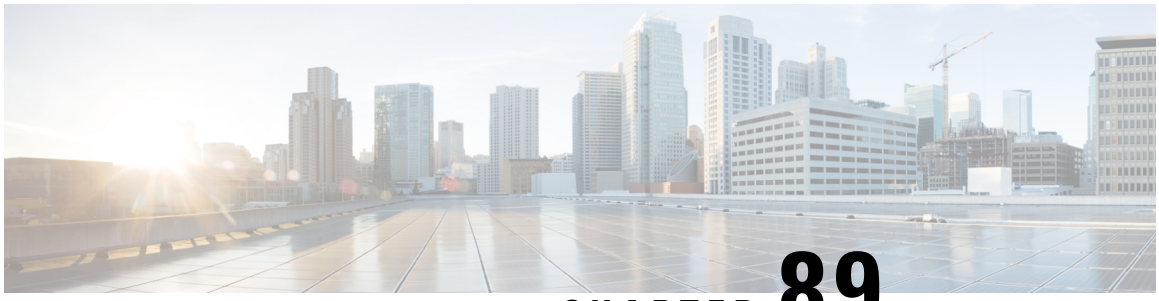
Feature Information for IPDR Streaming Protocol

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 224: Feature Information for Downstream Interface Configuration

Feature Name	Releases	Feature Information
IPDR Streaming Protocol	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Router .



CHAPTER 89

Usage-Based Billing (SAMIS)

This document describes the Usage-based Billing feature for the Cisco Cable Modem Termination System (CMTS) routers, which provides subscriber account and billing information in the Subscriber Account Management Interface Specification (SAMIS) format. The SAMIS format is specified by the Data-over-Cable Service Interface Specifications (DOCSIS) Operations Support System Interface (OSSI) specification.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1365
- Prerequisites for Usage-Based Billing (SAMIS), on page 1366
- Restrictions for Usage-based Billing, on page 1367
- Information About Usage-based Billing, on page 1368
- How to Configure the Usage-based Billing Feature, on page 1378
- Monitoring the Usage-based Billing Feature, on page 1404
- Configuration Examples for Usage-based Billing, on page 1406

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 225: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Usage-Based Billing (SAMIS)

The Usage-based Billing feature has the following prerequisites:

- Cable modems must be compliant with DOCSIS 1.0 or DOCSIS 2.0, OSSI version 3.0 and DOCSIS 3.0.
- Cable modems that are being monitored should use a DOCSIS configuration file that defines upstream and downstream primary service flows using Service Class Naming (SCN [TLV 24/25, subTLV 4]). If dynamically-created service flows are to be monitored, they should also be created with SCN names.
- When the feature is operating in File mode, an external billing server must log into the Cisco CMTS to copy the billing records to the external server, using either Secure Copy (SCP) or Trivial File Transfer Protocol (TFTP). The Cisco CMTS cannot operate as a FTP or secure FTP (SFTP) server.
- When the feature is operating in Streaming mode in non-secure mode, an external billing server must be configured to receive the billing records at a configurable TCP port.
- When the feature is operating in Streaming mode in secure mode, the following are required:
  - The external billing server must be configured to receive the billing records at a configurable TCP port using a secure socket layer (SSL) connection.

Tip Several third-party solutions for SSL support on the billing application server are available http://www.openssl.org/index.html.

- • A Certificate Authority (CA) must be configured and available to provide the required digital certificates to the billing application and Cisco CMTS router. The CA can be a public CA, such as Verisign, or a server on your private management network that is running software such as the Cisco Provisioning Center (CPC).

- To use the **full-records** keyword, the Cisco CMTS router must be running the Cisco IOS-XE releases.
- To use the **flow-aggregate** keyword for ipdr/ipdr-d3 the Cisco CMTS router must be running the Cisco IOS-XE releases.

When **flow-aggregate** is enabled, the service flows are combined into one record per cable modem:

- • ServiceClassName element always returns a null value in IPDR records, even when service flows on the cable modem have a valid service class name.
  - ServiceIdentifier element always returns a zero value.

Restrictions for Usage-based Billing

The Usage-based Billing feature has the following restrictions and limitations:

- SNMP commands can be used to display or modify the Usage-based Billing configuration, and SNMP traps can be used to notify the billing application system when a billing record is available. However, SNMP commands cannot be used to retrieve billing records.
- Enabling IPDR mode through SNMP is not supported.

During a line card switchover, the items in the line card side are lost. Similarly, during a PRE switchover, those items in the RP side of the sflog file are lost.

If the user uses the SAMIS file destination, a PRE switchover also reinitializes that output file

- Billing records do not include information about multicast service flows and traffic counters.
- The packet counters displayed by CLI commands are reset to zero whenever the Cisco CMTS router is rebooted. The packet counters displayed by SNMP commands are not retained across router reloads, and SNMP MIB counters cannot be preserved during reloads. These counters are 64-bit values and could roll over to zero during periods of heavy usage.
- When configuring cable metering in the usage-based billing File Mode, the source-interface cannot be specified immediately after using the cable metering filesystem command. Once the cable metering filesystem command is used, the cable metering file will write to the bootflash. Until this operation is complete, no cable metering configuration will be allowed. After the file write operation is complete, the source-interface command (cable metering source-interface) can then be configured; and the metering file in the bootflash would need to be removed so that billing packets have the source-interface's IP address.

Note This cable metering restriction will not be a problem during reload.

- When configuring cable metering in the usage-based billing Streaming Mode, make sure that the loopback interface is accessible from the collector server. Telnetting to the IP address of the loopback interface from the collector server is a good method of testing whether the loopback interface is accessible from the collector server or not.

Information About Usage-based Billing

Feature Overview

The Usage-based Billing feature provides a standards-based, open application approach to recording and retrieving traffic billing information for DOCSIS networks. When enabled, this feature provides the following billing information about the cable modems and customer premises equipment (CPE) devices that are using the cable network:

- IP and MAC addresses of the cable modem.
- Service flows being used (both upstream and downstream service flows are tracked).
- IP addresses for the CPE devices that are using the cable modem.
- Total number of octets and packets received by the cable modem (downstream) or transmitted by the cable modem (upstream) during the collection period.
- Total number of downstream packets for the cable modem that the CMTS dropped or delayed because they would have exceeded the bandwidth levels allowed by the subscriber's service level agreement (SLA).

Billing records are maintained in a standardized text format that the service provider can easily integrate into their existing billing applications. Service providers can use this information to determine which users might be potential customers for service upgrades, as well as those customers that might be trying to exceed their SLA limits on a regular basis.

Usage-Based Billing and DOCSIS Support on the Cisco CMTS Routers

The usage-based billing feature supports these DOCSIS features on the Cisco CMTS routers:

- DOCSIS 1.0, DOCSIS 2.0, and DOCSIS 3.0 compliant cable modems are supported.
- Best Effort service flows are supported for DOCSIS-compliant cable modems.
- Secondary service flows are supported for DOCSIS-compliant cable modems.
- Dynamic service flows are supported for DOCSIS-compliant cable modems.
- Information about deleted service flows is available only for DOCSIS 1.1 service flows but not for DOCSIS 1.0 service flows.
- Support for terminated service flows must be enabled using the **cable sflog** command in global mode.

Standards

The Usage-based Billing feature is based on several open standards, allowing it to be supported by a wide range of commercial and custom-written billing applications. The following standards provide the major guidelines for writing and using the billing records that the CMTS produces:

- Extensible Markup Language (XML)—A metalanguage that in turn can easily define other markup languages to contain any kind of structured information, such as billing records. An XML-based approach allows the collected billing information to be used by and distributed among many different billing applications from different vendors. It also allows the format to be easily updated and customized to meet the needs of different providers.
- IP Detail Record (IPDR)—An open, vendor-independent standard, defined in the *Network Data Management—Usage (NDM-U) For IP-Based Services* specification, to simplify billing and usage record-keeping for any type of services that can be delivered over an IP-based network. Service providers can use IPDR to create unified billing applications for all of their services, such as DOCSIS or Voice-over-IP, even though those services use different protocols and application servers.
- DOCSIS Operations Support System Interface (OSSI) specification—A DOCSIS specification that defines the requirements for the network management of a DOCSIS network, including a Subscriber Account Management Interface Specification (SAMIS) for a billing record interface. The DOCSIS 2.0 version of this specification states that a CMTS is not required to provide a billing interface, but if the CMTS does provide a billing interface, it must be based on the IPDR/XML standards.

Tip For further information about these standards, see the documents listed in the “Standards” section on page 38.

IPDR Service Definition Schemas

To standardize the management of objects, service definition schemas are associated with IPDR just as MIBs are associated to SNMP.

For more information, see the OSSI specification document at http://www.cablelabs.com/wp-content/uploads/specdocs/CM-SP-OSSIv3.0-I02-070223.pdf.

The schemas are supported on Cisco IOS-XE releases.

Table 226: IPDR Schema List for DOCSIS 3.0

Category	Service Definition	Schema Definition	Collection Method
SAMIS	SAMIS-TYPE-1	DOCSIS-SAMIS-TYPE-1_3.5.1-A.1.xsd	time interval, ad-hoc
	SAMIS-TYPE-2	DOCSIS-SAMIS-TYPE-2_3.5.1-A.1.xsd	time interval, ad-hoc

Category	Service Definition	Schema Definition	Collection Method
Diagnostic Log Service Definition Schemas	DIAG-LOG-TYPE	DOCSIS-DIAG-LOG-TYPE_3.5.1-A.1.xsd	ad-hoc
	DIAG-LOG-EVENT-TYPE	DOCSIS-DIAG-LOG-EVENT-TYPE_3.5.1-A.1.xsd	event
	DIAG-LOG-DETAIL-TYPE	DOCSIS-DIAG-LOG-DETAIL-TYPE_3.5.1-A.1.xsd	time interval, ad-hoc, event
Spectrum Management	SPECTRUM-MEASUREMENT-TYPE	DOCSIS-SPECTRUM-MEASUREMENT-TYPE_3.5.1-A.1.xsd	time interval, ad-hoc
CMTS CM Registration Status Information	CMTS-CM-REG-STATUS-TYPE	DOCSIS-CMTS-CM-REG-STATUS-TYPE_3.5.1-A.1.xsd	time interval, ad-hoc, event
CMTS CM Upstream Status Information	CMTS-CM-US-STATS-TYPE	DOCSIS-CMTS-CM-US-STATS-TYPE_3.5.1-A.1.xsd	time interval, ad-hoc
CMTS Topology	CMTS-TOPOLOGY-TYPE	DOCSIS-CMTS-TOPOLOGY-TYPE_3.5.1-A.1.xsd	ad-hoc, event
CPE Information	CPE-TYPE	DOCSIS-CPE-TYPE_3.5.1-A.1.xsd	ad-hoc, event
CMTS Utilization Statistics	CMTS-US-UTIL-STATS-TYPE	DOCSIS-CMTS-US-UTIL-STATS-TYPE_3.5.1-A.1.xsd	event
	CMTS-DS-UTIL-STATS-TYPE	DOCSIS-CMTS-DS-UTIL-STATS-TYPE_3.5.1-A.1.xsd	event

The schemas listed in the table are supported by implementing the respective Collectors, which work as SNMP agents to generate these IPDR records according to management information of the system.

IPDR CM-STATUS-2008

Cisco IOS-XE Release 16.5.1 supports the IPDR CM-STATUS 2008 version for forward compatibility to support old IPDR collectors. In the IPDR CM-STATUS 2008 version, the CmtsRcsId and CmtsTcsId objects are 16 bits in length whereas in the CM-STATUS version both these objects are 32 bits in length.

The CmtsRcsId object in the CM-STATUS-2008 version returns the lower 16 bits of value from the CM-STATUS version. But, the CmtsTcsId object returns the same value for both the CM-STATUS-2008 and CM-STATUS version since the value does not exceed 16 bits in both the schemas.

DOCSIS SAMIS Service Definitions

SAMIS for DOCSIS 3.0 service definitions are well structured and has two versions—SAMIS-TYPE-1 and SAMIS-TYPE-2 and provide a different level of information details than SAMIS.

DOCSIS 2.0 SAMIS supports only event session (default type) and DOCSIS 3.0 SAMIS TYPE 1 and DOCSIS 3.0 SAMIS TYPE 2 support only interval and ad-hoc sessions.

SAMIS is collected based on configurable time intervals. Each interval is a different document and the Exporter stops and starts a new session for a new interval. The interval starts from the last metering that has either succeeded or failed, unlike the time-interval session that has a fixed starting point and an interval.

Note The SAMIS schema can be configured with the **cable metering ipdr session** command SAMIS-TYPE-1 and SAMIS-TYPE-2 schemas can be configured through the **cable metering ipdr-d3** command. These schemas are mutually exclusive of each other.

Limitation To DOCSIS SAMIS

- Only schemas that are consistent with the **cable metering ipdr| ipdr-d3** command works. If none of the schemas are consistent, none of them work.
- Changing the SAMIS IPDR type cancels exporting IPDR data.

DOCSIS Diagnostic Log Service Definitions

This service definition defines the IPDR Streaming using a two-step process:

- SNMP or other configuration management interface, such as the CLI is used to configure the Diagnostic Log.
- IPDR/SP is used to stream the Diagnostic Log instances.

These Diagnostic Log service definition schemas support the following collection methods:

- The Cisco CMTS supports streaming of the DIAG-LOG-TYPE record collections as an ad-hoc session.
- The Cisco CMTS supports streaming of DIAG-LOG-EVENT-TYPE record collections as an event session. For event-based Diagnostic Log records, the Cisco CMTS streams the record when the event is logged in the Diagnostic Log and an IPDR message is transmitted to the Collector.
- The DOCSIS-DIAG-LOG-DETAIL-TYPE supports the following collection methods:
  - Time interval—Follows a schedule based on session configuration to export data on a periodic time interval. When a given time interval end is reached, the Exporter collects the diagnostic log, then streams the record to the Collector associated with this session. For time interval based Diagnostic Log records, the Cisco CMTS streams a snapshot of the Diagnostic Log at the scheduled collection time.
  - Ad-hoc—When the Exporter receives a "FlowStart" message, it triggers the application to collect the diagnostic record and send the data to the Collector.
  - Event—When a diagnostic log record is created, an ipdr message is transmitted to the Collector. For more information, see the Operations Support System Interface (OSSI) Specification.

DOCSIS Spectrum Measurement Service Definition

This service definition schema defines the IPDR schema for the enhanced signal quality monitoring feature.

The DOCSIS-SPECTRUM-MEASUREMENT-TYPE schema supports the following collection methods:

- Time interval—Follows a schedule based on session configuration to export data on a periodic time interval. When a given time interval end is reached, the Exporter collects the spectrum information, then streams the records to the Collector.
- Ad-hoc—When the Exporter receives a "FlowStart" message, it triggers the application to collect the spectrum information and send the data to the Collector.

DOCSIS CMTS CM Registration Status Service Definition

This service definition schema defines the IPDR service definition schema for the CMTS CM Registration Status information.

The DOCSIS-CMTS-CM-REG-STATUS-TYPE schema supports the following collection methods:

- Time interval—Follows a schedule based on session configuration to export data on a periodic time interval. When a given time interval end is reached, the Exporter collects the CM status information, then streams the records to the Collector.
- Ad-hoc—When the Exporter receives a "FlowStart" message, it triggers the application to collect all status information of the cable modems and send the data to the Collector.
- Event—When a cable modem goes from "offline" status to "online" or changes to "offline" from "online" (not including intermediate state changes), the Exporter invokes the application to collect the cable modem status information and sends the data to the Collector. For more information, see the Operations Support System Interface (OSSI) Specification.

DOCSIS CMTS CM Upstream Status Service Definition

This service definition schema define the cable modem registration status objects and upstream status objects from the cable modem and the Cisco CMTS perspective. In the CmtsCmUsEqData IPDR schema field, configure the **cable upstream equalization-coefficient** command under the corresponding MAC domain to enable the feature to have data. For more information on this command, see the Cisco IOS CMTS Cable Command Reference Guide .

The DOCSIS-CMTS-CM-US-STATS-TYPE schema support the following collection methods:

- Time interval—Follows a schedule based on session configuration to export data on a periodic time interval. When a given time interval end is reached, the Exporter collects the cable modem upstream status information, then streams the records to the Collector.
- Ad-hoc—When the Exporter receives a "FlowStart" message, it triggers the application to collect all upstream status information of the cable modem and send the data to the Collector.

DOCSIS CMTS Topology Service Definition

In the case of an event session, the event means a change of the topology.

This service definition schema defines the IPDR service definition schema for the CMTS Topology information.

The DOCSIS-CMTS-TOPOLOGY-TYPE schema supports the following collection methods:

- Ad-hoc—Sends the entire picture of all fiber-nodes.
- Event—Sends only the updated channels status of the fiber nodes.

DOCSIS CPE Service Definition

The DOCSIS-CPE-TYPE schema supports the following collection methods:

- Ad-hoc—Follows a schedule based on session configuration to export data on a periodic time interval. When a given time interval end is reached, the Exporter collects the CPE status information, then transfers the records to the Collector.
- Event—When new CPE is added, the status of the CPE changes (including change in IP address), or a new CPE replaces an old one (in this case, two messages are displaced— removal of the old CPE and addition of the new CPE). For more information, see the Operations Support System Interface (OSSI) Specification.

DOCSIS CMTS Utilization Statistics Service Definition

The CMTS Utilization Statistics mainly focuses on channel utilization. It covers CMTS MAC Domain, channel identifier, and the upstream or downstream utilization attributes and counters.

The DOCSIS-CMTS-US-UTIL-STATS-TYPE schemas defines upstream utilization statistics for a specified upstream logical channel interface for the specified Cisco CMTS. The interval can be configured through Channel Utilization Interval.

The DOCSIS-CMTS-DS-UTIL-STATS-TYPE schema defines downstream utilization statistics for a specified downstream interface for the specified Cisco CMTS. The interval can be configured through Channel Utilization Interval.

For more information, see the IPDR Streaming Protocol on the Cisco CMTS Routers guide at the following URL:

IPDR Streaming Protocol

These schemas support only interval-driven event session for the entire downstream and upstream. The interval is defined in the docsIfCmtsChannelUtilizationInterval MIB and it creates document for every exporting.

Note The UsUtilTotalCntnReqDataMslots, UsUtilUsedCntnReqDataMslots, and UsUtilCollCntnReqDataMslots MIBs are not supported on the Cisco CMTS implementation.

The DsUtilTotalBytes MIB for RF Gateway RF channels is the maximum counter of bytes this RF channel can pass during an interval.

Modes of Operation

The Usage-based Billing feature can operate in three modes:

- File Mode—In file mode, the CMTS collects the billing record information and writes the billing records to a file on a local file system, using a file name that consists of the router's hostname followed by a timestamp of when the file was written. A remote application can then log into the CMTS and transfer the billing record file to an external server where the billing application can access it.

The remote application can use the Secure Copy Protocol (SCP) or the Trivial File Transfer Protocol (TFTP) to transfer the file. After a successful transfer, the remote application then deletes the billing record file, which signals the CMTS that it can create a new file. The remote application can either periodically log into the CMTS to transfer the billing record file, or it can wait until the CMTS sends an SNMPv2 trap to notify the application that a billing record file is available.

- Streaming Mode—In streaming mode, the CMTS collects the billing record information and then regularly transmits the billing record file to an application on an external server, using either a non-secure TCP

connection or a secure sockets layer (SSL) connection. The billing record data collected is streamed in real time; and if streaming is unsuccessful, then the SAMIS data is sent only at the next interval.

If the CMTS fails to establish a successful connection with the external server, it retries the connection between one to three times, depending on the configuration. If the CMTS continues to fail to connect with the external server, the Cisco CMTS sends an SNMPv2 trap to notify the SNMP manager that this failure occurred.

In streaming mode, you can configure the CMTS to transmit the billing record file at regular intervals. Typically, the interval chosen would depend on the number of cable modems and the size of the billing record files that the CMTS produces.

- IPDR Mode—In the IPDR mode, the IPDR export process communicates with IPDR Collectors. The architecture supports multiple Collectors distinguished by priority value for failover purposes. The smaller the number of Collectors, the higher is the priority value. Associating one session to two or more Collectors with the same priority value is regarded as random priority. At any given time, data is sent to only the available highest priority Collector. If the highest priority Collector connection fails due to any reason, the data is sent to the next available highest priority Collector. After a higher priority Collector comes back online, it will fail over again. Depending on the network configuration, you can have different primary Collectors for different IPDR sessions. For example, there may be a billing Collector or a diagnostic Collector.

Billing Record Format

Each billing record is an ASCII text file using XML formatting to encode the billing record objects that are required by the DOCSIS specifications. This file can be read by any billing application that can be configured to parse XML data files.

The table lists the objects that are contained in each billing record that the CMTS generates. This table shows the object's name, as it appears in the billing record, and a description of that object.

Table 227: Billing Record Objects

Object Name	Description
IPDRcreationTime	(Appears in header of billing record) Date and time that the CMTS created the billing created.
serviceClassName	Service Class Name (SCN) identifying the service flow (for example, BronzeDS).
CMmacAddress	MAC Address of the cable modem, expressed as six hexadecimal bytes separated by dashes (for example, 00-00-0C-01-02-03).
CMipAddress	IP address for the cable modem, expressed in dotted decimal notation (for example, 192.168.100.101).
CMdocsisMode	Version of DOCSIS QoS provision that the cable modem is currently using (DOCSIS 1.0 or 1.1).
CPEipAddress	IP address for each CPE device that is using this cable modem, expressed in dotted decimal notation. This object is optional and can be suppressed to improve performance by reducing the size of the billing record files.
CMTSipAddress	IP address for the CMTS, expressed in dotted decimal notation.

Object Name	Description
CMTShostName	Fully qualified hostname for the CMTS (for example, cmts01.cisco.com).
CMTSsysUpTime	Amount of time, in hundredths of a second, since the last initialization of the CMTS management interface, expressed as a 32-bit decimal number (0 to 4,294,967,296).
RecType (SFType renamed to RecType in Cisco IOS Release 12.3(17a)BC)	Type of service flow being described: • **Interim**—the service flow was active throughout the collection period and should be reported as 1. • **Stop**—the service flow was deleted at some point during the collection period and should be reported as 2.
serviceIdentifier	Service flow ID assigned to this service flow by the CMTS, expressed as a decimal number. **Note** For DOCSIS 1.0 cable modems, the SFID field always shows the primary service flow for the upstream or downstream.
serviceDirection	Direction for the service flow (**Downstream** or **Upstream**).
serviceOctetsPassed	Total number of octets received by the cable modem (downstream service flows) or transmitted by the cable modem (upstream service flows) during the collection period, expressed as a 64-bit decimal number.
servicePktsPassed	Total number of packets received by the cable modem (downstream service flows) or transmitted by the cable modem (upstream service flows) during the collection period, expressed as a 64-bit decimal number.
SLAdropPkts	(Downstream service flows only) Total number of downstream packets for the cable modem that the CMTS dropped because otherwise they would have exceeded the bandwidth levels allowed by the subscriber's service level agreement (SLA), expressed as a 64-bit decimal number.
SLAdelayPkts	(Downstream service flows only) Total number of packets that the CMTS delayed transmitting on the downstream to the cable modem because otherwise they would have exceeded bandwidth levels allowed by the subscriber's service level agreement (SLA), expressed as a 64-bit decimal number.
CMTScatvIfIndex	The ifIndex of the MAC interface.
CMTScatvIfName	The ifName of the CMTS CATV (MAC) interface associated with this cable modem.
CMTSupIfName	The ifName of the CMTS Upstream interface associated with this cable modem.
CMTSdownIfName	The ifName of the CMTS Downstream interface associated with this cable modem.
CMcpeFqdn	FQDNs for cable modem associated CPEs.

Object Name	Description
serviceTimeCreated	Timestamp for SF creation (consistent with QoS MIB model).
serviceTimeActive	The active time of the SF in seconds.

Note Because the byte and packet counters are 64-bit values, it is possible for them to wrap around to zero during a billing period. The billing application should use the sysUpTime value along with the counters to determine whether the counters have wrapped since the last billing period. If a counter appears to regress, and if the current sysUpTime indicates this billing cycle is the next scheduled cycle for this particular cable modem, you can assume that the counter has wrapped during the billing cycle.

Note These billing record objects are defined in Appendix B, *IPDR Standards Submission for Cable Data Systems Subscriber Usage Billing Records* , in the *DOCSIS 2.0 OSSI Specification* (SP-OSSIv2.0-IO3-021218).

The following example shows a sample IPDR billing record for a downstream service flow:

```
<?xml version="1.0" encoding="UTF-8"?>
<IPDRDoc xmlns="http://www.ipdr.org/namespaces/ipdr"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="DOCSIS-3.1-B.0.xsd"
docId="C341A679-0000-0000-0000-000BBF54D000"
creationTime="2002-05-25T14:41:29Z"
IPDRRecorderInfo="CMTS01"
version="3.1">
</IPDR>
<IPDR xsi:type="DOCSIS-Type">
<IPDRcreationTime>2003-09-18T16:52:34Z</IPDRcreationTime>
<CMTShostname>R7519-UBR7246.cisco.com</CMTShostname>
<CMTSipAddress>1.8.8.21</CMTSipAddress>
<CMTSsysUpTime>287315    </CMTSsysUpTime>
<CMTScatvIfName>Cable8/0/0</CMTScatvIfName>
<CMTScatvIfIndex>13</CMTScatvIfIndex>
<CMTSupIfName>Ca8/0/0-upstream0</CMTSupIfName>
<CMTSupIfType>129</CMTSupIfType>
<CMTSdownIfName>Ca8/0/0-downstream</CMTSdownIfName>
<CMmacAddress>00-00-39-AB-D4-53</CMmacAddress>
<CMdocsisMode>1.0</CMdocsisMode>
<CMipAddress>3.8.21.3</CMipAddress>
<CPEipAddress></CPEipAddress>
<RecType>1</SFtype>
<serviceIdentifier>3</serviceIdentifier>
<serviceClassName></serviceClassName>
<serviceDirection>2</serviceDirection>
<serviceOctetsPassed>23457</ServiceOctetsPassed>
<servicePktsPassed>223</ServicePktsPassed>
<serviceSlaDropPkts>2</serviceSlaDropPkts>
<serviceSlaDelayPkts>0</serviceSlaDelayPkts>
<serviceTimeCreated>11000</serviceTimeCreated>
<serviceTimeActive>15890</serviceTimeActive>
</IPDR>
</IPDRDoc>
```

The following example shows a sample IPDR billing record for an upstream service flow:

<?xml version="1.0" encoding="UTF-8"?>

```
<IPDRDoc xmlns="http://www.ipdr.org/namespaces/ipdr"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="DOCSIS-3.1-B.0.xsd"
docId="docId="C3146152-0000-0000-0000-000BBF7D5800"
creationTime="2003-09-18T16:52:34Z"
IPDRRecorderInfo="CMTS01-UBR7246.cisco.com"
version="3.1">
<IPDR xsi:type=" DOCSIS-Type">
<IPDRcreationTime>2003-09-18T16:52:34Z</IPDRcreationTime>
<CMTShostname>R7519-UBR7246.cisco.com</CMTShostname>
<CMTSipAddress>1.8.8.21</CMTSipAddress>
<CMTSsysUpTime>287315    </CMTSsysUpTime>
<CMTScatvIfName>Cable8/0/0</CMTScatvIfName>
<CMTScatvIfIndex>13</CMTScatvIfIndex>
<CMTSupIfName>Ca8/0/0-upstream0</CMTSupIfName>
<CMTSupIfType>129</CMTSupIfType>
<CMTSdownIfName>Ca8/0/0-downstream</CMTSdownIfName>
<CMmacAddress>00-00-39-18-8A-4D</CMmacAddress>
<CMdocsisMode>1.0</CMdocsisMode>
<CMipAddress>3.8.21.14</CMipAddress>
<CPEipAddress></CPEipAddress>
<RecType>1</SFtype>
<serviceIdentifier>3</serviceIdentifier>
<serviceClassName></serviceClassName>
<serviceDirection>1</serviceDirection>
<serviceOctetsPassed>1404</ServiceOctetsPassed>
<servicePktsPassed>6</ServicePktsPassed>
<serviceSlaDropPkts>0</serviceSlaDropPkts>
<serviceSlaDelayPkts>0</serviceSlaDelayPkts>
<serviceTimeCreated>11000</serviceTimeCreated>
<serviceTimeActive>15890</serviceTimeActive>
</IPDR>
</IPDRDoc>
```

SNMP Support

Cisco cBR Series Converged Broadband Router s support the following MIBs that provide SNMPv2 support for the Usage-based Billing feature:

CISCO-CABLE-METERING-MIB

- Supports configuration of the usage-based billing feature using SNMPv2 commands.
- Displays the current usage-based billing configuration using SNMPv2 commands.
- Sends SNMPv2 traps based on the following usage-based billing events:
  - The Cisco CMTS reports that a new billing record is available.
  - The Cisco CMTS reports that a failure occurred in writing the most recent billing record (for example, the disk is full).
  - The Cisco CMTS reports that it could not successfully open a secure SSL connection to stream a billing record to the billing server.

CISCO-CABLE-WIDEBAND-MIB

Sets the polling interval for calculating the utilization of an RF channel by using the **ccwbRFChanUtilInterval** object.

DOCS-QOS-MIB

- Sets the load and utilization of both upstream and downstream physical channels through the **docsIfCmtsChannelUtilizationInterval** object. This information may be used for capacity planning and incident analysis, and may be particularly helpful in provisioning high value QoS.
- Displays information about all service flows (DOCSIS 1.1 service flows only) including multicast service flow is maintained in the **docsQosServiceFlowLogTable** in DOCS-QOS-MIB, **docsIetfQosServiceFlowLogTable** in DOCS-IETF-QOS-MIB, and **docsQos3ServiceFlowLogTable** in DOCS-QOS3-MIB.

To view information about deleted service flows, enable logging of deleted service flows using the **cable sflog** global configuration command.

Benefits

The usage-based billing feature provides the following benefits to cable service providers and their partners and customers:

- Allows service providers to integrate their billing applications for DOCSIS services with their other XML-capable billing applications.
- Standards-based approach that supports existing networks and services, such as DOCSIS and PacketCable, and is easily extensible to support future services as they are supported on the Cisco CMTS.

How to Configure the Usage-based Billing Feature

This section describes the following tasks that are required to implement the Usage-based Billing feature:

Enabling Usage-based Billing Feature File Mode Using CLI Commands

This section describes how to enable and configure the Usage-based Billing feature so that it operates in file mode, where it writes the billing record files to a local file system. The billing application must then log into the Cisco CMTS and retrieve the billing record files on a regular basis.

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	Example:
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	Example:
	`Router(config)#`
		Enters global configuration mode.

Step 3	**cable metering filesystem** *filesys* [**flow-aggregate**] [**cpe-list-suppress**] [**full-records**]
	Example:
	`Router(config)# cable metering filesystem harddisk:`
	Example:
	`Router(config)#`
		Enables the Usage-based Billing feature for file mode and configures it.
		The system will write the billing records on this file system using a file name that contains the hostname of the router followed by a timestamp when the record was written.

Step 4	**snmp-server enable traps cable metering**
	Example:
	`Router(config)# snmp-server enable traps cable metering`
	Example:
	`Router(config)#`
		(Optional) Enables SNMP traps for usage-based billing events. Traps are sent when a new billing record is available, or when the system encountered a failure (such as insufficient disk space) in writing the new billing record.

Step 5	**cable sflog max-entry** *number* **entry-duration** *time*
	Example:
	`Router(config)# cable sflog max-entry 2000 entry-duration 7200`
	Example:
	`Router(config)#`
		(Optional) Enables logging for deleted SNMP service flows, which allows the billing feature to include information about deleted service flows.

Step 6	**cable metering source-interface** *interface*
	Example:
	`Router(config)# cable metering source-interface loopback100`
	Example:
	`Router(config)#`
		(Optional) Enables specification of the source-interface for the billing packets, usually a loopback interface.

	Command or Action	Purpose

Step 7	**end**
	Example:
	`Router(config)# end`
	Example:
	`Router#`
		Exits global configuration mode and returns to privileged EXEC mode.

Enabling Usage-based Billing Feature File Mode Using SNMP Commands

This section describes how to enable and configure the Usage-based Billing feature so that it operates in file mode and writes the billing record files to a local file system. The billing application must then log into the Cisco CMTS and retrieve the billing record files on a regular basis.

To configure the Cisco CMTS for Usage-based Billing feature in file mode, you must set a number of objects in the CISCO-CABLE-METERING-MIB.

In addition, to include information about deleted service flows in the billing records (supported for DOCSIS 1.1 service flows), you must enable the logging of deleted service flows, using the **cable sflog** global configuration command.

Table 228: SNMP Objects to be Configured for File Mode

Object	Type	Description
ccmtrCollectionType	Integer	Enables or disables the Usage-based Billing feature. The valid values are: • 1—none. The Usage-based Billing feature is disabled (default). • 2—local. The Usage-based Billing feature is enabled and configured for file mode. • 3—stream. The Usage-based Billing feature is enabled and configured for streaming mode. Set ccmtrCollectionType to 2 (local) to enable the feature for file mode.
ccmtrCollectionFilesystem	DisplayString	Specifies the file system where the billing record file should be written. This object has a maximum length of 25 characters and must specify a valid file system on the router (such as slot0, disk1, or flash). **Note** The Cisco CMTS writes the billing records to this file system using a file name that consists of the router's hostname followed by a timestamp when the record was written.
ccmtrCollectionCpeList	TruthValue	(Optional) Indicates whether IP addresses for customer premises equipment (CPE) devices are omitted from the billing records, so as to reduce the size of the billing records and to improve performance. The valid values are the following: • true—CPE information is present (default). • false—CPE information is omitted. **Note** When set to true, a maximum of 5 CPE IP addresses for each cable modem.

Object	Type	Description
ccmtrCollectionAggregate	TruthValue	(Optional) Indicates whether all information for an individual cable modem is combined into one record. Separate counters are maintained for upstream and downstream traffic, but those counters include all service flows in that direction. The valid values are as follows: • true—All service flow information for each cable modem is aggregated into a single billing record. In this configuration, the service flow ID (SFID) for the billing record is set to 0 and the service class name (SCN) is blank. • false—Information for each cable modem is not aggregated into a single billing record, but instead each service flow is recorded into its own record (default).
ccmtrCollectionSrcIfIndex	TruthValue	(Optional) Specifies the source-interface for the billing packets.

Note The following steps use the standard SNMP commands that are available on many Unix and Linux systems. For each step, replace *ip-address* with the IP address of the Cisco CMTS, and replace *rw-community-string* with an SNMP community string that provides read-write access to the router.

Procedure

Step 1 Set the ccmtrCollectionType object to 2, to enable the Usage-based Billing feature and to configure it for file mode:

Example:

```
workstation# setany -v2c

ip-address rw-community-string

ccmtrCollectionType.0 -i 2

workstation#
```

Step 2 Set the ccmtrCollectionFilesystem object to the local file system where the Cisco CMTS should write the billing records:

Example:

```
workstation# setany -v2c

ip-address rw-community-string
 ccmtrCollectionFilesystem.0 -D disk0:

workstation#
```

Step 3 (Optional) To omit the IP addresses of CPE devices from the billing records, set the ccmtrCollectionCpeList object to 2 (false). The default is to include the CPE information.

Example:

```
workstation# setany -v2c

ip-address rw-community-string
 ccmtrCollectionCpeList.0 -i 2
workstation#
```

Step 4 (Optional) To aggregate all service flow information for each cable modem in a single record, set the ccmtrCollectionAggregate object to 1 (true). The default is for each service flow to be written in a separate record:

Example:

```
workstation# setany -v2c

ip-address rw-community-string
 ccmtrCollectionAggregate.0 -i 1
workstation#
```

Step 5 (Optional) To specify the source-interface for the billing packets, set the ccmtrtrCollectionSrcIfIndex object to 1 (true). The default is for the billing packets to automatically select a source-interface.

Example:

```
workstation# setany -v2c

ip-address rw-community-string
 ccmtrtrCollectionSrcIfIndex.0 -i 1
workstation#
```

Examples for Enabling Usage Billing using SNMP Mode

The following example shows the Usage-based Billing feature being configured using SNMP commands. The following display shows that a Cisco CMTS router at IP address 10.8.8.21 is configured with the default configuration (the Usage-based Billing feature is disabled):

```
workstation# getmany -v2c 10.8.8.21 rw-string ciscoCableMeteringMIB

ccmtrCollectionType.0 = none(1)
ccmtrCollectionFilesystem.0 =
ccmtrCollectionCpeList.0 = true(1)
ccmtrCollectionAggregate.0 = false(2)
ccmtrCollectionStatus.0 = 0
ccmtrCollectionDestination.0 =
ccmtrCollectionTimestamp.0 = 00 00  00 00   00 00  00 00
ccmtrCollectionNotifEnable.0 = true(1)
workstation#
```

The following SNMP commands are then given to enable the Usage-based Billing feature and to configure it for file mode:

```
workstation# setany -v2c 10.8.8.21 rw-string ccmtrCollectionType.0 -i 2
workstation# setany -v2c 10.8.8.21 rw-string
ccmtrCollectionFilesystem
.0 -D disk1:
```

```
workstation#
```

These commands add the following line to the router's running configuration file:

```
Router# show running-config | include metering

cable metering filesystem disk1:
Router#
```

The following SNMP display shows the new configuration, after the Cisco CMTS has successfully written a billing record:

```
workstation# getmany -v2c 10.8.8.21 rw-string ciscoCableMeteringMIB

ccmtrCollectionType.0 = local(2)
ccmtrCollectionFilesystem.0 = disk1:
ccmtrCollectionCpeList.0 = true(1)
ccmtrCollectionAggregate.0 = false(2)
ccmtrCollectionStatus.0 = success(1)
ccmtrCollectionDestination.0 = disk1:UBR7246.cisco.com-20030925-185827
ccmtrCollectionTimestamp.0 = 07 d3  09 19   12 3a  1c 00
ccmtrCollectionNotifEnable.0 = true(1)
workstation#
```

Enabling Usage-based Billing Feature Streaming Mode Using CLI Commands

This section describes how to enable and configure the Usage-based Billing feature so that it operates in streaming mode, where it regularly transmits the billing records to an external server for use by the billing application.

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	Example:
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	Example:
	`Router(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 3	**cable metering destination** *ip-address port* [*ip-address2 port2*] *retries minutes* {**non-secure** | **secure**} [**flow-aggregate**] [**cpe-list-suppress**] [**full-records**]
	Example:
	`Router(config)# cable metering destination 10.10.21.3 5300 10.10.21.4 5300 2 30 secure`
	Example:
	`Router(config)#`
		Enables the Usage-based Billing feature for streaming mode and configures it with the following parameters:

Step 4	**snmp-server enable traps cable metering**
	Example:
	`Router(config)# snmp-server enable traps cable metering`
	Example:
	`Router(config)#`
		(Optional) Enables SNMP traps for usage-based billing events. Traps are sent when a new billing record is available, or when the system encountered a failure (such as insufficient disk space) in writing the new billing record.

Step 5	**cable sflog max-entry** *number* **entry-duration** *time*
	Example:
	`Router(config)# cable sflog max-entry 2000 entry-duration 7200`
	Example:
	`Router(config)#`
		(Optional) Enables logging for deleted SNMP service flows, which allows the billing feature to include information about deleted service flows.

Step 6	**cable metering source-interface** *interface*
	Example:
	`Router(config)# cable metering source-interface loopback100`
	Example:
	`Router(config)#`
		(Optional) Enables specification of the source-interface for the billing packets, usually a loopback interface.

Step 7	**end** **Example:** `Router(config)# end` **Example:**	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router#`	

Enabling Usage-based Billing Feature Streaming Mode Using SNMP Commands

This section describes how to use SNMP commands to enable and configure the Usage-based Billing feature so that it operates in streaming mode, where it regularly transmits the billing records to an external server for use by the billing application.

To configure the Cisco CMTS for Usage-based Billing feature in streaming mode, you must set a number of objects in the CISCO-CABLE-METERING-MIB.

Note In addition, to include information about deleted service flows (DOCSIS 1.1 service flows only) in the billing records, you must enable the logging of deleted service flows, using the **cable sflog** global configuration command. See the *Cisco IOS CMTS Cable Command Reference Guide* on Cisco.com:

Cisco CMTS Cable Command Reference

Table 229: SNMP Objects to be Configured for Streaming Mode

Object	Type	Description
ccmCollectionType	Integer	Enables or disables the Usage-based Billing feature. The valid values are: • 1—none. The Usage-based Billing feature is disabled (default). • 2—local. The Usage-based Billing feature is enabled and configured for file mode. • 3—stream. The Usage-based Billing feature is enabled and configured for streaming mode. Set ccmCollectionType to 3 (stream) to enable the feature for streaming mode.
ccmCollectionIpAddress	InetAddress	IP address for the external collection server. This value must be specified.
ccmCollectionPort	Unsigned32	TCP port number at the external collection server to which the billing records should be sent. The valid range is 0 to 65535, but you should not specify a port in the well-known range of 0 to 1024. This value must be specified.
Note You can configure the ccmCollectionIpAddress and ccmCollectionPort objects twice, to specify a primary collection server and a secondary collection server.		
ccmCollectionIpAddrType	InetAddressType	(Optional) Type of IP address being used for the collection server. The only valid value is ipv4, which is the default value.
ccmCollectionInterval	Unsigned32	(Optional) Specifies how often, in minutes, the billing records are streamed to the external server. The valid range is 2 to 1440 minutes (24 hours), with a default of 30 minutes. (We recommend a minimum interval of 30 minutes.)

Object	Type	Description
ccmCollectionRetries	Unsigned32	(Optional) Specifies the number of retry attempts that the CMTS will make to establish a secure connection with the external server before using the secondary server (if configured) and sending an SNMP trap about the failure. The valid range for *n* is 0 to 5, with a default of 0.
Note The ccmCollectionInterval and ccmCollectionRetries parameters are optional when configuring usage-based billing for streaming mode with SNMP commands, but these parameters are required when configuring the feature with CLI commands.		
ccmCollectionSecure	TruthValue	(Optional) Specifies whether the Cisco CMTS should use a secure socket layer (SSL) connection when connecting with the billing application on the external server. The valid values are: • true(1)—The Cisco CMTS uses a SSL connection. This option is available only on CMTS software images that support Baseline Privacy Interface (BPI) encryption. • false(2)—The Cisco CMTS uses an unencrypted TCP connection. This is the default value.
ccmCollectionCpeList	TruthValue	(Optional) Indicates whether IP addresses for customer premises equipment (CPE) devices are omitted from the billing records, so as to reduce the size of the billing records and to improve performance. The valid values are the following: • true—CPE information is present (default). • false—CPE information is omitted. **Note** When set to true, a maximum of 5 CPE IP addresses for each cable modem.
ccmCollectionAggregate	TruthValue	(Optional) Indicates whether all information for an individual cable modem is combined into one record. Separate counters are maintained for upstream and downstream traffic, but those counters include all service flows in that direction. The valid values are as follows: • true—All service flow information for each cable modem is aggregated into a single billing record. In this configuration, the service flow ID (SFID) for the billing record is set to 0 and the service class name (SCN) is blank. • false—Information for each cable modem is not aggregated into a single billing record, but instead each service flow is recorded into its own record (default).
ccmtrCollectionSrcIfIndex	TruthValue	(Optional) Specifies the source-interface for the billing packets.

Note The following steps use the standard SNMP commands that are available on many Unix and Linux systems. For each step, replace *ip-address* with the IP address of the Cisco CMTS, and replace *rw-community-string* with an SNMP community string that provides read-write access to the router.

Procedure

Step 1 Set the ccmCollectionType object to 3, to enable the Usage-based Billing feature and to configure it for streaming mode:

Example:

```
workstation# setany -v2c
ip-address rw-community-string
 ccmCollectionType.0 -i 3
workstation#
```

Step 2 Set the ccmCollectionIpAddress and ccmCollectionPort objects to the IP address of the external collection server and the TCP port number to which billing records should be sent:

Example:

```
workstation# setany -v2c
ip-address rw-community-string
 ccmCollectionIpAddress.1 -o '0a 08 06 0b'

workstation# setany -v2c
ip-address rw-community-string
 ccmCollectionPort.1 -g 6789

workstation#
```

Step 3 (Optional) Set the ccmCollectionIpAddress and ccmCollectionPort objects a second time to specify the IP address and TCP port number of a second external collection server to which billing records should be sent, in the case that the Cisco CMTS cannot connect to the primary collection server:

Example:

```
workstation# setany -v2c
ip-address rw-community-string
 ccmCollectionIpAddress.1 -o '0a 08 06 0c'

workstation# setany -v2c
ip-address rw-community-string
 ccmCollectionPort.1 -g 7000

workstation#
```

Step 4 (Optional) To change any of the other default parameters, set the appropriate objects to the desired values. For example, the following lines configure the Usage-based Billing feature for a non-secure connection, with a collection interval of 45 minutes, and a maximum number of 3 retries.

Example:

```
workstation# setany -v2c
ip-address rw-community-string
 ccmCollectionSecure.1 -i 2
workstation# setany -v2c
ip-address rw-community-string
 ccmCollectionInterval.1 -i 45
workstation# setany -v2c
ip-address rw-community-string
```

```
 ccmCollectionRetries.1 -i 3
workstation#
```

Step 5 (Optional) To omit the IP addresses of CPE devices from the billing records, set the ccmCollectionCpeList object to 2 (false). The default is to include the CPE information.

Example:

```
workstation# setany -v2c

ip-address rw-community-string
 ccmCollectionCpeList.0 -i 2
workstation#
```

Step 6 (Optional) To aggregate all service flow information for each cable modem in a single record, set the ccmCollectionAggregate object to 1 (true). The default is for each service flow to be written in a separate record:

Example:

```
workstation# setany -v2c

ip-address rw-community-string
 ccmCollectionAggregate.0 -i 1
workstation#
```

Step 7 (Optional) To specify the source-interface for the billing packets, set the ccmtrCollectionSrcIfIndex object to 1 (true). The default is for the billing packets to automatically select a source-interface.

Example:

```
workstation# setany -v2c

ip-address rw-community-string
 ccmtrCollectionSrcIfIndex.0 -i 1
workstation#
```

Examples for SNMP Commands

The following example shows the Usage-based Billing feature being configured using SNMP commands. The following display shows that a Cisco CMTS router at IP address 10.8.8.21 is configured with the default configuration (the Usage-based Billing feature is disabled):

```
workstation# getmany -v2c 10.8.8.21 rw-string ciscoCableMeteringMIB

ccmCollectionType.0 = none(1)
ccmCollectionFilesystem.0 =
ccmCollectionCpeList.0 = true(1)
ccmCollectionAggregate.0 = false(2)
ccmCollectionStatus.0 = 0
ccmCollectionDestination.0 =
ccmCollectionTimestamp.0 = 00 00  00 00   00 00  00 00
ccmCollectionNotifEnable.0 = true(1)
workstation#
```

The following SNMP commands are then given to enable the Usage-based Billing feature and to configure it for streaming mode:

```
workstation# setany -v2c 10.8.8.21 rw-string ccmCollectionType.0 -i 3
workstation# setany -v2c 10.8.8.21 rw-string ccmCollectionIpAddress.1 -o '0a 08 06 0b'

workstation# setany -v2c 10.8.8.21 rw-string ccmCollectionPort.1 -g 6789

workstation# setany -v2c 10.8.8.21 rw-string ccmCollectionSecure.1 -i 2
workstation# setany -v2c 10.8.8.21 rw-string ccmCollectionRetries.1 -i 3
workstation# setany -v2c 10.8.8.21 rw-string ccmCollectionInterval.1 -i 45
workstation#
```

These commands add the following line to the router's running configuration file:

```
Router# show running-config | include metering

cable metering destination 10.8.6.11 6789 3 45 non-secure
Router#
```

The following SNMP display shows the new configuration:

```
workstation# getmany -v2c 10.8.8.21 rw-string ciscoCableMeteringMIB

ccmCollectionType.0 = stream(3)
ccmCollectionFilesystem.0 =
ccmCollectionIpAddrType.1 = ipv4(1)
ccmCollectionIpAddress.1 = 0a 08  06 0b
ccmCollectionPort.1 = 6789
ccmCollectionInterval.1 = 45
ccmCollectionRetries.1 = 3
ccmCollectionSecure.1 = false(2)
ccmCollectionRowStatus.1 = active(1)
ccmCollectionCpeList.0 = true(1)
ccmCollectionAggregate.0 = false(2)
ccmCollectionStatus.0 = 0
ccmCollectionDestination.0 =
ccmCollectionTimestamp.0 = 00 00  00 00   00 00  00 00
ccmCollectionNotifEnable.0 = true(1)
workstation#
```

Enabling and Configuring the Secure Copy Protocol (optional)

This section describes how to configure the Cisco CMTS for the Secure Copy Protocol (SCP), which allow an external server to log in to the Cisco CMTS and copy the billing records from the Cisco CMTS to the external server.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose

	`Router> enable`
	Example:
	`Router#`	

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	Example:
	`Router(config)#`
		Enters global configuration mode.

Step 3	**aaa new-model**
	Example:
	`Router(config)# aaa new-model`
	Example:
	`Router(config)#`
		Enables the Authentication, Authorization, and Accounting (AAA) access control model.

Step 4	**aaa authentication login** {**default** | *list-name* } *method1* [*method2* ...]
	Example:
	`Router(config)# aaa authentication login default enable`
	Example:
	`Router(config)#`
		Enables AAA access control authentication at login, using the following parameters:
		Valid methods include **enable**, **line**, and **local**.

Step 5	**aaa authorization exec** {**default** | *list-name* } *method1* [*method2* ...]
	Example:
	`Router(config)# aaa authorization exec default local`
	Example:
	`Router(config)#`
		Configures the CMTS to allow users to run an EXEC shell and access the CLI to run the Secure Copy commands.
		Valid methods include **local**.

Step 6	**username** *name* **privilege** *level* **password** *encryption-type password* **Example:**	(Optional) Creates a user account for login access and specifies the privilege level and password for that account:

	Command or Action	Purpose

	`Router(config)# username billingapp privilege 15 password 7 billing-password`
	Example:
	`Router(config)#`
		Note This step is optional but for the purposes of security and management, Cisco recommends creating a unique account for the billing application to use when logging into the CMTS.

Step 7	**ip ssh time-out** *seconds*
	Example:
	`Router(config)# ip ssh time-out 120`
	Example:
	`Router(config)#`
		Enables Secure Shell (SSH) access on the Cisco CMTS, which is required for SCP use. The *seconds* parameter specifies the maximum time allowed for SSH authentication, in seconds, with a valid range of 0 to 120 seconds, with a default of 120 seconds.

Step 8	**ip ssh authentication-retries** *n*
	Example:
	`Router(config)# ip ssh authentication-retries 3`
	Example:
	`Router(config)#`
		Specifies the maximum number of login attempts a user is allowed before the router disconnects the SSH session. The valid range is 1 to 5, with a default of 3 attempts.

Step 9	**ip scp server enable**
	Example:
	`Router(config)# ip scp server enable`
	Example:
	`Router(config)#`
		Enables SCP access on the Cisco CMTS.

Step 10	**end**
	Example:
	`Router(config)# end`
	`Router#`
		Exits global configuration mode and returns to privileged EXEC mode.

Configuring the Cisco CMTS for SSL Operation

This section describes the procedures to configure the Cisco CMTS for secure socket layer (SSL) operation, so that the Usage-based Billing feature can use an SSL connection to transfer the billing record files in streaming mode.

Note This procedure is required only when using the **secure** option with the **cable metering destination** command.

Prerequisites for CA

- The billing application server must be configured for SSL operations.
- A Certificate Authority (CA) must be configured to provide the required digital certificates to the billing application and Cisco CMTS router. The CA can be a public CA, such as Verisign, or a server on your private management network that is running software such as the Cisco Provisioning Center (CPC).

SUMMARY STEPS

To prepare the Cisco CMTS router for SSL operation, you must perform the following configuration steps:

- Configuring the router's host name and IP domain name, if not already done.
- Generating an RSA key pair.
- Declaring a Certification Authority.
- Configuring a Root CA (Trusted Root).
- Authenticating the CA.
- Requesting the Certificates.

For the detailed steps in performing these procedures, see the Configuring Certification Authority Interoperability

Retrieving Records from a Cisco CMTS in File Mode

When the Usage-based Billing feature is enabled and configured for File mode, the billing application server must regularly retrieve the billing records from the Cisco CMTS. This is typically done by a script that either logs in to the Cisco CMTS and uses CLI commands to transfer the file, or by a script that uses SNMP commands to transfer the file.

When using CLI commands, the procedure is typically as follows:

1. The billing application server receives an SNMP trap from the Cisco CMTS when a billing record is written. This notification contains the file name of the billing record that should be retrieved.
2. The billing application server starts a custom-written script to retrieve the billing record. This script would do one of the following:
   a. If using CLI commands, the script logs in to the Cisco CMTS using a telnet connection, and then transfers the billing record to the billing application server, using the **copy** CLI command. The transfer can be done using either the Secure Copy Protocol (SCP) or the Trivial File Transfer Protocol (TFTP).

Note You could also use the File Transfer Protocol (FTP) to transfer files from the Cisco CMTS to an external FTP server, but this is not recommended, because the FTP protocol transmits the login username and password in cleartext.

1. If using SNMP commands, the script sets the ciscoFlashCopyEntry objects in the CISCO-FLASH-MIB to transfer the billing record to the application server, using TFTP.

2. After transferring the billing record, the script deletes it on the Cisco CMTS file system, so that the Cisco CMTS can begin writing a new billing record.

The following sections show examples of how this can be done, using each method.

Tip The following examples are given for illustration only. Typically, these commands would be incorporated in automated scripts that would retrieve the billing records.

Using SCP

To transfer billing records using SCP, you must first enable and configure the router for SCP operation, using the procedure given in the "Enabling and Configuring Secure Copy (optional)" section on page 21 . Then, the application server must log in to the Cisco CMTS and use the **copy** command at the privileged EXEC prompt. The **copy** command needs to specify the location of the billing record on the local filesystem and the destination server for the SCP transfer.

The following example shows a typical session where a billing record on slot0 is transferred to an FTP server with the hostname of billserver.mso-example.com:

```
CMTS01# copy slot0:CMTS01_20030211-155025 scp://billingapp-server.mso-example.com/

Address or name of remote host [billingapp-server.mso-example.com]?
Destination username [billing-app]?
Destination filename [CMTS01_20030211-155025]?
Writing CMTS01_20030211-155025
Password: billing-password

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 1403352/1024 bytes]
1403352 bytes copied in 17.204 secs (85631 bytes/sec)
CMTS01# delete slot0:CMTS01_20030211-155025

CMTS01# squeeze slot0:

CMTS01#
```

Note The billing application must delete the billing record after it has been successfully transferred, so that the Cisco CMTS can write the next record. The **squeeze** command frees up the deleted disk space on Flash Memory and old-style PCMCIA cards (bootflash, flash, slot0, slot1). It is not needed on the newer ATA-style PCMCIA cards (disk0, disk1, disk2). However, because the **squeeze** command takes several seconds to complete, it should be given only when insufficient disk space exists for a new billing record. To avoid this problem, Cisco recommends using a 64 MB (or larger) ATA-style PCMCIA memory card, which automatically reclaims disk space for deleted files.

Using TFTP

To transfer billing records using TFTP, you must first configure an external workstation to be a TFTP server. For security, the TFTP server should be isolated from the Internet or any external networks, so that only authorized TFTP clients, such as the Cisco CMTS router, can access the server.

To transfer the billing records, the application server must log in to the Cisco CMTS and use the **copy** command at the privileged EXEC prompt. The **copy** command needs to specify the location of the billing record on the local filesystem and the destination server for the TFTP transfer.

The following example shows a typical session where a billing record on slot0 is transferred to an TFTP server with the hostname of billserver.mso-example.com.

```
Router# copy slot0:CMTS01_20030211-155025 tftp://billingapp-server.mso-example.com/incoming

Address or name of remote host [billingapp-server.mso-example.com]?
Destination username [billing-app]?
Destination filename [CMTS01_20030211-155025]?
Writing CMTS01_20030211-155025
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 1102348/1024 bytes]
1102348 bytes copied in 14.716 secs (63631 bytes/sec)
Router# delete slot0:CMTS01_20030211-155025

Router# squeeze slot0:

Router#
```

Note The billing application must delete the billing record after it has been successfully transferred, so that the Cisco CMTS can write the next record. The **squeeze** command frees up the deleted disk space on Flash Memory and old-style PCMCIA cards (bootflash, flash, slot0, slot1). It is not needed on the newer ATA-style PCMCIA cards (disk0, disk1, disk2). However, because the **squeeze** command takes several seconds to complete, it should be given only when insufficient disk space exists for a new billing record. To avoid this problem, Cisco recommends using a 64 MB (or larger) ATA-style PCMCIA memory card, which automatically reclaims disk space for deleted files.

Using SNMP

To transfer billing record file using SNMP commands, you must set a number of objects in the CISCO-FLASH-MIB to transfer the file to a TFTP server. After the file has been successfully transferred, you can then use SNMp commands to delete the billing record file.

Note Before proceeding with these steps, ensure that the TFTP server is properly configured to receive to receive the billing records. At the very least, this means creating a directory that is readable and writable by all users. On some servers, the TFTP server software also requires that you create a file with the same name as the file that is to be received, and this file should also be readable and writable by all users.

To transfer a billing record file to a TFTP server, using SNMP commands, you must set a number of objects in the CISCO-FLASH-MIB.

Table 230: Transferring a File to a TFTP Server Using SNMP Commands

Object	Type	Description
ciscoFlashCopyEntryStatus	RowStatus	Status of this table entry. Typically, this object is first set to 5 (create-and-wait). Then after all other parameters are specified, it is set to Active (1) to execute the command.
ciscoFlashCopyCommand	INTEGER	Type of copy command to be performed. To copy a billing record file to a TFTP server, set this object to 3 (copyFromFlash).
ciscoFlashCopyServerAddress	IpAddress	IP address of the TFTP server. **Note** This parameter defaults to the broadcast address of 255.255.255.255, which means it will transfer the billing record file to the first TFTP server that responds. For security, this object should always be set to the IP address of the authorized TFTP server.
ciscoFlashCopySourceName	DisplayString	Name of the billing record file to be transferred, including the Flash device on which it is stored.
ciscoFlashCopyDestinationName	DisplayString	(Optional) Name for the billing record, including path, on the TFTP server. If not specified, the copy operation defaults to saving the billing record at the top-most directory on the TFTP server, using the original file name. **Note** A file with the destination file name should already exist on the TFTP server. This file should be readable and writable by all users, so that it can be replaced with the billing record file.
ciscoFlashCopyProtocol	INTEGER	(Optional) Specifies the protocol to be used when copying the file. For a TFTP transfer, set this object to 1 (tftp), which is the default.
ciscoFlashCopyNotifyOnCompletion	TruthValue	(Optional) Specifies whether the Cisco CMTS should generate a trap upon the completion of the copy operation. The default is false (no trap is generated).

After transferring the billing records file, you must then set a number of objects in the CISCO-FLASH-MIB to delete the file, so that the Cisco CMTS can begin writing a new file. If the Flash memory is not ATA-compatible, you must also set a number of objects to squeeze the Flash memory to make the deleted space available for new files. Table 231: Deleting a File Using SNMP Commands , on page 1395 describes each of these objects, and whether they are required or optional.

Table 231: Deleting a File Using SNMP Commands

Object	Type	Description
ciscoFlashMiscOpCommand	INTEGER	Specifies the operation to be performed: • 3—Delete the file. • 5—Squeeze the Flash memory, so as to recover the deleted space and make it available for new files.

Object	Type	Description
ciscoFlashMiscOpDestinationName	DisplayString	When deleting a file, the name of the file to be deleted, including the name of the file system, up to a maximum of 255 characters. When squeezing a file system, the name of the file system to be squeezed (slot0:, slot1:, flash:, or bootflash:).
ciscoFlashMiscOpEntryStatus	RowStatus	Status of this table entry. Typically, this object is first set to 5 (create-and-wait). Then after all other parameters are specified, it is set to Active (1) to execute the command.
ciscoFlashMiscOpNotifyOnCompletion	TruthValue	(Optional) Specifies whether the Cisco CMTS should generate a trap upon the completion of the operation. The default is false (no trap is generated).

DETAILED STEPS

Note The following steps use the standard SNMP commands that are available on many Unix and Linux systems. For each step, replace *ip-address* with the IP address of the Cisco CMTS, and replace *rw-community-string* with an SNMP community string that provides read-write access to the router.

Copying the Billing Record File to the TFTP Server

Procedure

Step 1 The script performing the copy should generate a 32-bit number to be used as the index entry for this copy command. The script can generate this number in any convenient way, so long as the index number is not currently being used for another operation.

Step 2 Create the table entry for the copy command, by using the number that was generated in Step 1 and setting the ciscoFlashCopyEntryStatus object to the create-and-wait state (5):

Example:

```
workstation# setany -v2c ip-address rw-community-string ciscoFlashCopyEntryStatus.582 -i 5

workstation#
```

Step 3 Set the ciscoFlashCopyCommand to 3 (copyFromFlash) to specify that the billing record file should be copied from the router's Flash file system:

Example:

```
workstation# setany -v2c ip-address rw-community-string ciscoFlashCopyCommand
.582

-i 3
workstatioN#
```

Step 4 Set the ciscoFlashCopyServerAddress object to the IP address of the TFTP server:

Example:

```
workstation# setany -v2c ip-address rw-community-string ciscoFlashCopyServerAddress
.582

-a "172.20.12.193"

workstatioN#
```

Step 5 Set the ciscoFlashCopySourceName object to the file name, including the device name, of the billing record file to be transferred:

Example:

```
workstation# setany -v2c ip-address rw-community-string ciscoFlashCopySourceName
.582
-D
"slot0:CMTS01_20030211-155025
"
workstation#
```

Step 6 (Optional) To specify a specific destination on the TFTP server, set the ciscoFlashCopyDestinationName object to the path name and file name for the billing record file on the TFTP server. (Typically, the path name and file name should already exist on the TFTP server.)

Example:

```
workstation# setany -v2c ip-address rw-community-string ciscoFlashCopyDestinationName
.582
-D
"/cmts01-billing/billing-file
"
workstation#
```

Step 7 To execute the command, set the ciscoFlashCopyEntryStatus object to the active state (1):

Example:

```
workstation# setany -v2c ip-address rw-community-string ciscoFlashCopyEntryStatus.582 -i 1

workstation#
```

Step 8 Periodically poll the ciscoFlashCopyStatus object until the file transfer completes:

Example:

```
workstation# getmany -v2c ip-address rw-community-string ciscoFlashCopyStatus
.582
  ciscoFlashCopyStatus.582 = copyInProgress(1)
workstation# getmany -v2c ip-address rw-community-string ciscoFlashCopyStatus
.582
  ciscoFlashCopyStatus.582 = copyInProgress(1)
workstation# getmany -v2c ip-address rw-community-string ciscoFlashCopyStatus
.582
  ciscoFlashCopyStatus.582 = copyOperationSuccess(2)
workstation#
```

If the file transfer fails, the most common status values that are reported by the ciscoFlashCopyStatus object are:

- 3—copyInvalidOperation. This indicates that the operation failed on the TFTP server, typically because the destination file name and path name do not exist on the TFTP server, or they exist but are not writable by all users.
- 5—copyInvalidSourceName. The file name for the billing record, as specified in ciscoFlashCopySourceName does not exist. Verify that you specified the correct device name and that no spaces exist in the file name.
- 6—copyInvalidDestName. The destination path name and file name specified in ciscoFlashCopyDestinationName is not accessible on the TFTP server. This could be because the path name does not exist or is not configured to allow write-access. This error could also occur if a file with the same path name and file name already exists on the TFTP server.
- 7—copyInvalidServerAddress. The IP address of the TFTP server specified in ciscoFlashCopyServerAddress is invalid, or the TFTP server is not responding.
- 14—copyFileTransferError. A network error occurred that prevented the file transfer from completing.

Step 9 After the file transfer has completed successfully, set the ciscoFlashCopyEntryStatus object to 6 (delete) to delete the row entry for this copy command:

Example:

```
workstation# setany -v2c ip-address rw-community-string ciscoFlashCopyEntryStatus.582 -i 6

workstation#
```

What to do next

Deleting the Billing Record File

Using SNMP

After the billing record file has been successfully transferred, use the following procedure to delete the billing record on the Cisco CMTS flash file system, so that the Cisco CMTS can write the new billing record.

Procedure

Step 1 Generate another random number to be used as an index entry and configure the following objects in the ciscoFlashMiscOpTable:

Example:

```
workstation# setany -v2c ip-address rw-community-string ciscoFlashMiscOpEntryStatus
.31 -i 5

workstation# setany -v2c ip-address rw-community-string ciscoFlashMiscOpCommand
.31 -i 3
workstation# setany -v2c ip-address rw-community-string ciscoFlashMiscOpDestinationName
.31 -D
"/cmts01-billing/CMTS01_20030211-155025
"
workstation# setany -v2c ip-address rw-community-string ciscoFlashMiscOpEntryStatus
.31 -i 1

workstation#
```

Step 2 Periodically poll the ciscoFlashMiscOpStatus object until the file transfer completes:

Example:

```
workstation# getmany -v2c ip-address rw-community-string ciscoFlashMiscOpStatus
.31
  ciscoFlashCopyStatus.31 = miscOpInProgress(1)
workstation# getmany -v2c ip-address rw-community-string ciscoFlashMiscOpStatus
.31
  ciscoFlashCopyStatus.582 = miscOpOperationSuccess(2)
workstation#
```

Step 3 If the Flash memory system is not ATA-compatible (slot0:, slot1:, flash:, or bootflash:), configure the following objects in the ciscoFlashMiscOpTable to squeeze the Flash file system to recover the deleted file space:

Example:

```
workstation# setany -v2c ip-address rw-community-string ciscoFlashMiscOpEntryStatus
.32
-i 5

workstation# setany -v2c ip-address rw-community-string ciscoFlashMiscOpCommand
.32 -i 5
workstation# setany -v2c ip-address rw-community-string ciscoFlashMiscOpDestinationName
.32 -D slot0:
workstation# setany -v2c ip-address rw-community-string ciscoFlashMiscOpEntryStatus
.32
-i 1

workstation#
```

Examples To Transfer Using SNMP

The following SNMP commands transfer a file named CMTS01_20030211-155025 to a TFTP server at the IP address 10.10.31.3. After the file is successfully transferred, the row entry for this copy command is deleted.

```
workstation# setany -v2c 10.8.8.21 rw-string
 ciscoFlashCopyEntryStatus.582 -i 5

workstation# setany -v2c 10.8.8.21 rw-string
 ciscoFlashCopyCommand
.582
 -i 3
workstation# setany -v2c 10.8.8.21 rw-string
 ciscoFlashCopyServerAddress
.582
 -a "10.10.31.3"

workstation# setany -v2c 10.8.8.21 rw-string
 ciscoFlashCopySourceName
.582 -D
"slot0:CMTS01_20030211-155025
"
workstation# setany -v2c 10.8.8.21 rw-string
 ciscoFlashCopyDestinationName
.582 -D
"/cmts01-billing/CMTS01_20030211-155025
"
```

```
workstation# setany -v2c 10.8.8.21 rw-string
 ciscoFlashCopyEntryStatus.582 -i 1

workstation# getmany -v2c 10.8.8.21 rw-string
 ciscoFlashCopyStatus
.582
  ciscoFlashCopyStatus.582 = copyInProgress(1)
workstation# getmany -v2c 10.8.8.21 rw-string
 ciscoFlashCopyStatus
.582
  ciscoFlashCopyStatus.582 = copyOperationSuccess(2)
workstation# setany -v2c 10.8.8.21 rw-string
 ciscoFlashCopyEntryStatus.582 -i 6

workstation#
```

The following commands show a billing record file being deleted on the Cisco CMTS file system, and the deleted file space being recovered by a squeeze operation:

```
workstation# setany -v2c 10.8.8.21 rw-string
 ciscoFlashMiscOpEntryStatus
.31 -i 5

workstation# setany -v2c 10.8.8.21 rw-string
 ciscoFlashMiscOpCommand
.31 -i 3
workstation# setany -v2c 10.8.8.21 rw-string
 ciscoFlashMiscOpDestinationName
.31 -D
"/cmts01-billing/CMTS01_20030211-155025
"
workstation# setany -v2c 10.8.8.21 rw-string
 ciscoFlashMiscOpEntryStatus
.31 -i 1

workstation# getmany -v2c 10.8.8.21 rw-string
 ciscoFlashMiscOpStatus
.31
  ciscoFlashCopyStatus.31 = miscOpInProgress(1)
workstation# getmany -v2c 10.8.8.21 rw-string
 ciscoFlashMiscOpStatus
.31
  ciscoFlashCopyStatus.582 = miscOpOperationSuccess(2)
workstation# setany -v2c 10.8.8.21 rw-string
 ciscoFlashMiscOpEntryStatus
.32 -i 5

workstation# setany -v2c 10.8.8.21 rw-string
 ciscoFlashMiscOpCommand
.32 -i 5
workstation# setany -v2c 10.8.8.21 rw-string
 ciscoFlashMiscOpDestinationName
.32 -D slot0:
workstation# setany -v2c 10.8.8.21 rw-string
 ciscoFlashMiscOpEntryStatus
.32 -i 1

workstation#
```

Disabling the Usage-based Billing Feature

This section describes how to disable the Usage-based Billing. Giving this command immediately stops the collection of billing information. If a billing record is currently written or being streamed to an external server, the CMTS completes the operation before disabling the usage-based billing feature.

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	Example:
	`Router#`
		Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	Example:
	`Router(config)#`
		Enters global configuration mode.

Step 3	**no cable metering**
	Example:
	`Router(config)# no cable metering`
	Example:
	`Router(config)#`
		Immediately disables the Usage-based Billing feature and stops the collection of billing information.

Step 4	**no snmp-server enable traps cable metering**
	Example:
	`Router(config)# no snmp-server enable traps cable metering`
	Example:
	`Router(config)#`
		(Optional) Disables SNMP traps for usage-based billing events.

Step 5	**no cable sflog** **Example:** `Router(config)# no cable sflog` **Example:**	(Optional) Disables the logging of deleted service flows.

	Command or Action	Purpose
	`Router(config)#`	

Step 6	**no cable metering source-interface**
	Example:
	`Router(config)# no cable metering source-interface`
	Example:
	`Router(config)#`
		(Optional) Disables a specified source-interface for the billing packets.

Step 7	**exit**
	Example:
	`Router(config)# exit`
	Example:
	`Router#`
		Exits global configuration mode.

Configuring Certified SSL Servers for Usage-Based Billing

Cisco introduces adds support for the Secure Socket Layer (SSL) Server, used with the usage-based billing feature of the Cisco CMTS. Usage-based billing implements the DOCSIS Subscriber Account Management Interface Specification (SAMIS) format.

This new capability enables the configuration of the SSL server between the Cisco CMTS and a collection server. Certificate creation steps and **debug** commands are added or enhanced to support the SSL Server and certificates. This section describes general steps.

Refer also to the Configuring the Cisco CMTS for SSL Operation, on page 1391 section.

Generating SSL Server Certification

These general steps describe the creation and implementation of certification for the Secure Socket Layer (SSL) Server.

1. Generate the CA key.
2. Set up the open SSL environment, to include directory and sub-directory.
3. Copy files to the appropriate directories.
4. Generate the SSL Server certification request.
5. Grant the SSL Server certification request.
6. Convert the SSL Server certification to DER format.
7. Copy the SSL certification to Bootflash memory (write mem).
8. Start the SSL server.

Configuring and Testing the Cisco CMTS for Certified SSL Server Support

Perform the following steps to configure the Cisco router to support the SSL Server and certification.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip domain name** *domain* **Example:** `Router(config)# ip domain name Cisco.com`	Defines a default domain name that the Cisco IOS software uses to complete unqualified host names (names without a dotted-decimal domain name). Do not include the initial period that separates an unqualified name from the domain name. **Note** See the Domain Name System (DNS) document on Cisco.com for additional DNS information.
Step 4	**crypto key generate rsa** **Example:** `Router(config)# crypto key generate rsa`	Generates RSA key pairs.

Step 5	Ctrl-Z
	Example:
	`Router(config)# Ctrl-Z`
	Example:
	`Router#`
		Returns to privileged EXEC mode.

Step 6	**test cable read certificate** **Example:** `Router# test cable read certificate`	Verifies the certificate is valid and operational on the Cisco CMTS.
Step 7	**show crypto ca certificate** **Example:**	Displays the available certificates on the Cisco CMTS.

	Command or Action	Purpose
	`Router# show crypto ca certificate`	

Step 8	**configure terminal**
	Example:
	`Router# configure terminal`
	Example:
	`Router(config)#`
		Enters global configuration mode.

Step 9	**cable metering destination** *ip-addr num-1 num-2 num-3* **secure** **Example:** `Router(config)# cable metering destination 1.7.7.7 6789 0 15 secure`	Defines the destination IP address for cable metering, to be used with the certificate.
Step 10	test cable metering **Example:** `Router# test cable metering`	Tests cable metering in light of the supported SSL server and metering configuration.

Monitoring the Usage-based Billing Feature

To display the most current billing record, use the **show cable metering-status** command. The following example shows typical output when usage-based billing is configured to write the billing records to a local file system:

```
CMTS01# show cable metering-status

destination                          complete-time   flow   cpe     status
                                                    aggr suppress
disk0:R7519-UBR7246-20000308-004428 Jun 12 09:33:05  No     No     success
CMTS01#
```

The following example shows a typical output for the **show cable metering-status** command when usage-based billing is configured to stream the billing records to an external server:

```
Router# show cable metering-status

destination                      complete-time   flow cpe  full status
                                                 aggr supp rec
10.11.37.2 :1234                 Jun 12 09:33:05  No   No  No success
Router#
```

The following example shows a typical output for the **show cable metering-status** command using verbose option:

```
Router# show cable metering-status verbose
Last export status
Destination : disk0:sunethra10k-20070129-190423
Complete Time : Jan29 19:04:38
Flow Aggregate : No
Full records : No
Cpe list suppression : No
Source interface : FastEthernet0/0/0
Status of last export : success
Current export status : In progress
```

The following example shows a typical output for the **show cable metering-status** command when usage-based billing is configured to use the IPDR Exporter to stream the billing records to an external server:

```
Router# show cable metering-status
destination                 complete-time   flow  cpe   full    status
               aggr  supp  rec
IPDR_Session2               Apr12 16:51:15  No    No    No    success
```

The following example shows a typical output for the verbose form of the **show cable metering-status** command when usage-based billing is configured to use the IPDR Exporter to stream the billing records to an external server:

```
Router# show cable metering-status
 verbose

Last export status
Destination                : IPDR_Session2
Complete Time          : Apr12 16:51:15
Flow Aggregate         : No
Full records               :No
Cpe list suppression   : No
Source interface       : Not defined
Status of last export  : success
```

Note If the **show cable metering-status** command displays the status of a streaming operation as "success" but the records were not received on the billing application server, verify that the Cisco CMTS and server are configured for the same type of communications (non-secure TCP or secure SSL). If the Cisco CMTS is configured for non-secure TCP and the server is configured for secure SSL, the Cisco CMTS transmits the billing record successfully, but the server discards all of the data, because it did not arrive in a secure SSL stream.

Tip The **show cable metering-status** command continues to show the status of the last billing record operation, until that billing record is deleted. If the record is not deleted, no new records are created.

To display information about the state of the IPDR Exporter, use the show ipdr Exporter command. The following example shows typical output:

```
Router#configure terminal
Router#show ipdr exporter
```

IPDR exporter is started.

Configuration Examples for Usage-based Billing

This section lists the following sample configurations for the Usage-based Billing feature:

File Mode Configuration (with Secure Copy)

The following excerpt from a configuration file shows a typical configuration for the Usage-based Billing feature when operating in file mode and enabling Secure Copy (SCP) for file transfers.

```
!
cable metering filesystem disk1:
snmp-server enable traps cable metering
...
aaa new-model
aaa authentication login default group tacacs+
aaa authorization exec default group tacacs+
username billingapp level 15 password 7 billing-password
ip ssh time-out 120
ip ssh authentication-retries 3
ip scp server enable
```

Non-Secure Streaming Mode Configuration

The following excerpt from a configuration file shows a typical configuration for the Usage-based Billing feature when operating in streaming mode and specifying both a primary and a secondary external server. The data is sent using standard TCP packets, without any security.

```
cable metering destination 10.10.10.171 5321 10.10.10.173 5321 2 30 non-secure
snmp-server enable traps cable metering
```

The following excerpt from a configuration file shows a typical configuration for the Usage-based Billing feature when operating in streaming mode and specifying only a primary external server:

```
cable metering destination 10.10.11.181 6789 2 30 non-secure
snmp-server enable traps cable metering
```

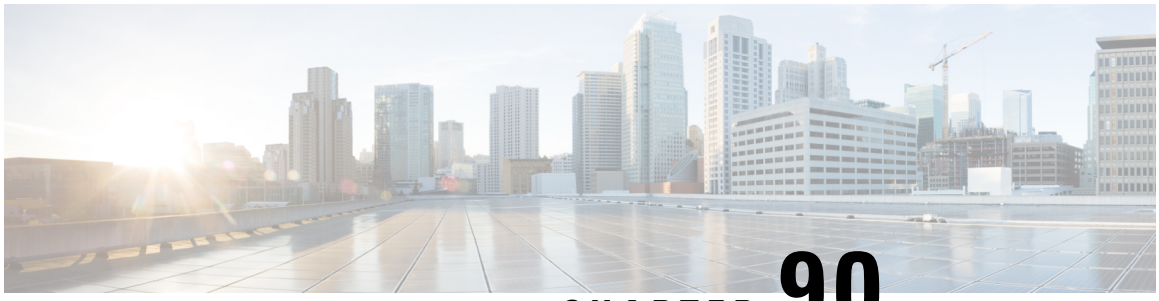
Note You must ensure that the billing application server is configured for standard TCP communications. If the billing application server is configured for SSL communications when the Cisco CMTS is configured for standard TCP, the Cisco CMTS is able to send the billing records to the server, but the server discards all of that information because it is not arriving in a secure stream.

Secure Streaming Mode Configuration

The following excerpt from a configuration file shows a typical configuration for the Usage-based Billing feature when operating in streaming mode and specifying only a primary external server. Secure socket layer (SSL) TCP connections are used to transmit the data, which requires the configuration of a digital certificate.

```
cable metering destination 10.10.11.181 6789 2 30 secure cpe-list-suppress
snmp-server enable traps cable metering
...
crypto ca trustpoint SSL-CERT
!
crypto ca certificate chain SSL-CERT
 certificate ca 00
  308204A6 3082038E A0030201 02020100 300D0609 2A864886 F70D0101 04050030
  8198310B 30090603 55040613 02555331 13301106 03550408 130A4361 6C69666F
  726E6961 3111300F 06035504 07130853 616E204A 6F736531 1C301A06 0355040A
  13134369 73636F20 53797374 656D732C 20496E63 2E311130 0F060355 040B1308
  4361626C 65204255 310E300C 06035504 03130553 65656D61 3120301E 06092A86
...
  3E65DBBA 337627E8 589980D6 C8836C7E 3D3C3BC1 F21973BF 7B287D7A 13B16DA2
  02B2B180 C2A125C7 368BDA4C 0B8C81B7 7D5BEFF9 A6618140 1E95D19E BD0A84F5
  B43702AB 39B5E632 87BA36AC A3A8A827 C5BAC0F1 B24B8F4D 55615C49 5B6E4B61
  B15CC48A 8EF566C8 6E449B49 BF8E9165 317C1734 9A48A240 78A356B5 403E9E9B
  88A51F5B 0FE38CC2 F431
  quit
!
```

Note You must ensure that the billing applications server is also configured for SSL communications.



CHAPTER 90

Frequency Allocation Information for the Cisco CMTS Routers

- Frequency Allocation for the Cisco CMTS Routers, on page 1409

Frequency Allocation for the Cisco CMTS Routers

The table below provides information about the NTSC 6-MHz channel bands:

Table 232: NTSC Cable Television Channels and Relative Frequencies in MHz

Channel Number	Bandwidth	Video Carrier	Color Carrier	Audio Carrier
T 7	5.75 - 11.75	7	10.58	11.5
T 8	11.75 - 17.75	13	16.58	17.5
T9	17.75-23.75	19	22.58	23.5
T10	23.75-29.75	25	28.58	29.5
T11	29.75-35.75	31	34.58	35.5
T12	35.75-41.75	37	40.58	41.5
T13	41.75-47.75	43	46.58	47.5
TV-IF	40.0-46.0	45.75	42.17	41.25
2-2	54.0-60.0	55.25	58.83	59.75
3-3	60.0-66.0	61.25	64.83	65.75
4-4	66.0-72.0	67.25	70.83	71.75
5-5	76.0-82.0	77.25	80.83	81.75
6-6	82.0-88.0	83.25	86.83	87.75

Channel Number	Bandwidth	Video Carrier	Color Carrier	Audio Carrier
FM	88.0-108.0			
A-5-95	90.0-96.0	91.25	94.83	95.75
A-4-96	96.0-102.0	97.25	100.83	101.75
A-3-97	102.0-108.0	103.25	106.83	107.75
A-2-98	108.0-114.0	109.25	112.83	113.75
A-1-99	114.0-120.0	115.25	118.83	119.75
A-14	120.0-126.0	121.25	124.83	125.75
B-15	126.0-132.0	127.25	130.83	131.75
C-16	132.0-138.0	133.25	136.83	137.75
D-17	138.0-144.0	139.25	142.83	143.75
E-18	144.0-150.0	145.25	148.83	149.75
F-19	150.0-156.0	151.25	154.83	155.75
G-20	156.0-162.0	157.25	160.83	161.75
H-21	162.0-168.0	163.25	166.83	167.75
I-22	168.0-174.0	169.25	172.83	173.75
7-7	174.0-180.0	175.25	178.83	179.75
8-8	180.0-186.0	181.25	184.83	185.75
9-9	186.0-192.0	187.25	190.83	191.75
10-10	192.0-198.0	193.25	196.83	197.75
11-11	198.0-204.0	199.25	202.83	203.75
12-12	204.0-210.0	205.25	208.83	209.75
13-13	210.0-216.0	211.25	214.83	215.75
J-23	216.0-222.0	217.25	220.83	221.75
K-24	222.0-228.0	223.25	226.83	227.75
L-25	228.0-234.0	229.25	232.83	233.75
M-26	234.0-240.0	235.25	238.83	239.75
N-27	240.0-246.0	241.25	244.83	245.75

Channel Number	Bandwidth	Video Carrier	Color Carrier	Audio Carrier
O-28	246.0-252.0	247.25	250.83	251.75
P-29	252.0-258.0	253.25	256.83	257.75
Q-30	258.0-264.0	259.25	262.83	263.75
R-31	264.0-270.0	265.25	268.83	269.75
S-32	270.0-276.0	271.25	274.83	275.75
T-33	276.0-282.0	277.25	280.83	281.75
U-34	282.0-288.0	283.25	286.83	287.75
V-35	288.0-294.0	289.25	292.83	293.75
W-36	294.0-300.0	295.25	298.83	299.75
AA-37	300.0-306.0	301.25	304.83	305.75
BB-38	306.0-312.0	307.25	310.83	311.75
CC-39	312.0-318.0	313.25	316.83	317.75
DD-40	318.0-324.0	319.25	322.83	323.75
EE-41	324.0-330.0	325.25	328.83	329.75
FF-42	330.0-336.0	331.25	334.83	335.75
GG-43	336.0-342.0	337.25	340.83	341.75
HH-44	342.0-348.0	343.25	346.83	347.75
II-45	348.0-354.0	349.25	352.83	353.75
JJ-46	354.0-360.0	355.25	358.83	359.75
KK-47	360.0-366.0	361.25	364.83	365.75
LL-48	366.0-372.0	367.25	370.83	371.75
MM-49	372.0-378.0	373.25	376.83	377.75
NN-50	378.0-384.0	379.25	382.83	383.75
OO-51	384.0-390.0	385.25	388.83	389.75
PP-52	390.0-396.0	391.25	394.83	395.75
QQ-53	396.0-402.0	397.25	400.83	401.75
RR-54	402.0-408.0	403.25	406.83	407.75

Channel Number	Bandwidth	Video Carrier	Color Carrier	Audio Carrier
SS-55	408.0-414.0	409.25	412.83	413.75
TT-56	414.0-420.0	415.25	418.83	419.75
UU-57	420.0-426.0	421.25	424.83	425.75
VV-58	426.0-432.0	427.25	430.83	431.75
WW-59	432.0-438.0	433.25	436.83	437.75
XX-60	438.0-444.0	439.25	442.83	443.75
YY-61	444.0-450.0	445.25	448.83	449.75
ZZ-62	450.0-456.0	451.25	454.83	455.75
AAA-63	456.0-462.0	457.25	460.83	461.75
BBB-64	462.0-468.0	463.25	466.83	467.75
CCC-65	468.0-474.0	469.25	472.83	473.75
DDD-66	474.0-480.0	475.25	478.83	479.75
EEE-67	480.0-486.0	481.25	484.83	485.75
FFF-68	486.0-492.0	487.25	490.83	491.75
GGG-69	492.0-498.0	493.25	496.83	497.75
HHH-70	498.0-504.0	499.25	502.83	503.75
III-71	504.0-510.0	505.25	508.83	509.75
JJJ-72	510.0-516.0	511.25	514.83	515.75
KKK-73	516.0-522.0	517.25	520.83	521.75
LLL-74	522.0-528.0	523.25	526.83	527.75
MMM-75	528.0-534.0	529.25	532.83	533.75
NNN-76	534.0-540.0	535.25	538.83	539.75
OOO-77	540.0-546.0	541.25	544.83	545.75
PPP-78	546.0-552.0	547.25	550.83	551.75
QQQ-79	552.0-558.0	553.25	556.83	557.75
RRR-80	558.0-564.0	559.25	562.83	563.75
SSS-81	564.0-570.0	565.25	568.83	569.75

Channel Number	Bandwidth	Video Carrier	Color Carrier	Audio Carrier
TTT-82	570.0-576.0	571.25	574.83	575.75
UUU-83	576.0-582.0	577.25	580.83	581.75
VVV-84	582.0-588.0	583.25	586.83	587.75
WWW-85	588.0-594.0	589.25	592.83	593.75
XXX86	594.0-600.0	595.25	598.83	599.75
YYY-87	600.0-606.0	601.25	604.83	605.75
ZZZ-88	606.0-612.0	607.25	610.83	611.75
89-89	612.0-618.0	613.25	616.83	617.75
90-90	618.0-624.0	619.25	622.83	623.75
91-91	624.0-630.0	625.25	628.83	629.75
92-92	630.0-636.0	631.25	634.83	635.75
93-93	636.0-642.0	637.25	640.83	641.75
94-94	642.0-648.0	643.25	646.83	647.75
100-100	648.0-654.0	649.25	652.83	653.75
101-101	654.0-660.0	655.25	658.83	659.75
102-102	660.0-666.0	661.25	664.83	665.75
103-103	666.0-672.0	667.25	670.83	671.75
104-104	672.0-678.0	673.25	676.83	677.75
105-105	678.0-684.0	679.25	682.83	683.75
106-106	684.0-690.0	685.25	688.83	689.75
107-107	690.0-696.0	691.25	694.83	695.75
108-108	696.0-702.0	697.25	700.83	701.75
109-109	702.0-708.0	703.25	706.83	707.75
110-110	708.0-714.0	709.25	712.83	713.75
111-111	714.0-720.0	715.25	718.83	719.75
112-112	720.0-726.0	721.25	724.83	725.75
113-113	726.0-732.0	727.25	730.83	731.75

Channel Number	Bandwidth	Video Carrier	Color Carrier	Audio Carrier
114-114	732.0-738.0	733.25	736.83	737.75
115-115	738.0-744.0	739.25	742.83	743.75
116-116	744.0-750.0	745.25	748.83	749.75
117-117	750.0-756.0	751.25	754.83	755.75
118-118	756.0-762.0	757.25	760.83	761.75
119-119	762.0-768.0	763.25	766.83	767.75
120-120	768.0-674.0	769.25	772.83	773.75
121-121	774.0-780.0	775.25	778.83	779.75
122-122	780.0-786.0	781.25	784.83	785.75
123-123	786.0-792.0	787.25	790.83	791.75
124-124	792.0-798.0	793.25	796.83	797.75
125-125	798.0-804.0	799.25	802.83	803.75
126-126	804.0-810.0	805.25	808.83	809.75
127-127	810.0-816.0	811.25	814.83	815.75
128-128	816.0-822.0	817.25	820.83	821.75
129-129	822.0-828.0	823.25	826.83	827.75
130-130	828.0-834.0	829.25	832.83	833.75
131-131	834.0-840.0	835.25	838.83	839.75
132-132	840.0-846.0	841.25	844.83	845.75
133-133	846.0-852.0	847.25	850.83	851.75
134-134	852.0-858.0	853.25	856.83	857.75
135-135	858.0-864.0	859.25	862.83	863.75
136-136	864.0-870.0	865.25	868.83	869.75
137-137	870.0-876.0	871.25	874.83	875.75
138-138	876.0-882.0	877.25	880.83	881.75
139-139	882.0-888.0	883.25	886.83	887.75
140-140	888.0-894.0	889.25	892.83	893.75

Channel Number	Bandwidth	Video Carrier	Color Carrier	Audio Carrier
141-141	894.0-900.0	895.25	898.83	899.75
142-142	900.0-906.0	901.25	904.83	905.75
143-143	906.0-912.0	907.25	910.83	911.75
144-144	912.0-918.0	913.25	916.83	917.75
145-145	918.0-924.0	919.25	922.83	923.75
146-146	924.0-930.0	925.25	928.83	929.75
147-147	930.0-936.0	931.25	934.83	935.75
148-148	936.0-942.0	937.25	940.83	941.75
149-149	942.0-948.0	943.25	946.83	947.75
150-150	948.0-954.0	949.25	952.83	953.75
151-151	954.0-960.0	955.25	958.83	959.75
152-152	960.0-966.0	961.25	964.83	965.75
153-153	966.0-972.0	967.25	970.83	971.75
154-154	972.0-978.0	973.25	976.83	977.75
155-155	978.0-984.0	979.25	982.83	983.75
156-156	984.0-990.0	985.25	988.83	989.75
157-157	990.0-996.0	991.25	994.83	995.75
158-158	996.0-1002.0	997.25	1000.83	1001.75

The table below provides information on the Phase Alternating Line (PAL) and Systeme Electronique Couleur Avec Memoire (SECAM) 8-MHz channel bands:

Table 233: European Cable Television Channels and Relative Frequencies in MHz

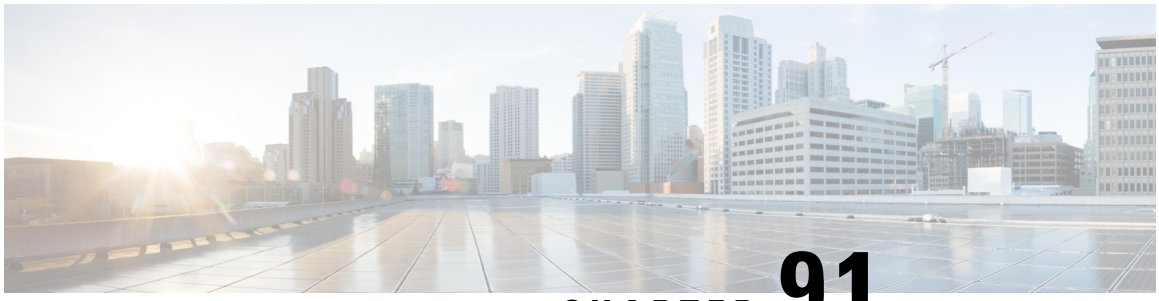
Channel Number	Bandwidth	Video Carrier	Audio Carrier
2	47-54	48.25	48.25
3	54-61	55.25	55.25
4	61-68	62.25	62.25
S2	111-118	112.25	112.25
S3	118-125	119.25	119.25

Channel Number	Bandwidth	Video Carrier	Audio Carrier
S4	125-132	126.25	126.25
S5	132-139	133.25	133.25
S6	139-146	140.25	140.25
S7	146-153	147.25	147.25
S8	153-160	154.25	154.25
S9	160-167	161.25	161.25
S10	167-174	168.25	168.25
5	174-181	175.25	175.25
6	181-188	182.25	182.25
7	188-195	189.25	189.25
8	195-202	196.25	196.25
9	202-209	203.25	203.25
10	209-216	210.25	210.25
11	216-223	217.25	217.25
12	223-230	224.25	224.25
S11	230-237	231.25	231.25
S12	237-244	238.25	238.25
S13	244-251	245.25	245.25
S14	251-258	252.25	252.25
S15	258-265	259.25	259.25
S16	265-272	266.25	266.25
S17	272-279	273.25	273.25
S18	279-286	280.25	280.25
S19	286-293	287.25	287.25
S20	293-300	294.25	294.25
S21	302-310	303.25	303.25
S22	310-318	311.25	311.25

Channel Number	Bandwidth	Video Carrier	Audio Carrier
S23	318-326	319.25	319.25
S24	326-334	327.25	327.25
S25	334-342	335.25	335.25
S26	342-350	343.25	343.25
S27	350-358	351.25	351.25
S28	358-366	359.25	359.25
S29	366-374	367.25	367.25
S30	374-382	375.25	375.25
S31	382-390	383.25	383.25
S32	390-398	391.25	391.25
S33	398-406	399.25	399.25
S34	406-414	407.25	407.25
S35	414-422	415.25	415.25
S36	422-430	423.25	423.25
S37	430-438	431.25	431.25
S38	438-446	439.25	439.25
21	470-478	471.25	471.25
22	478-486	479.25	479.25
23	486-494	487.25	487.25
24	494-502	495.25	495.25
25	502-510	503.25	503.25
26	510-518	511.25	511.25
27	518-526	519.25	519.25
28	526-534	527.25	527.25
29	534-542	535.25	535.25
30	542-550	543.25	543.25
31	550-558	551.25	551.25

Channel Number	Bandwidth	Video Carrier	Audio Carrier
32	558-566	559.25	559.25
33	566-574	567.25	567.25
34	574-582	575.25	575.25
35	582-590	583.25	583.25
36	590-598	591.25	591.25
37	598-606	599.25	599.25
38	606-614	607.25	607.25
39	614-622	615.25	615.25
40	622-630	623.25	623.25
41	630-638	631.25	631.25
42	638-646	639.25	639.25
43	646-654	647.25	647.25
44	654-662	655.25	655.25
45	662-670	663.25	663.25
46	670-678	671.25	671.25
47	678-686	679.25	679.25
48	686-694	687.25	687.25
49	694-702	695.25	695.25
50	702-710	703.25	703.25
51	710-718	711.25	711.25
52	718-726	719.25	719.25
53	726-734	727.25	727.25
54	734-742	735.25	735.25
55	742-750	743.25	743.25
56	750-758	751.25	751.25
57	758-766	759.25	759.25
58	766-774	767.25	767.25

Channel Number	Bandwidth	Video Carrier	Audio Carrier
59	774-782	775.25	775.25
60	782-790	783.25	783.25
61	790-798	791.25	791.25
62	798-806	799.25	799.25
63	806-814	807.25	807.25
64	814-822	815.25	815.25
65	822-830	823.25	823.25
66	830-838	831.25	831.25
67	838-846	839.25	839.25
68	846-854	847.25	847.25
69	854-862	855.25	855.25



CHAPTER 91

Flap List Troubleshooting

This document describes how to configure and use the Flap List Troubleshooting feature on the Cisco Cable Modem Termination System (CMTS) routers. The flap list is a patented tool for the Cisco CMTS routers to diagnose potential problems with a particular cable modem or with a particular cable interface. The flap list tracks "flapping" cable modems, which are cable modems that have intermittent connectivity problems. Excessive flapping could indicate a problem with a particular cable modem or with the upstream or downstream portion of the cable plant.

- Finding Feature Information, on page 1421
- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1422
- Prerequisites for Flap List Troubleshooting, on page 1422
- Restrictions for Flap List Troubleshooting, on page 1422
- Information About Flap List Troubleshooting, on page 1423
- How to Configure Flap List Troubleshooting, on page 1425
- How to Monitor and Troubleshoot Using Flap Lists, on page 1431
- Configuration Examples for Flap List Troubleshooting, on page 1437
- Additional References, on page 1437
- Feature Information for Flap List Troubleshooting, on page 1439

Finding Feature Information

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 234: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Prerequisites for Flap List Troubleshooting

- To configure and access the flap list using SNMP commands, you must be using an SNMPv3 manager and have configured the Cisco CMTS router for SNMP operations.

Restrictions for Flap List Troubleshooting

- The Flap List Troubleshooting feature can be used only with two-way cable modems. The flap-list does not support telco-return cable modems or set-top boxes.

Note Since the cable flap list was originally developed, polling mechanisms have been enhanced to have an increased rate of 1/sec when polls are missed. Cable modems can go offline faster than the frequency hop period, which can cause the frequency to stay fixed while cable modems go offline. To compensate for this, reduce the hop period to 10 seconds.

Information About Flap List Troubleshooting

This section describes the following information about the Flap List Troubleshooting feature:

Feature Overview

The Flap List Troubleshooting is a patented tool that is incorporated in the Cisco IOS software for the Cisco Cable Modem Termination System (CMTS) routers. The flap list tracks "flapping" cable modems, which are cable modems that have intermittent connectivity problems. A flapping cable modem can indicate either a problem with that particular cable modem, or it could indicate an RF noise problem with the upstream or downstream portion of the cable plant.

The flap-list feature supports any cable modem that conforms to the Data-over-Cable Service Interface Specifications (DOCSIS) because it does use any special messaging to poll cable modems or to request any special information from them. Instead, this feature monitors the normal registration and station maintenance activity that is already performed over a DOCSIS cable network.

This allows the Cisco CMTS to collect the flap-list data without generating additional packet overhead and without impacting network throughput and performance. It also means that although the Flap List Troubleshooting feature is a proprietary feature for Cisco CMTS routers, it is compatible with all DOCSIS-compliant cable modems. In addition, unlike other monitoring methods that use the Simple Network Management Protocol (SNMP), the flap list uses zero bandwidth.

Information in the Flap List

The Flap List Troubleshooting feature tracks the following situations:

- Reinsertions—A reinsertion occurs when the cable modem re-registers more frequently than the user-specified insertion time. A pattern of reinsertions can indicate either potential problems in the downstream or that the cable modem is being improperly provisioned.
- Hits and Misses—A hit occurs when a cable modem successfully responds to the station maintenance messages (MAC-layer "keepalive" messages) that the Cisco CMTS sends out to conform to the DOCSIS standard. A miss occurs when the cable modem does not respond to the request within the user-specified timeout period. A pattern of misses can indicate a potential problem in either the downstream or upstream path, or that a problem can be occurring in the registration process.
- Power Adjustments—DOCSIS cable modems can adjust their upstream transmission power levels to adjust to unstable cable plant signal levels, up to a maximum allowable power level. Repeated power adjustments usually indicate a problem with an amplifier in the upstream return path.

The flap-list feature is automatically enabled, but to use the flap list effectively, the cable system administrator should also typically do the following:

- Set up a script to periodically poll the flap list, for example, every 15 minutes.

- Examine the resulting data and perform trend analysis to identify cable modems that are consistently in the flap list.
- Query the billing and administrative database for cable modem MAC address-to-street address translation and generate a report. The reports can be given to the customer service department or the cable plant's operations and maintenance department. Using these reports, maintenance personnel can quickly discern how characteristic patterns of flapping cable modems, street addresses, and flap statistics indicate which amplifier or feeder lines are faulty. The reports also help to quickly discern whether problems exist in your downstream or upstream path and whether the problem is ingress noise or equipment related.

The flap list provides a quick way to quickly diagnose a number of possible problems. For example, if a subscriber reports a problem, but the flap list for the cable interface that is providing services to them shows little or no flap-list activity, the cable technician can assume that the Cisco CMTS and cable plant are communicating reliably. The problem, therefore, is probably in the subscriber's computer equipment or in the local connection to the cable modem.

Similarly, a cable technician can use the pattern of reinsertions, hits and misses, and power adjustments to quickly troubleshoot the following types of problems:

- If a subscriber's cable modem shows a lot of flap-list activity, it is having some kind of communication problem. Either the cable modem's hardware is faulty, its installation is faulty, the coaxial cable being used is faulty, or some portion of the cable plant that services this cable modem is faulty.
- Focus on the top 10 percent of cable modems that are most active in the flap list, since these are the most likely to indicate consistent and pervasive plant or equipment problems that will continue to disrupt communication with the headend.
- Cable modems with more than 50 power adjustments per day have a suspect upstream path.
- Cable modems with approximately the same number of hits and misses and with a lot of insertions have a suspect downstream path (for example, low level into the cable modem).
- All cable modems incrementing the insertion at the same time indicates a problem with the provisioning servers.
- Cable modems with high cyclic redundancy check (CRC) errors have bad upstream paths or in-home wiring problems.
- Correlating cable modems on the same physical upstream port with similar flap-list statistics can quickly resolve outside plant problems to a particular node or geography.

In addition, the cable network administrators can use the flap list to collect quality control and upstream performance data. Typically, the network operations center (NOC) saves the flap list to a database on a local computer on a daily basis, providing the ability to generate reports that track upstream performance and installation quality control, as well as to provide trend reports on cable plant problems.

Tip The system supports automatic power adjustments. The show cable flap-list and show cable modem commands indicate when the headend cable router has detected an unstable return path for a particular modem and has compensated with a power adjustment. An asterisk (*) appears in the power-adjustment field for a modem when a power adjustment has been made; an exclamation point (!) appears when the modem has reached its maximum power-transmit level and cannot increase its power level any further.

Cisco Cable Manager and Cisco Broadband Troubleshooter

The Flap List Troubleshooting feature is supported by Cisco Cable Manager (CCM), Release 2.0 or later, which is a UNIX-based software suite that manages routers and DOCSIS-compliant cable modems, generates performance reports, troubleshoots connectivity problems, views the network graphically, and edits DOCSIS

configuration files. You can access the CCM locally from the CCM server console or remotely from a UNIX workstation or a PC.

The Flap List Troubleshooting feature also works together with the Cisco Broadband Troubleshooter (CBT), which is a graphical-based application to manage and diagnose problems on the hybrid fiber-coaxial (HFC) network. Radio frequency (RF) technicians can quickly isolate plant and provisioning problems and characterize upstream and downstream trouble patterns, including analyzing flapping modems.

Benefits

The Flap List Troubleshooting feature is a proactive way to manage and troubleshoot problems on an HFC network. Its use of passive monitoring is more scalable and efficient than techniques that send special messages to cable modems or that regularly poll the cable modems using Simple Network Management Protocol (SNMP) commands. Because it uses mechanisms that already exist in a DOCSIS network, it can be used with any DOCSIS-certified cable modem or set-top box.

The flap list provides a cable technician with both real-time and historical cable health statistics for quick, accurate problem isolation and network diagnosis. Using the flap list, a cable technician is able to do the following:

- Quickly learn how to characterize trouble patterns in the hybrid fiber-coaxial (HFC) network.
- Determine which amplifier or feeder line is faulty.
- Distinguish an upstream path problem from a downstream one.
- Isolate an ingress noise problem from a plant equipment problem.

How to Configure Flap List Troubleshooting

This section describes how to configure the flap list operation on the Cisco CMTS. You can use either the command-line interface (CLI) commands or Simple Network Management Protocol (SNMP) commands to configure the flap list, to remove a cable modem from the list, or to clear the flap-list counters.

Configuring Flap List Operation Using the CLI (optional)

To configure the operation of the flap list, use the following procedure, beginning in EXEC mode. This procedure is optional, unless you want to change the default values for the flap list.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**cable flap-list insertion-time** *seconds* **Example:** `Router(config)# cable flap-list insertion-time 3600`	(Optional) Specifies the minimum insertion (registration) time interval in seconds. Any cable modem that makes a registration request more frequently than this period of time is placed in the flap list.
Step 4	**cable flap-list power-adjust threshold** *db* **Example:** `Router(config)# cable flap-list power-adjust threshold 5`	(Optional) Specifies the minimum power adjustment, in dB, that constitutes a flap-list event. **Note** A threshold of less than 2 dB can cause excessive flap-list event recording. If you need to change this parameter from its default, Cisco recommends setting it to 3 dB or higher.
Step 5	**cable flap-list miss-threshold** *misses* **Example:** `Router(config)# cable flap-list miss-threshold 10`	(Optional) Specifies the number of MAC-layer station maintenance (keepalive) messages that can be missed in succession before the CMTS places the cable modem in the flap list. **Note** A high miss rate indicates potential plant problems, such as intermittent upstream problems, fiber laser clipping, or common-path distortion.
Step 6	**cable flap-list aging** *minutes* **Example:** `Router(config)# cable flap-list aging 20160`	(Optional) Specifies how long, in minutes, the Cisco CMTS should keep information for cable modems in the flap list.
Step 7	**cable flap-list size** *number* **Example:** `Router(config)# cable flap-list size 4000`	Specifies the maximum number of cable modems that can be kept in the flap list. **Tip** To avoid wasting processor memory, do not set this value beyond the actual number of cable modems being serviced by the Cisco CMTS.
Step 8	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Clearing the Flap List and Counters Using the CLI (optional)

To clear one or more cable modems from the flap list, or to clear the flap list counters for one or more cable modems (while still keeping the modems in the flap list), use the following procedure, beginning in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**clear cable flap-list** *mac-addr* | **all**} [**save-counters**]
	Example:
	`Router# clear cable flap-list 0102.0304.0506 save-counters`
	Example:
	`Router# clear cable flap-list 000C.0102.0304`
		Clears one or all cable modems from the flap list.

Step 3	**clear cable modem** {*mac-addr* | *ip-addr* | [**cable** *interface*] **all** | **oui***string* | **reject**} } **counters**
	Example:
	`Router# clear cable modem 172.12.23.45 counters`
	Example:
	`Router# clear cable modem oui Cisco counters`
	Example:
	`Router# clear cable modem reject counters`
	Example:
	`Router# clear cable modem c4/0 counters`
		Sets the flap-list counters to zero for one or more CMs.

	Command or Action	Purpose
	Example:	

Enabling or Disabling Power Adjustment Using the CLI (optional)

The Cisco CMTS can automatically monitor a cable modem's power adjustments and determine whether a particular cable modem requires a chance in the power adjustment method. To enable a cable interface to make automatic power adjustments, and to set the frequency threshold for when those adjustments are made, use the following procedure, beginning in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface cable** *x/y* **Example:** `Router(config)# interface cable 4/0`	Enters cable interface configuration mode for the specified cable interface.

Step 4	**cable upstream** *n* **power-adjust** {**continue** *pwr-level* | **noise** *perc-pwr-adj* | **threshold value**}
	Example:
	`Router(config-if)# cable upstream 0 power-adjust threshold 2`
	Example:
	`Router(config-if)# cable upstream 0 power-adjust noise 50`
		Enables automatic power adjustment on an upstream port for this cable interface.
		Note Repeat 4 for each upstream port on the cable interface.

Step 5	**cable upstream** *n* **freq-adj averaging** *percent* **Example:** `Router(config-if)# cable upstream 0`	Specifies the percentage of frequency adjustment packets needed to change the adjustment method from the regular

	Command or Action	Purpose
	`freq-adj averaging 50`	power-adjustment method to the automatic power adjustment method.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

What to do next

Caution The default settings are adequate for system operation. Amplitude averaging is an automatic procedure. In general, Cisco does not recommend that you adjust values. Cisco does recommend, however, that you clean up your cable plant should you encounter flapping cable modems.

Note In some instances, you might adjust certain values for the **cable upstream power-adjust** command: If CMs cannot complete ranging because they have reached maximum power levels, increase the **continue** *pwr-level* parameter beyond the default value of 2 dB. Values larger than 10 dB on "C" versions of cable interface line cards, or 5 dB on FPGA versions, are not recommended. If the flap list shows CMs with a large number of power adjustments, but the CMs are not detected as "noisy," decrease the **noise** *perc-pwr-adj* value. If too many CMs are unnecessarily detected as "noisy," increase the percentage.

Configuring Flap List Operation Using SNMP (optional)

To configure the Flap List Troubleshooting feature on the Cisco CMTS using SNMP, set the appropriate cssFlapObjects attributes in the CISCO-CABLE-SPECTRUM-MIB. the table lists each of the configurable attributes:

Table 235: Flap-List Configuration Attributes

Attribute	Type	Range	Description
ccsFlapListMaxSize	Integer32	1 to 65536[12]	The maximum number of modems that a flap list can support per line card. The default is 100.[13]
ccsFlapListCurrentSize	Integer32	1 to 65536	The current number of modems in the flap list.[14]
ccsFlapAging	Integer32	1 to 86400	The flap entry aging threshold in minutes. The default is 10080 minutes (180 hours or 7 days).

Attribute	Type	Range	Description
ccsFlapInsertionTime	Integer32	60 to 86400	The worst-case insertion time, in seconds. If a cable modem has not completed the registration stage within this interval, the cable modem is inserted into the flap list. The default value is 90 seconds.
ccsFlapPowerAdjustThreshold	Integer32	1 to 10	When the power of the modem is adjusted beyond the power adjust threshold, the modem is inserted into the flap list.
ccsFlapMissThreshold	Unsigned32	1 to 12	When a cable modem does not acknowledge this number of consecutive MAC-layer station maintenance (keepalive) messages, the cable modem is placed in the flap list.

[12] The allowable range when using SNMP for these parameters is 1 to 65536 (a 32-bit value), but the valid operational range is 1 to 8191.

[13] This value is the same as set by the **cable flap-list size** command and is applied only to the command output. The flap list entries displayed via SNMP are not affected by this.

[14] The number of SNMP entries is the same as this value. The number of the CLI entries depends on the value set by **ccsFlapListMaxSize**.

Note **ccsFlapListMaxSize** controls the display of the flap list per downstream cable interface. As long as the number of flap list entries per line card does not exceed 8191, these entries will be stored in the system, and will not be displayed via CLI.

ccsFlapListCurrentSize reflects the number of flap list entries of all the line cards that in the system, regardless of their visibility to the CLI.

Clearing the Flap List and Counters Using SNMP (optional)

To remove a cable modem from the flap list or to clear one or all of the flap-list counters, set the appropriate cssFlapObjects attributes in the CISCO-CABLE-SPECTRUM-MIB. the table lists the attributes that clear the SNMP counters.

Table 236: Attributes to Clear the Flap List

Attribute	Type	Description
ccsFlapResetAll	Boolean	Setting this object to True (1) resets all flap-list counters to zero.
ccsFlapClearAll	Boolean	Setting this object to True (1) removes all cable modems from the flap list, and destroys all entries in the ccsFlapTable. If a modem keeps flapping, the modem is added again into the flap list as a new entry.

Note The ccsFlapLastClearTime attribute contains the date and time that the entries in the ccsFlapTable table were last cleared.

How to Monitor and Troubleshoot Using Flap Lists

Displaying the Flap List Using the show cable flap-list Command

To display the current contents of the flap list, use the show cable flap-list command in privileged EXEC mode. This command has the following syntax:

- **show cable flap-list**—Displays the complete flap list.
- **show cable flap-list sort-interface**—Displays the complete flap list sorted by cable interface.
- **show cable flap-list cable** *interface* **upstream** *port*—Displays the flap list for a specific cable interface, or for a specific upstream port on that cable interface.

To change the way the output is sorted, add one of the following optional keywords:

- **sort-flap**—Sorts the output by the number of times that the cable modem has flapped.
- **sort-time**—Sorts the output by the most recent time that the cable modem flapped.

The following example shows typical output of the **show cable flap-list** command.

```
Router# show cable flap-list
Mac Addr       CableIF Ins    Hit   Miss    CRC  P-Adj   Flap    Time
0010.9500.461f C1/0 U1 56  18857    887      0      1    116 Jun 1  14:09:12
0010.9500.446e C1/0 U1 38  18686   2935      0      1     80 Jun 2  19:03:57
0010.9500.38ec C1/0 U2 63  18932   1040      0      8    138 Jun 2  23:50:53
0010.9500.4474 C1/0 U2 65  18913   1053      0      3    137 Jun 2  09:30:09
0010.9500.4672 C1/0 U2 56  18990   2327      0      6    124 Jun 2  10:44:14
0010.9500.38f0 C1/0 U2 50  18964   2083      0      5    111 Jun 2  20:46:56
0010.9500.e8cb C1/0 U2  0   6537    183      0      1      5 Jun 2  22:35:48
0010.9500.38f6 C1/0 U3 50  19016   2511      0      2    104 Jun 2  07:46:31
0010.9500.4671 C1/0 U3 43  18755   3212      1      1     89 Jun 1  19:36:20
0010.9500.38eb C1/0 U0 57  36133   1608      0      6    126 Jun 2  20:04:58
0010.9500.3ce2 C1/0 U0 44  35315   1907      0      4     99 Jun 2  16:42:47
0010.9500.e8d0 C1/0 U2  0  13213    246      0      1      5 Jun 3  04:15:30
0010.9500.4674 C1/0 U2 56  36037   2379      0      4    121 Jun 3  00:34:12
0010.9500.4677 C1/0 U2 40  35781   2381      0      4     91 Jun 2  12:14:38
0010.9500.4614 C1/0 U2 40  21810   2362      0    502    586 Jun 2  21:43:02
0010.9500.3be9 C1/0 U2 63  22862    969      0      0    128 Jun 1  14:09:03
0010.9500.4609 C1/0 U2 55  22723   2127      0      0    112 Jun 1  14:08:02
0010.9500.3cb8 C1/0 U2 49  22607   1378      0      0    102 Jun 1  14:08:58
0010.9500.460d C1/0 U3 46  22477   2967      0      2     96 Jun 2  17:03:48
0010.9500.3cba C1/0 U3 39  22343   3058      0      0     81 Jun 1  14:13:16
0010.9500.3cb4 C1/0 U3 38  22238   2936      0      0     79 Jun 1  14:09:26
0010.9500.4612 C1/0 U3 38  22306   2928      0      0     79 Jun 1  14:09:29
Router#
```

Displaying the Flap List Using the show cable modem flap Command

To display the contents of the flap list for a specific cable modem, use the **show cable modem flap** command in privileged EXEC mode. This command has the following syntax:

- **show cable modem** [*ip-address* | *mac-address*] **flap**—Displays the flap list for a specific cable modem, as identified by its IP address or MAC address.
- **show cable modem cable***interface* [**upstream** *port*] **flap**—Displays the flap list for all cable modems on a specific cable interface.

Note The **show cable modem flap** command displays information similar to that shown by the **show cable flap-list** command, except it displays this information on a per-modem basis.

The following example shows sample output for the **show cable modem flap** command for a particular cable modem:

```
Router# show cable modem 0010.7bb3.fcd1 flap
MAC Address    I/F       Ins   Hit   Miss  CRC   P-Adj   Flap  Time
0010.7bb3.fcd1 C5/0/U5   0     36278 92    0     369     372   Jun 1  13:05:23 (18000msec)
```

The following example shows sample output for the **show cable modem flap** command for all cable modems on a specific cable interface:

```
Router# show cable modem cable 6/0/0 flap
MAC Address    I/F           Ins   Hit   Miss  CRC   P-Adj   Flap  Time
0025.2e34.4386 C6/0/0/U0     0     46778 3980  0      0      0     (14212 msec)
0025.2e2f.d4b6 C6/0/0/U0     0     48002 1899  0      0      0     (18000 msec)
0025.2e2f.d4de C6/0/0/U0     0     48098 1889  0      0      0     (19552 msec)
0023.bee1.e96b C6/0/0/U0     0     46658 4351  0      0      0     (22432 msec)
0025.2e2f.d4d8 C6/0/0/U0     0     21979 781   0      0      0     ( -- )
0025.2e2f.d48c C6/0/0/U0     0     48048 1835  0      0      0     ( -- )
0025.2e2f.d490 C6/0/0/U0     0     48029 1819  0      0      0     ( -- )
```

Displaying the Flap List Using SNMP

To display the contents of the flap list using SNMP, query the ccsFlapTable table in the CISCO-CABLE-SPECTRUM-MIB. This table contains an entry for each cable modem. the table briefly describes each attribute in this table.

Table 237: cssFlapTable Attributes

Attribute	Type	Description
cssFlapMacAddr	MacAddress	MAC address of the cable modem's cable interface. Identifies a flap-list entry for a flapping cable modem.
ccsFlapUpstreamIfIndex	InterfaceIndex	Upstream being used by the flapping cable modem.
ccsFlapDownstreamIfIndex	InterfaceIndex	Downstream being used by the flapping cable modem.
ccsFlapLastFlapTime	DateAndTime	Time stamp for the last time the cable modem flapped.
ccsFlapCreateTime	DateAndTime	Time stamp that this entry was added to the table.
ccsFlapRowStatus	RowStatus	Control attribute for the status of this entry.

Attribute	Type	Description
ccsFlapInsertionFailNum	Unsigned32	Number of times the CM comes up and inserts itself into the network. This counter is increased when the time between initial link establishment and a reestablishment was less than the threshold parameter configured using the **cable flap-list insertion-time** command or ccsFlapInsertionTime attribute. When the cable modem cannot finish registration within the insertion time (ccsFlapInsertionTime), it resends the Initial Maintenance packet. When the CMTS receives the packet sooner than expected, the CMTS increments this counter.
ccsFlapHitNum	Unsigned32	Number of times the CM responds to MAC-layer station maintenance (keepalive) messages. (The minimum hit rate is once per 30 seconds.)
ccsFlapMissNum	Unsigned32	Number of times the CM misses and does not respond to a MAC-layer station maintenance (keepalive) message. An 8 percent miss rate is normal for the Cisco cable interface line cards. If the CMTS misses a ranging request within 25 msec, then the miss number is incremented.
ccsFlapCrcErrorNum	Unsigned32	Number of times the CMTS upstream receiver flagged a packet with a CRC error. A high value indicates that the cable upstream may have a high noise level. The modem may not be flapping yet, but this could become a possible problem.
ccsFlapPowerAdjustmentNum	Unsigned32	Number of times the cable modem upstream transmit power is adjusted during station maintenance. When the adjustment is greater than the power-adjustment threshold, the number is incremented.
ccsFlapTotalNum	Unsigned32	Number of times a modem has flapped, which is the sum of the following: • When ccsFlapInsertionFailNum is increased • When the CMTS receives a miss followed by a hit • When ccsFlapPowerAdjustmentNum is increased
ccsFlapResetNow	Boolean	Setting this object to True (1) resets all flap-list counters to zero.
ccsFlapLastResetTime	DateAndTime	Time stamp for when all the counters for this particular entry were reset to zero.

Displaying Flap-List Information for Specific Cable Modems

To use SNMP requests to display flap-list information for a specific cable modem, use the cable modem's MAC address as the index to retrieve entries from the ccsFlapTable. Use the following procedure to retrieve flap-list entries for a particular cable modem.

Procedure

Step 1 Convert the cable modem's MAC address into a dotted decimal string. For example, the MAC address 000C.64ff.eb95 would become 0.12.100.255.235.149.

Step 2 Use the dotted decimal version of the MAC address as the instance for requesting information from the ccsFlapTable. For example, to retrieve the ccsFlapHits, ccsFlapMisses, and ccsFlapPowerAdjustments values for this cable modem, you would make an SNMP request for the following objects:

- ccsFlapHits.0.12.100.255.235.149
- ccsFlapMisses.0.12.100.255.235.149
- ccsFlapPowerAdjustments.0.12.100.255.235.149

Example

Assume that you want to retrieve the same flap-list information as the **show cable flap-list** command for a cable modem with the MAC address of 000C.64ff.eb95:

```
Router# show cable flap-list
MAC Address     Upstream     Ins   Hit   Miss  CRC   P-Adj Flap  Time
000C.64ff.eb95  Cable3/0/U4  3314  55605 50460 0    *42175 47533 Jan 27 02:49:10
Router#
```

Use an SNMP tool to retrieve the ccsFlapTable and filter it by the decimal MAC address. For example, using the standard Unix **getone** command, you would give the following command:

```
csh% getmany -v2c 192.168.100.121 public ccsFlapTable | grep 0.12.100.255.235.149

ccsFlapUpstreamIfIndex.0.12.100.255.235.149 = 15
ccsFlapDownstreamIfIndex.0.12.100.255.235.149 = 17
ccsFlapInsertionFails.0.12.100.255.235.149 = 3315
ccsFlapHits.0.12.100.255.235.149 = 55608
ccsFlapMisses.0.12.100.255.235.149 = 50460
ccsFlapCrcErrors.0.12.100.255.235.149 = 0
ccsFlapPowerAdjustments.0.12.100.255.235.149 = 42175
ccsFlapTotal.0.12.100.255.235.149 = 47534
ccsFlapLastFlapTime.0.12.100.255.235.149 = 07 d4  01 1b   02 33  1a 00
ccsFlapCreateTime.0.12.100.255.235.149 = 07 d4  01 16   03 23  22 00
ccsFlapRowStatus.0.12.100.255.235.149 = active(1)
ccsFlapInsertionFailNum.0.12.100.255.235.149 = 3315
ccsFlapHitNum.0.12.100.255.235.149 = 55608
ccsFlapMissNum.0.12.100.255.235.149 = 50460
ccsFlapCrcErrorNum.0.12.100.255.235.149 = 0
ccsFlapPowerAdjustmentNum.0.12.100.255.235.149 = 42175
ccsFlapTotalNum.0.12.100.255.235.149 = 47534
ccsFlapResetNow.0.12.100.255.235.149 = false(2)
ccsFlapLastResetTime.0.12.100.255.235.149 = 07 d4  01 16   03 20  18 00
csh%
```

To request just one particular value, use the decimal MAC address as the instance for that object:

```
csh% getone -v2c 172.22.85.7 public ccsFlapMisses.0.12.100.255.235.149

ccsFlapMisses.0.12.100.255.235.149 = 50736
csh %
```

Troubleshooting Suggestions

This section provides tips on how to interpret the flap-list counters, as well as how to determine the optimum power level for a flapping cable modem.

Troubleshooting Tips

This section includes suggestions on how to interpret different network conditions based on the flap-list statistics:

- Condition 1: Low miss or hit ratio, low insertion, low P-Adj, low flap counter, and old time stamp. Analysis: This exhibits an optimal network situation.
- Condition 2: High ratio of misses over hits (> 10 percent). Analysis: Hit and miss analysis should be done after the Ins count stops incrementing. In general, if the hit and miss counts are about the same order of magnitude, the upstream can be experiencing noise. If the miss count is greater, then the modem is probably dropping out frequently and not completing registration. The upstream or downstream might not be stable enough for reliable link establishment. Very low hits and miss counters and high insertion counters indicate provisioning problems.
- Condition 3: Relatively high power-adjustment counter. Analysis: Indicates that the power-adjustment threshold is probably set at default value of 2 dB. The modem transmitter step size is 1.5 dB, but the headend can command 0.25 dB step sizes. Tuning your power threshold to 6 dB is recommended to decrease irrelevant entries in the flap list. The power-adjustment threshold can be set using cable flap power threshold <0-10 dB> in the Cisco IOS global configuration mode. A properly operating HFC network with short amplifier cascades can use a 2 to 3 dB threshold.
- Condition 4: High P-Adj and CRC errors. Analysis: This condition can indicate that the fiber node is clipping the upstream return laser. Evaluate the modems with the highest CRC count first. If the modems are not going offline (Ins = 0), this is not noticed by subscribers. However, they could receive slower service due to dropped IP packets in the upstream. This condition also results in input errors on the Cisco CMTS router cable interface.
- Condition 5: High insertion rate. Analysis: If link reestablishment happens too frequently, the modem is usually having a registration problem. This is indicated by a high Ins counter, which tracks the Flap counter.

Performing Amplitude Averaging

The CMTS uses an averaging algorithm to determine the optimum power level for a cable modem with low carrier-to-noise ratio that is making excessive power adjustments—known as flapping. To avoid dropping flapping cable modems, the CMTS averages a configurable number of RNG-REQ messages before it makes power adjustments. By compensating for a potentially unstable return path, the CMTS maintains connectivity with affected cable modems. You can interpret these power adjustments, however, as indicating unstable return path connections.

The **show cable flap-list** and **show cable modem** commands are expanded to indicate to which paths the CMTS is making power adjustments and which modems have reached maximum transmit power settings. These conditions indicate unstable paths that should be serviced.

The following example shows the output of the **show cable flap-list** command:

```
Router# show cable flap-list
MAC Address     Upstream     Ins   Hit   Miss  CRC   P-Adj Flap  Time
 0010.7bb3.fd19  Cable1/0/U1  0     2792  281   0     *45    58    Jul 27 16:54:50
 0010.7bb3.fcfc  Cable1/0/U1  0     19    4     0     !43    43    Jul 27 16:55:01
 0010.7bb3.fcdd  Cable1/0/U1  0     19    4     0     *3     3     Jul 27 16:55:01
```

The asterisk (*) indicates that the CMTS is using the power-adjustment method on this modem. An exclamation point (!) indicates that the modem has reached maximum transmit power.

Output of the **show cable modem** command appears below:

```
Router# show cable modem
 Interface   Prim Online     Timing Rec    QoS CPE IP address      MAC address
             Sid  State      Offset Power
 Cable1/0/U0 1    online     2257    0.00  3   0   10.30.128.142   0090.8330.0217
 Cable1/0/U0 2    online     2262  *-0.50  3   0   10.30.128.145   0090.8330.020f
 Cable1/0/U0 3    online     2260    0.25  3   0   10.30.128.146   0090.8330.0211
 Cable1/0/U0 4    online     2256   *0.75  3   0   10.30.128.143   0090.8330.0216
 Cable1/0/U0 5    online     2265   *0.50  3   0   10.30.128.140   0090.8330.0214
 Cable1/0/U0 6    online     2256    0.00  3   0   10.30.128.141   0090.8330.0215
 Cable1/0/U0 7    online     4138  !-1.00  3   1   10.30.128.182   0050.7366.124d
 Cable1/0/U0 8    online     4142  !-3.25  3   1   10.30.128.164   0050.7366.1245
 Cable1/0/U0 9    online     4141  !-3.00  3   1   10.30.128.185   0050.7366.17e3
 Cable1/0/U0 10   online     4142  !-2.75  3   0   10.30.128.181   0050.7366.17ab
 Cable1/0/U0 11   online     4142  !-3.25  3   1   10.30.128.169   0050.7366.17ef
```

Similar to the **show cable flap-list** command display, the * symbol in the **show cable modem** command output indicates that the CMTS is using the power-adjustment method on this CM. The ! symbol indicates that the CM has reached maximum transmit power.

Using Other Related Commands

The following related Cisco IOS commands can be used to do maintenance on or display information about a cable modem.

- The following clears the counters for a cable modem (or all cable modems) in the station maintenance list:

```
clear cable modem {mac-addr | ip-addr | all} counters
```

- The following displays the QoS, modem status, In and Out octets, IP and MAC addresses per SID:

```
show int cable slot/port sid
```

- The following drops the modem's RF link by removing a modem from the keepalive polling list. This forces the modem to reset. Note the warning below.

```
clear cable-modem {mac-addr | ip-addr | all} reset
```

Tip The **clear cable-modem** *all* **reset** command causes all modems to go offline and disrupt service for your users. It is best used in a test or nonproduction environment.

- The following uses a MAC-layer ping to determine if the cable modem is online. It uses smaller data units on the wire than a standard IP ping, resulting in lower overhead. This command works even if the IP layer in the modem is down or has not completed registration:

```
ping DOCSIS cable-modem mac-addr | IP address
```

- The following displays the timing offset, receive power, and QoS values by cable interface, SID, and MAC address:

```
show cable modem [ip-address | MAC-address]
```

- The following displays the current allocation table and frequency assignments:

```
show cable spectrum-group [spectrum group number]
```

- The following displays maximum, average, and minimum percent of online time and offline time for a given SID on a given cable router interface:

```
show int slot/port sid connectivity
```

- The following command displays input and output rates, input errors, CRC, frames, overruns, underruns, collisions, interface resets. High input errors in the CMTS retrieved from this query suggest noisy upstream. In older versions of the chassis, loose midplane and line card screws caused a similar problem:

```
show interface slot/downstream-port
```

- The following command displays upstream packet discards, errors, error-free packets, correctable and uncorrectable errors, noise, and micro-reflection statistics.

```
show interface slot/downstream-port upstream
```

Configuration Examples for Flap List Troubleshooting

The following excerpt from a configuration file shows a typical flap-list configuration:

```
!
cable flap-list insertion-time 120
cable flap-list power-adjust threshold 3
cable flap-list miss-threshold 4
cable flap-list aging 8
cable flap-list size 8191
...
```

Additional References

For additional information related to the Flap List Troubleshooting feature, refer to the following references:

Related Documents

Related Topic	Document Title
CMTS Command Reference	Cisco CMTS Cable Command Reference
Cisco Broadband Troubleshooter	http://www.cisco.com/c/en/us/support/cloud-systems-management/broadband-trou

Standards

Standards[15]	Title
ANSI/SCTE 22-1 2012 (formerly SP-RFI-C01-011119)	Data-Over-Cable Service Interface Specification DOCSIS 1.0 Radio Frequency Interface (RFI)
SP-RFIv1.1-I08-020301	Data-over-Cable Service Interface Specifications Radio Frequency Interface Specification
SP-BPI+-I08-020301	DOCSIS Baseline Privacy Interface Plus Specification

[15] Not all supported standards are listed.

MIBs

MIBs[16]	MIBs Link
CISCO-CABLE-SPECTRUM-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://tools.cisco.com/ITDIT/MIBS/servlet/index

[16] Not all supported MIBs are listed.

RFCs

Description	Link
No new or modified RFCs are supported by this feature.	To locate and download Request for Comments (RFCs) and Internet Drafts, see the Internet Engineering Task Force (IETF) web site at the following URL: http://www.ietf.org/index.html

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

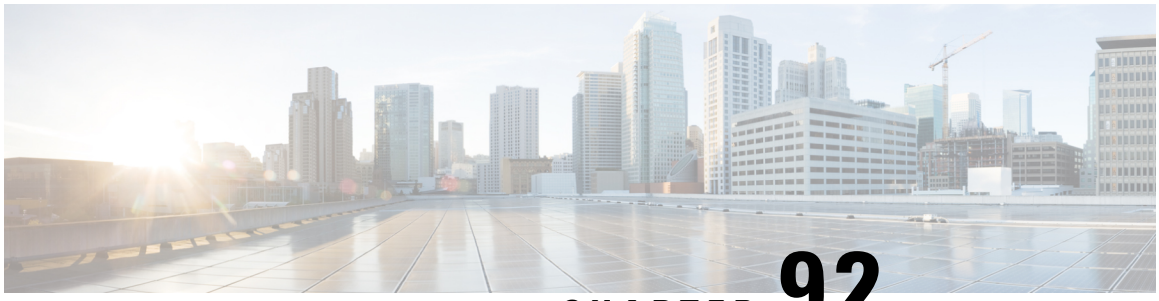
Feature Information for Flap List Troubleshooting

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 238: Feature Information for Flap List Troubleshooting

Feature Name	Releases	Feature Information
Flap List Troubleshooting	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 92

Maximum CPE and Host Parameters

This document describes how to use different methods to control subscriber access that are allowed by the Data-over-Cable Service Interface Specifications (DOCSIS) for use on cable networks.

- Finding Feature Information, on page 1441
- Hardware Compatibility Matrix for Cisco cBR Series Routers, on page 1441
- Information About the MAX CPE and Host Parameters, on page 1442
- How to Configure the MAX CPE and Host Parameters, on page 1446
- Configuration Examples, on page 1448
- Additional References, on page 1449
- Feature Information for Maximum CPE and Host Parameters, on page 1450

Finding Feature Information

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Hardware Compatibility Matrix for Cisco cBR Series Routers

Note The hardware components introduced in a given Cisco IOS-XE Release are supported in all subsequent releases unless otherwise specified.

Table 239: Hardware Compatibility Matrix for the Cisco cBR Series Routers

Cisco CMTS Platform	Processor Engine	Interface Cards
Cisco cBR-8 Converged Broadband Router	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 Supervisor : • PID—CBR-CCAP-SUP-160G • PID—CBR-CCAP-SUP-60G • PID—CBR-SUP-8X10G-PIC	**Cisco IOS-XE Release 16.5.1 and Later Releases** Cisco cBR-8 CCAP Line Cards: • PID—CBR-LC-8D30-16U30 • PID—CBR-LC-8D31-16U30 • PID—CBR-RF-PIC • PID—CBR-RF-PROT-PIC • PID—CBR-CCAP-LC-40G-R Cisco cBR-8 Downstream PHY Modules: • PID—CBR-D30-DS-MOD • PID—CBR-D31-DS-MOD Cisco cBR-8 Upstream PHY Modules: • PID—CBR-D30-US-MOD

Information About the MAX CPE and Host Parameters

The DOCSIS specification includes a number of provisions to allow service providers to control the number of subscribers who can access the network through any particular cable modem.

The following are the parameters that controls the number of CPE that can access the network:

Note In addition, the DOCSIS configuration file contains a Network Access parameter that specifies whether the CPE devices behind the cable modem can access the cable network. If the Network Access parameter is set to Disabled, no CPE devices behind a cable modem are able to access the network.

Tip Also, the Cisco CMTS lists offline cable modems in its internal database for 24 hours. The CMTS does not reset the CPE counts for these offline cable modems until the 24 hour period expires and the cable modems come back online. If the cable modems come back online before the 24 hour period expires, the CMTS continues to use the existing CPE counts.

All of these methods are similar in purpose, but they are configured differently and have a different impact on cable modems and their CPE devices.

The cable modem enforces the MAX CPE value, the CMTS enforces the MAX Host, MAX CPE IP, and MAX CPE IPv6 values.

Note The MAX CPE parameter provides Layer 2 control of CPE devices. The MAX CPE IP and MAX CPE IPv6 parameters provide Layer 3 control of CPE devices. The two methods are complimentary but not otherwise related.

MAX CPE

The MAX CPE is a required parameter and used to control the number of CPE devices that can access the network during the current session. In DOCSIS 1.0 cable networks, the MAX CPE parameter is the primary means of controlling the number of CPE devices that can connect to the cable network using any particular cable modem. This parameter is configured in the DOCSIS configuration file (TLV 18). If this parameter is not specified in the DOCSIS configuration file, it defaults to a value of 1.

Note In DOCSIS 1.1 cable networks, the CMTS ignores the MAX CPE parameter that is specified in the DOCSIS configuration file, and uses the MAX Host parameter instead.

Each time a new CPE device attempts to connect to the cable network, the cable modem logs the hardware (MAC) address. If the cable modem has not reached the MAX CPE number of MAC addresses, the new CPE device is allowed to access the network. If the cable modem has reached the MAX CPE limit, it drops the traffic from any additional CPE devices.

By default, the cable modem learns new MAC addresses on a first-come, first-served basis. You can also preconfigure the allowable MAC addresses for CPE devices by entering those MAC addresses in the DOCSIS configuration file (TLV 14). These cable modem gives these preconfigured MAC addresses preference in connecting to the network.

The DOCSIS specification does not allow cable modems to age out MAC addresses, so a MAC address stays in the log table of the cable modem until the cable modem is reset. You should therefore think of this parameter as specifying the maximum number of CPE devices that can connect during any particular session, instead of the maximum number of CPE devices that can simultaneously connect to the cable network.

For example, if you set MAX CPE to 2, a customer could use their cable modem to connect a maximum of two CPE devices (two MAC addresses) to the cable network. A customer could choose to connect two PCs simultaneously to their cable modem and use both to access the network.

However, if the customer then disconnected these PCs and connected two new PCs, the cable modem would not allow the new PCs to come online, because they would be the third and fourth MAC addresses that are connected to the cable modem. The customer would have to reset the cable modem before being able to use the new PCs.

Note The MAX CPE value, if present, must be a positive integer in DOCSIS 1.0 configuration files. This parameter can be zero in DOCSIS 1.1 configuration files, but if so, the cable modem uses a MAX CPE value of 1. If the MAX CPE parameter is not present in either type of DOCSIS configuration file, it defaults to 1.

MAX Host

The MAX Host parameter is an optional parameter and is configured on the Cisco CMTS and specifies the maximum number of CPE devices (MAC addresses) that the CMTS will allow to have network access. You can control this parameter for individual cable modems, for all cable modems on a particular cable interface, or for all cable modems on the Cisco CMTS, depending on the CLI command being used:

- **cable modem max-cpe**—Configures MAX Host for all cable modems on the Cisco CMTS. You can use the **unlimited** keyword to specify that the Cisco CMTS should not enforce a MAX Host limit for cable modems.

When this is enabled, the Cisco CMTS learns a MAC address the first time that the CPE device accesses the cable network. After the Cisco CMTS has logged the maximum number of MAC addresses specified by a MAX Host parameter, it drops all traffic from CPE devices that have any other MAC address.

Tip In DOCSIS 1.1 cable networks, when both the MAX CPE IP and MAX Host parameters are configured, the Cisco CMTS uses the lesser value to determine the maximum number of CPE devices that are allowed behind each cable modem. By default, MAX Host is set to 16.

Note The entire MAX Host address table is cleared whenever the Cisco TS is reset. You can also clear an entry for a particular CPE device using the **clear cable host** command.

Specifying an Unlimited Value for Max Host

The **cable modem max-cpe** command, which affects all cable modems on the CMTS, supports the **unlimited** keyword, which specifies that the CMTS should not enforce any limit on CPE devices. When you configure the CMTS with the unlimited **keyword**, this setting, you are allowing cable modems to support any number of CPE devices.

Do not use the **unlimited** option without also specifying the proper value for MAX CPE in the DOCSIS configuration file, so that each cable modem can control the maximum number of CPE devices it supports. In addition, to prevent users from requesting an unlimited number of IP address, be sure to configure the DHCP servers so that they control how many IP addresses are assigned to the CPE devices behind each cable modem.

MAX CPE IP

The MAX CPE IP parameter is applicable only in DOCSIS 1.1 cable networks and is an optional parameter. This parameter specifies whether the cable modem should perform IP address filtering on the CPE devices. If so, this attribute also specifies the maximum number of simultaneous IP addresses that are permitted behind the modem at any one time.

The MAX CPE IP parameter is configured in the DOCSIS configuration file (TLV 35), or by using SNMP commands to set the docsDevCpeIpMax attribute (in DOCS-CABLE-DEVICE-MIB) for the cable modem. By default, this parameter is not enabled and the Cisco CMTS does not actively manage CPE devices, unless you enable the use of the MAX CPE IP parameter by using the **cable submgmt default active** command. The **cable submgmt default max-cpe** command can be used to limit the number of IP addresses behind the cable modem.

If this feature is enabled, the cable modem learns the allowable IP addresses the first time that the CPE device sends an IP packet out into the network. The IP addresses are added to the docsDevFilterCpeTable table. This address table is cleared automatically when the cable modem is reset or powered off, or you can manually clear the IP address table by setting the docsSubMgtCpeControlReset attribute in the appropriate table entry for this cable modem.

Tip The CMTS uses the MAX CPE IP value as part of its own filtering process, but the two filters operate independently on the cable modem and CMTS.

MAX CPE IPv6

The MAX CPE IPv6 parameter is an optional parameter and specifies the maximum number of simultaneous IPv6 addresses that are permitted for a cable modem at any time.

The MAX CPE IPv6 parameter is configured in the DOCSIS 3.0 configuration file (TLV 63), or by using the SNMP commands to set the docsSubmgt3BaseCpeMaxIpv6PrefixDef attribute (in DOCS-SUBMGT3-MIB) for the cable modem. By default, this parameter is not enabled and the Cisco CMTS does not actively manage CPE devices, unless the use of the MAX CPE IPv6 parameter is enabled by using the **cable submgmt default active** command. The **cable submgmt default max-ipv6-cpe** command can be used to limit the number of IPv6 addresses allowed behind a cable modem.

When the MAX CPE IPv6 feature is enabled, the cable modem learns the allowable IPv6 addresses the first time that the CPE device sends an IPv6 packet out into the network. The IPv6 addresses are added to the IPv6 address table. The address table is cleared automatically when the cable modem is reset or powered off.

Interoperation of the Maximum CPE Parameters

The different methods of CPE control can all be active simultaneously. They can interact with one another but do not conflict with one another. The table lists each method and compares their characteristics.

Table 240: Comparison of the Different Max CPE and Max Host Control Mechanisms

CM Configuration Parameter	Function	CMTS Equivalent	CMTS Enforcement Priority
Network Access Control	Prevents all network access for CPE devices	Cable submgmt default learnable	CMTS overrides CM Config File
MAX CPE	Limits MAC addresses per CM	Cable modem max-hosts	Least restrictive is enforced
MAX CPE IP	Limits IP addresses per CM	Cable submgmt default max-cpe	Most restrictive is enforced
MAX CPE IPv6	Limits IPv6 addresses per CM	Cable submgmt default max-ipv6-cpe	Most restrictive is enforced

The table lists the MAX CPE parameters in order of priority. For example, the Network Access Control and MAX CPE parameters interact as follows:

- If the Network Access Control field for a cable modem is set to Disabled, none of that modem's CPE devices will be able to access the network, regardless of how the other parameters are set.
- If Network Access Control is Enabled and MAX CPE is set to 1 for a cable modem, then a maximum of one CPE device will be able to access the network, no matter how the remaining parameters are configured.

Benefits

- CMTS flexibility allows multiple service operator provisioners, service providers, and other users to synchronize between the CMTS and the cable modem, the maximum number of CPE devices, maximum number of IPv4 addresses, and maximum number of IPv6 addresses that can be connected behind a cable modem.
- Changes can be made by using CLI commands or by using SNMP commands.

How to Configure the MAX CPE and Host Parameters

To reset the maximum number of permitted CPE devices recognized by the CMTS, use one of the following configuration commands. All procedures are optional, depending on the requirements.

Note The CMTS assigns the MAX Host value to a cable modem at the time that the cable modem registers with the CMTS. Changing any of the MAX Host commands affects only cable modems that register after the change.

Configuring the Maximum Number of CPE Devices on the Cisco CMTS

To configure the maximum number of CPE devices per cable modem, use the following procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cable modem max-cpe** [*number* | **unlimited**] **Example:**	Sets the value of the MAX CPE parameter on the Cisco CMTS for all cable interfaces.

	Command or Action	Purpose
	`Router(config)# cable modem max-cpe 8`	The **show cable modem subscriber** displays the MAXIMUM value of cable modem max-cpe and the MAX CPE value in the DOCSIS configuration file of the cable modem. The number of the CPE that can be online is determined based on one of the following aspects: • If the number of the CPE is lower than the MAX CPE value in the DOCSIS configuration file of the cable modem, then the **cable modem max-cpe** command overrides the configuration file value. • If number of the CPE is higher than the MAX CPE value in the DOCSIS configuration file of the cable modem or is set as unlimited, then the value set in the configuration file takes precedence. **Note** If the value in the configuration file is zero and **no cable modem max-cpe** is configured, then no CPE device is able to obtain an IP address.
Step 4	**cable submgmt default active** **Example:** `Router(config)# cable submgmt default active`	Specifies that the CMTS should actively manage CPE devices. The default is the **no** version of this command, so that the CMTS does not actively manage CPE devices.
Step 5	**cable submgmt default max-cpe** *cpe-ip* **Example:** `Router(config)# cable submgmt default max-cpe 4`	(Optional) Specifies the default value for the MAX CPE IP.
Step 6	**cable submgmt default max-ipv6-cpe** *ipv6-num* **Example:** `Router(config)# cable submgmt default max-ipv6-cpe 4`	(Optional) Specifies the default value for the MAX IPv6 CPE.
Step 7	**exit** **Example:**	Exits global configuration mode.

	Command or Action	Purpose
	`Router(config)# exit`	

What to do next

Note Use of the **cable modem max-cpe unlimited** command can open a security hole in the system by enabling denial of service attacks. It could allow a single user to obtain a large number of IP addresses, and thereby cause the entire network to go down after this single user has reserved all available IP addresses.

Configuration Examples

To display the current configuration and status of a cable interface, use the **show running-config** command in privileged EXEC mode. The following is sample output that shows that the CMTS permits up to five CPE devices to use the specified cable interface to pass traffic.

```
interface Cable3/0
 ip address 192.168.1.1 255.255.255.0 secondary
 ip address 10.1.1.1 255.255.255.0
 load-interval 30
 no keepalive
 cable downstream annex B
 cable downstream modulation 256qam
 cable downstream interleave-depth 32
 cable downstream frequency 507000000
 cable upstream 0 frequency 27008000
 cable upstream 0 power-level 0
 cable upstream 0 minislot-size 32
 cable upstream 0 modulation-profile 2
 no cable upstream 0 shutdown
 cable upstream 1 frequency 29008000
 cable upstream 1 power-level 0
 cable upstream 1 channel-width 3200000
 cable upstream 1 minislot-size 4
 no cable upstream 1 shutdown
 cable dhcp-giaddr policy
 cable helper-address 172.17.110.131
end
```

You can also use the **more system:running-config** command to verify the maximum number of permitted CPE devices for a cable interface.

```
CMTS01# more system:running-config
Building configuration...
Current configuration:
!
interface Cable6/0
 ip address 10.1.1.1 255.255.255.0
 no keepalive
 cable insertion-interval 2000
 cable downstream annex B
 cable downstream modulation 64qam
```

```
cable downstream interleave-depth 32
cable downstream symbol-rate 5056941
cable upstream 0 frequency 15008000
cable upstream 0 fec
cable upstream 0 scrambler
no cable upstream 0 shutdown
```

Additional References

For additional information related to configuring the MAX CPE and Host parameters on the Cisco CMTS, refer to the following references:

Related Documents

Related Topic	Document Title
Cisco CMTS Commands	Cisco CMTS Cable Command Reference
Interaction of MAX CPE Parameters	Using the max-cpe Command in the DOCSIS and CMTS

Standards

Standards[17]	Title
SP-RFIv1.1-I08-020301	*Data-Over-Cable Service Interface Specifications Radio Frequency Interface Specification , version 1.1* (http://www.cablelabs.com/cablemodem/)

[17] Not all supported standards are listed.

MIBs

MIBs[18]	MIBs Link
DOCS-CABLE-DEVICE-MIB DOCS-SUBMGT-MIB DOCS-SUBMGT3-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

[18] Not all supported MIBs are listed.

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

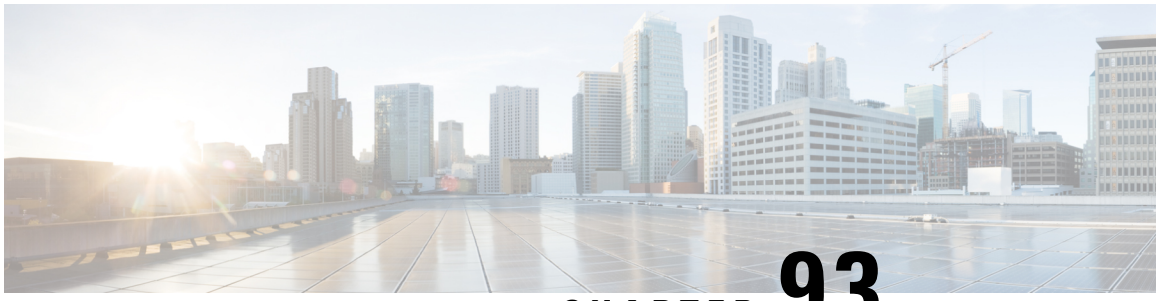
Feature Information for Maximum CPE and Host Parameters

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 241: Feature Information for Maximum CPE and Host Parameters

Feature Name	Releases	Feature Information
Maximum CPE and Host Parameters	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 93

SNMP Background Synchronization

The SNMP Background Synchronization features provides periodic background synchronization of DOCSIS MIB data from line card to Supervisor in order to improve the performance of the SNMP polling of these MIB tables.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/. An account on http://www.cisco.com/ is not required.

Contents

- Information About SNMP Background Synchronization, on page 1451
- How to Configure SNMP Background Synchronization, on page 1452
- Configuring Example for SNMP Background Synchronization, on page 1459
- Reference, on page 1459

Information About SNMP Background Synchronization

To improve SNMP performance, SNMP background synchronization feature is introduced to synchronize the SNMP MIB information between the line card and the Supervisor . It is based on raw socket and uses TCP protocol. The benefits of the SNMP Background Synchronization include:

- Bundles small packets together before sending out, increases IPC channel utilization.
- Use pre-allocated static buffer to send/receive message, avoid buffer allocation at run time.
- In order not to burden CPU when the system is in high load, SNMP background synchronization receive process can sleep based on CPU utilization, so it will not compete with other priority processes.
- Significantly improve SNMP polling performance for supported MIB tables, and reduce the CPU utilization in both Supervisor and line card.

The following MIB tables are supported in SNMP background synchronization:

- docsQosParamSetEntry
- docsIetfQosParamSetEntry
- docsQos3ParamSetEntry
- docsIf3CmtsCmUsStatusEntry
- docsIfCmtsCmStatusEntry
- docsSubMgtCpeControlEntry
- docsSubMgtCmFilterEntry
- cdxCmtsCmStatusExtEntry
- docsLoadBalCmtsCmStatusEntry
- docsIf3CmtsCmRegStatusTable
- docsIfSignalQualityTable
- docsifCmtsServiceTable
- cdxCmtsServiceExtEntry

How to Configure SNMP Background Synchronization

Enabling SNMP Background Synchronization

Before you begin

To use the **cable bgsync** command, you must configure the **service internal** command in global configuration mode.

SNMP background synchronization is enabled by default, use **no cable bgsync active** to disable this feature, and use **cable bgsync active** to enable it again. The following procedure lists detailed steps to enable SNMP background synchronization:

```
enable
configure terminal
cable bgsync active
exit
```

Setting Data Interval

Before you begin

To use the **cable bgsync** command, you must configure the **service internal** command in global configuration mode. Use the **cable bgsync** command carefully as it can impact the CPU utilization.

To set the data intervals for the background synchronization of SNMP MIB data on the Cisco cBR routers, use the **cable bgsync** {**itime** *i-interval*|**ptime** *p-interval*} command in global configuration mode. To disable

background synchronization, use the **no** form of this command. The following procedure lists detailed steps to set data interval:

```
enable
configure terminal
service internal
cable bgsync itime i-interval
cable bgsync ptime p-interval
exit
```

itime is the interval of synchronizing all related MIB tables from line card to Supervisor . The valid range is from 5 to 31536000. The default value is 86400. **ptime** is the interval of synchronizing the changed MIB content from line card to Supervisor .

Verifying SNMP Background Synchronization

- To display the current status of the SNMP background synchronization, use the **show cable bgsync** command as shown in the example below:

```
Router#show cable bgsync
Background Sync is active, uptime is 5 minutes, 14 seconds.
Background Sync last active time is 5 minutes, 14 seconds. ago.
I-packet interval time is 1 day,  P-packet interval time is 5 senconds.
Line Card with bg-sync: 3/0
Line Card working on I syncing:
Last clear cable bg sync counters Time:
Total bytes: 85864
Total background sync packets: 2109
  Ack packets: 0
  Run Ctrl Msg packets: 2
  Data packets: 0
Interval packets: 2002
  I Type packets: 230
  P Type packets: 1772
Bg sync data IPC lost packets: 0

Background Sync statistics for the last 00:07:34
================================================================
 ipc packets 0-30k:      105
 ipc packets 30-60k:     0
 ipc packets 60-100k:    0
 msg per packet average: 20
 msg per packet max:     113
 msg per packet min:     1
 msg per packet under 3: 60
================================================================
type       packets        cpu-total(ms)    avg(us)   max(us)
serv flow  904            3                3         1000
sflog      0              0                0         0
cm         17             0                0         0
cmtx       296            0                0         0
paramset   112            0                0         0
DXIF       298            0                0         0
sid        208            0                0         0
uschan     167            1                5         1000
----------------------------------------------------------------
IPC PKTs   105            4                0      ms 1      ms
================================================================================================
slot type       packets        bytes            pps        Bps        wrong_len_pkts
0    serv flow  0              0                0.0        0.0        0
```

0	sflog	0	0	0.0	0.0	0
0	cm	0	0	0.0	0.0	0
0	cmtx	0	0	0.0	0.0	0
0	paramset	0	0	0.0	0.0	0
0	DXIF	0	0	0.0	0.0	0
0	sid	0	0	0.0	0.0	0
0	uschan	0	0	0.0	0.0	0
1	serv flow	0	0	0.0	0.0	0
1	sflog	0	0	0.0	0.0	0
1	cm	0	0	0.0	0.0	0
1	cmtx	0	0	0.0	0.0	0
1	paramset	0	0	0.0	0.0	0
1	DXIF	0	0	0.0	0.0	0
1	sid	0	0	0.0	0.0	0
1	uschan	0	0	0.0	0.0	0
2	serv flow	0	0	0.0	0.0	0
2	sflog	0	0	0.0	0.0	0
2	cm	0	0	0.0	0.0	0
2	cmtx	0	0	0.0	0.0	0
2	paramset	48	7680	0.0	0.0	0
2	DXIF	0	0	0.0	0.0	0
2	sid	16	512	0.0	0.0	0
2	uschan	0	0	0.0	0.0	0
3	serv flow	904	25104	4.4	115.4	0
3	sflog	0	0	0.0	0.0	0
3	cm	17	981	0.0	2.0	0
3	cmtx	296	8607	0.7	20.6	0
3	paramset	64	8368	0.0	0.0	0
3	DXIF	298	21876	0.9	74.3	0
3	sid	192	4756	0.1	6.8	0
3	uschan	167	5832	0.3	10.7	0
6	serv flow	0	0	0.0	0.0	0

```
6    sflog      0              0              0.0        0.0        0
6    cm         0              0              0.0        0.0        0
6    cmtx       0              0              0.0        0.0        0
6    paramset   0              0              0.0        0.0        0
6    DXIF       0              0              0.0        0.0        0
6    sid        0              0              0.0        0.0        0
6    uschan     0              0              0.0        0.0        0
7    serv flow  0              0              0.0        0.0        0
7    sflog      0              0              0.0        0.0        0
7    cm         0              0              0.0        0.0        0
7    cmtx       0              0              0.0        0.0        0
7    paramset   0              0              0.0        0.0        0
7    DXIF       0              0              0.0        0.0        0
7    sid        0              0              0.0        0.0        0
7    uschan     0              0              0.0        0.0        0
8    serv flow  0              0              0.0        0.0        0
8    sflog      0              0              0.0        0.0        0
8    cm         0              0              0.0        0.0        0
8    cmtx       0              0              0.0        0.0        0
8    paramset   0              0              0.0        0.0        0
8    DXIF       0              0              0.0        0.0        0
8    sid        0              0              0.0        0.0        0
8    uschan     0              0              0.0        0.0        0
9    serv flow  0              0              0.0        0.0        0
9    sflog      0              0              0.0        0.0        0
9    cm         0              0              0.0        0.0        0
9    cmtx       0              0              0.0        0.0        0
9    paramset   0              0              0.0        0.0        0
9    DXIF       0              0              0.0        0.0        0
9    sid        0              0              0.0        0.0        0
9    uschan     0              0              0.0        0.0        0
```

- To display all the SNMP background sync data on Supervisor side or line card side, use the **show cable bgsync sync-info cable** command as shown in the example below:

```
Router#show cable bgsync sync-info cable 9/0/1
 part1 for srv template:
srv_tmp_id   min_rate    max_rate    max_burst
0           0           0           0
1           0           64000       0
2           0           1000000     0
3           0           1000000     3044
4           0           0           3044
5           0           110000000   30000
6           0           0           3044
7           0           2000000000  5000000
8           0           0           3044
 part2 for srv flow:
sfid     prov_qos   adm_qos    act_qos    wb_mode    octets    pkts    delay_pkts
drop_pkts    gate_id    create_time    total_active_time
1        0          0          0          0          0         0          0
0          0               0              0
2        0          0          0          0          0         0          0
0          0               0              0
3        0          0          0          0          0         0          0
0          0               0              0
4        0          0          0          0          0         0          0
0          0               0              0
5        0          0          0          0          0         0          0
0          0               0              0
6        0          0          0          0          0         0          0
0          0               0              0
7        0          0          0          0          0         0          0
0          0               0              0
8        0          0          0          0          0         0          0
0          0               0              0
9        0          0          0          0          0         0          0
0          0               0              0
10       0          0          0          0          0         0          0
0          0               0              0
11       0          0          0          0          0         0          0
0          0               0              0
12       0          0          0          0          0         0          0
0          0               0              0
13       0          0          0          0          0         0          0
0          0               0              0
14       0          0          0          0          0         0          0
0          0               0              0
15       3          3          3          0          0         0          0
0          0               3600           179
16       3          3          3          0          0         0          0
0          0               3600           179
17       3          3          3          0          0         0          0
0          0               3600           179
18       3          3          3          0          0         0          0
0          0               3600           179
19       3          3          3          0          0         0          0
0          0               3600           179
20       3          3          3          0          0         0          0
0          0               3600           179
21       3          3          3          0          0         0          0
0          0               3600           179
22       3          3          3          0          0         0          0
0          0               3600           179
23       3          3          3          0          0         0          0
```

```
0          0             3600          179
24       3          3          3          0          0          0          0
0          0             3600          179
25       3          3          3          0          0          0          0
0          0             3600          179
26       3          3          3          0          0          0          0
0          0             3600          179
27       4          5          5          0          8925       42         0
0          0             12700         88
28       6          7          7          3          0          0          0
0          0             12700         88
29       4          5          5          3          3855       21         0
0          0             11500         100
30       6          7          7          3          0          0          0
0          0             11500         100
31       8          8          8          3          222        3          0
0          0             11500         100
32       4          5          5          3          1277       11         0
0          0             12100         94
33       6          7          7          0          0          0          0
0          0             12100         94
34       4          5          5          0          3851       21         0
0          0             12300         92
35       6          7          7          3          0          0          0
0          0             12300         92
36       8          8          8          0          148        2          0
0          0             12100         94
37       4          5          5          0          3855       21         0
0          0             12700         88
38       6          7          7          3          0          0          0
0          0             12700         88
39       8          8          8          3          222        3          0
0          0             12300         92
40       4          5          5          3          3281       20         0
0          0             13100         84
41       6          7          7          3          0          0          0
0          0             13100         84
42       8          8          8          3          222        3          0
0          0             12700         88
43       8          8          8          3          222        3          0
0          0             12700         88
44       4          5          5          3          3308       21         0
0          0             13100         84
45       6          7          7          3          0          0          0
0          0             13100         84
46       8          8          8          3          296        4          0
0          0             13100         84
47       8          8          8          3          296        4          0
0          0             13100         84
48       4          5          5          3          73         2          0
0          0             14500         70
49       6          7          7          3          0          0          0
0          0             14500         70
50       8          8          8          3          74         1          0
0          0             14500         70
 part3 for sid
 sid_entry[1] sid 1 service_class 2 create_time 127 total_octets 8925
 sid_entry[2] sid 2 service_class 2 create_time 115 total_octets 3855
 sid_entry[3] sid 3 service_class 2 create_time 121 total_octets 1277
 sid_entry[4] sid 4 service_class 2 create_time 123 total_octets 3851
 sid_entry[5] sid 5 service_class 2 create_time 127 total_octets 3855
 sid_entry[6] sid 6 service_class 2 create_time 131 total_octets 3281
 sid_entry[7] sid 7 service_class 2 create_time 131 total_octets 3308
 sid_entry[8] sid 8 service_class 2 create_time 145 total_octets 73
```

```
part4 for cm and cmtx
cm_mac: 68ee.9633.0699, tcsbmp: 0x1, admin_status 1, md_sg_id 0x1510505, rcc_status_id
0x4, rcs_id 0x1520005, tcs_id 0x1 last_reg_time 1444372688, RCP ID:00 10 00 00 10
usch 1, modulation_type 2, rx_power -5, signal_noise 390, time_offset 2085
cm_mac: e448.c70c.96e7, tcsbmp: 0x4, admin_status 1, md_sg_id 0x1510505, rcc_status_id
0x4, rcs_id 0x1520005, tcs_id 0x3 last_reg_time 1444372678, RCP ID:00 10 00 00 08
usch 3, modulation_type 2, rx_power -15, signal_noise 381, time_offset 1785
cm_mac: 0019.474a.c126, tcsbmp: 0x1, admin_status 1, md_sg_id 0x1510505, rcc_status_id
0x0, rcs_id 0x22, tcs_id 0x1 last_reg_time 1444372682, RCP ID:00 00 00 00 00
usch 1, modulation_type 2, rx_power -15, signal_noise 390, time_offset 1792
cm_mac: e448.c70c.982b, tcsbmp: 0x1, admin_status 1, md_sg_id 0x1510505, rcc_status_id
0x4, rcs_id 0x1520005, tcs_id 0x1 last_reg_time 1444372685, RCP ID:00 10 00 00 08
usch 1, modulation_type 2, rx_power -10, signal_noise 390, time_offset 1786
cm_mac: e448.c70c.96d5, tcsbmp: 0x2, admin_status 1, md_sg_id 0x1510505, rcc_status_id
0x4, rcs_id 0x1520005, tcs_id 0x2 last_reg_time 1444372688, RCP ID:00 10 00 00 08
usch 2, modulation_type 2, rx_power -15, signal_noise 381, time_offset 1786
cm_mac: e448.c70c.9819, tcsbmp: 0x1, admin_status 1, md_sg_id 0x1510505, rcc_status_id
0x4, rcs_id 0x1520005, tcs_id 0x1 last_reg_time 1444372692, RCP ID:00 10 00 00 08
usch 1, modulation_type 2, rx_power -10, signal_noise 390, time_offset 1789
cm_mac: e448.c70c.980d, tcsbmp: 0x4, admin_status 1, md_sg_id 0x1510505, rcc_status_id
0x4, rcs_id 0x1520005, tcs_id 0x3 last_reg_time 1444372695, RCP ID:00 10 00 00 08
usch 3, modulation_type 2, rx_power -10, signal_noise 390, time_offset 1783
cm_mac: e448.c70c.96f3, tcsbmp: 0x1, admin_status 1, md_sg_id 0x1510505, rcc_status_id
0x4, rcs_id 0x1520005, tcs_id 0x1 last_reg_time 1444372723, RCP ID:00 10 00 00 04
usch 1, modulation_type 2, rx_power 0, signal_noise 420, time_offset 1798
part5 for dxif info ifnum 1
basedata[1][1]: cmstatusindex 2375681, cm_mac 68ee.9633.0699, cm_ip 0x5011961F, cm_ds_if
59881, cm_us_if 204952
cmregmode 2, cmmodultype 2, cmdocmode 2
basedata[1][2]: cmstatusindex 2375682, cm_mac e448.c70c.96e7, cm_ip 0x5011961D, cm_ds_if
59882, cm_us_if 204954
cmregmode 2, cmmodultype 2, cmdocmode 2
basedata[1][3]: cmstatusindex 2375683, cm_mac 0019.474a.c126, cm_ip 0x50119602, cm_ds_if
59914, cm_us_if 204952
cmregmode 2, cmmodultype 2, cmdocmode 2
basedata[1][4]: cmstatusindex 2375684, cm_mac e448.c70c.982b, cm_ip 0x50119612, cm_ds_if
59881, cm_us_if 204952
cmregmode 2, cmmodultype 2, cmdocmode 2
basedata[1][5]: cmstatusindex 2375685, cm_mac e448.c70c.96d5, cm_ip 0x5011960D, cm_ds_if
59881, cm_us_if 204953
cmregmode 2, cmmodultype 2, cmdocmode 2
basedata[1][6]: cmstatusindex 2375686, cm_mac e448.c70c.9819, cm_ip 0x5011961E, cm_ds_if
59881, cm_us_if 204952
cmregmode 2, cmmodultype 2, cmdocmode 2
basedata[1][7]: cmstatusindex 2375687, cm_mac e448.c70c.980d, cm_ip 0x5011961A, cm_ds_if
59882, cm_us_if 204954
cmregmode 2, cmmodultype 2, cmdocmode 2
basedata[1][8]: cmstatusindex 2375688, cm_mac e448.c70c.96f3, cm_ip 0x5011960E, cm_ds_if
59882, cm_us_if 204952
cmregmode 2, cmmodultype 2, cmdocmode 2
part6 uschan for ifnum 1
usport 1 micro_reflections 0 us_snr 390 snmp_sigq_unerroreds 0 snmp_sigq_correcteds 0
 snmp_sigq_uncorrectables 0
usport 2 micro_reflections 0 us_snr 381 snmp_sigq_unerroreds 0 snmp_sigq_correcteds 0
 snmp_sigq_uncorrectables 0
usport 3 micro_reflections 0 us_snr 390 snmp_sigq_unerroreds 0 snmp_sigq_correcteds 0
 snmp_sigq_uncorrectables 0
usport 4 micro_reflections 0 us_snr 0 snmp_sigq_unerroreds 0 snmp_sigq_correcteds 0
snmp_sigq_uncorrectables 0
```

- To display raw socket interprocess communication (IPC) infrastructure statistics for specified field replaceable unit (FRU), use the **show platform software ios** *slot-id* **socket statistics** command as shown in the example below:

```
Router#show platform software ios R0 socket statistics 0
-------------------------------------------

Session Slot             : 2
Socket FD                : 93
Client ID                : 0
Message Receive Count    : 0
Message Receive Bytes    : 0

-------------------------------------------

Session Slot             : 2
Socket FD                : 93
Client ID                : 1
Message Receive Count    : 30155
Message Receive Bytes    : 1326820

-------------------------------------------

Session Slot             : 3
Socket FD                : 86
Client ID                : 0
Message Receive Count    : 0
Message Receive Bytes    : 0

-------------------------------------------

Session Slot             : 3
Socket FD                : 86
Client ID                : 1
Message Receive Count    : 29611
Message Receive Bytes    : 69782901
```

Configuring Example for SNMP Background Synchronization

The following example shows how to configure SNMP background synchronization:

```
enable
configure terminal
cable bgsync active
service internal
cable bgsync itime 200
cable bgsync ptime 500
exit
```

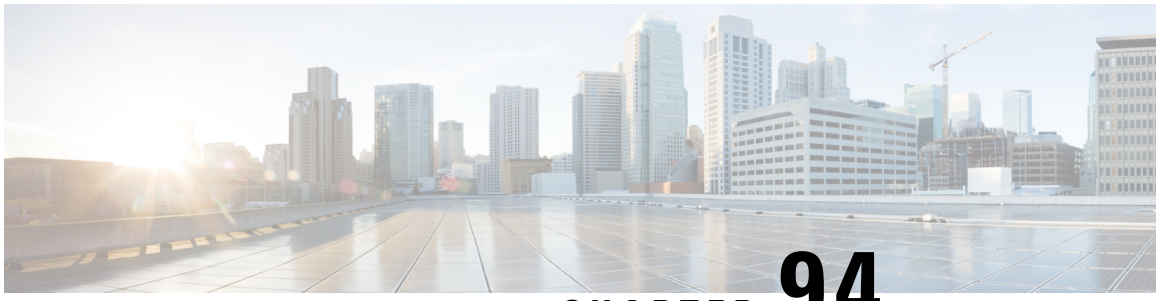
Reference

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 242: Feature Information for SNMP Background Synchronization

Feature Name	Releases	Feature Information
SNMP Background Synchronization	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Router s.



CHAPTER 94

Online Offline Diagnostics

Online Offline Diagnostics (OOD) Field Diagnostics feature allows the customer to test and verify hardware-related issues on a line card deployed in the field. The test results can be used to verify whether a line card is fault and troubleshoot network issues.

- Overview of Online Offline Diagnostics, on page 1461
- How to Configure Online Offline Diagnostics, on page 1462
- Configuration Example for Online Offline Diagnostics, on page 1463
- Feature Information for Online Offline Diagnostics, on page 1463

Overview of Online Offline Diagnostics

The Online Offline Diagnostics is a field diagnostic mechanism that allows the customers to test and verify the line card hardware problems.

To perform a hardware diagnostic test on a line card in the Cisco cBR universal broadband router, download an OOD Field Diagnostic image for free from Cisco.com and use it to verify if the line card problems are due to hardware failure. The customer can run field diagnostic tests on the standby line card at any time without interrupting service. Testing the standby line card improves high availability of the system by ensuring the standby line card is ready for a switchover.

Field Diagnostic Image Information

Field Diagnostic image is used to run diagnostic tests on a line card and is available from Cisco.com.

First, download it from Cisco.com to one of the flash file systems on the router. Then move it to the line card, and the line card is automatically taken offline. Once field diagnostic tests are complete and the test results are gathered, the Field Diagnostic image must be unloaded from the line card. Normal line card operation will automatically resume after the Field Diagnostic image is unloaded from the line card.

Benefits of Online Offline Diagnostics

- **Improved Troubleshooting**. Field diagnostics verifies whether a line card problem is hardware-related or not. If the problem is software-related, the Field Diagnostic image allows customer to quickly rule out hardware related cause and focus on fixing the software issue causing the problem.
- **Pre-installation Line Card Hardware Verification**. Field diagnostics verifies whether a line card has hardware problems before installing the line card in a Cisco cBR Series router.

- **Onsite Fault Detection**. Field diagnostics helps to confirm if the problem is hardware-related and if it is necessary to replace the line card.
- **Additional Uptime**. Field diagnostics ensures that line cards are not mistakenly taken offline if the problem is not hardware-related, thereby increasing network uptime.

Prerequisites for Online Offline Diagnostics

- Before running the OOD Field Diagnostic tests on the working (active) line card in an N + 1 redundancy setup, it is advisable to switch over to the protect (standby) line card before loading the Field Diagnostic image to the line card to avoid service interruption.
- After an OOD Field Diagnostic image is loaded to the line card, the line card goes offline. Therefore, schedule a downtime for the line card to be tested before performing field diagnostic tests.
- Before performing any field diagnostic test, unplug all cables on the device that connect to other interfaces. If the cables that connect interfaces are not unplugged, some field diagnostic tests may send packets to connected devices, which increments packet counters on the receiving interfaces.

Restrictions for Online Offline Diagnostics

- When accessing a router through Telnet while running an OOD Field Diagnostic test, testing progress messages do not appear on the screen.
- If supervisor switchover occurs during a field diagnostic test, the test stops immediately and the line card run-time image automatically replaces the Field Diagnostic image on the line card.
- This feature is supported on CBR-CCAP-LC-40G line card in Cisco IOS-XE release 3.18.0S and later releases.
- It is suggested to run OOD on one line card at a time to avoid service impact.
- To run OOD on multiple line cards, leave 5 to 10 minutes gap before loading the OOD image to the next line card.

How to Configure Online Offline Diagnostics

Configuring Field Diagnostic Test

To load the field diagnostic image and start field diagnostic test, complete the following procedure:

```
copy tftp:image-file {harddisk: | bootflash: | flash:}
request platform hardware diagnostic load slot slot-id image-file autostart
```

Verifying the Testing Process

To verify whether the field diagnostic tests are running, use the **show platform hardware diagnostic status slot** *slot-id* command as shown in the example below:

```
Router# show platform hardware diagnostic status slot 0
Online Offline Diagnostic Status (P=Passed, F=Failed, U=Untested)
State                 Overall Test Num        Test Done Num          Test Result

--------------------------------------------------------------------------------
Running Auto Test     75                      70                     P:69 F:1 U:5
```

Note If the test result shows that the failed test number is not 0, please copy the full log and contact Cisco TAC team for support. You can use **dir harddisk:ood/** command to list the log files.

Removing the Field Diagnostic Image from a Line Card

To unload the Field Diagnostic image, use the **request platform hardware diagnostic unload slot** *slot-id* command as below:

request platform hardware diagnostic unload slot *slot-id*

Then the line card will be reloaded to run-time image.

Note To retain the results of a diagnostic test, copy and paste the **show platform hardware diagnostic status slot** command output into a separate file before unloading the Field Diagnostic image. The output of the **show platform hardware diagnostic status slot** command cannot be gathered after unloading the Field Diagnostic image from the line card.

Configuration Example for Online Offline Diagnostics

The following example shows running output for Online Offline Diagnostics:

```
copy tftp:field_diag harddisk:
request platform hardware diagnostic load slot 0 harddisk:field_diag autostart

Mar 2 16:00:51.933 CST: %IOSXE_OIR-6-REMCARD: Card (cc) removed from slot x
Mar 2 16:00:51.934 CST: %CABLE_CLC-5-LOGGER_LC_REMOVED: Carrier Card x removed
```

Feature Information for Online Offline Diagnostics

Use Cisco Feature Navigator to find information about the platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to the www.cisco.com/go/cfn link. An account on the Cisco.com page is not required.

Note The following table lists the software release in which a given feature is introduced. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 243: Feature Information for Online Offline Diagnostics

Feature Name	Releases	Feature Information
Online Offline Diagnostics	Cisco IOS-XE Release 16.5.1	This feature was integrated into Cisco IOS-XE Release 16.5.1 on the Cisco cBR Series Converged Broadband Router s.