MPLS Layer 3 VPNs Configuration Guide, Cisco IOS Release 12.4T

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

CONTENTS

Configuring MPLS Layer 3 VPNs 1

- Finding Feature Information 1
- Prerequisites for MPLS Layer 3 VPNs 1
- Restrictions for MPLS Layer 3 VPNs 2
- Information About MPLS Layer 3 VPNs 3
  - MPLS VPN Definition 4
  - How an MPLS VPN Works 5
    - How Virtual Routing and Forwarding Tables Work in an MPLS VPN 5
    - How VPN Routing Information Is Distributed in an MPLS VPN 5
    - BGP Distribution of VPN Routing Information 6
    - MPLS Forwarding 6
  - Major Components of MPLS VPNs 6
  - Benefits of an MPLS VPN 7
- How to Configure MPLS Layer 3 VPNs 9
  - Configuring the Core Network 9
    - Assessing the Needs of MPLS VPN Customers 9
    - Configuring Routing Protocols in the Core 10
    - Configuring MPLS in the Core 10
    - Configuring Multiprotocol BGP on the PE Routers and Route Reflectors 10
      - Troubleshooting Tips 12
  - Connecting the MPLS VPN Customers 12
    - Defining VRFs on the PE Routers to Enable Customer Connectivity 12
    - Configuring VRF Interfaces on PE Routers for Each VPN Customer 14
    - Configuring Routing Protocols Between the PE and CE Routers 15
      - Configuring BGP as the Routing Protocol Between the PE and CE Routers 15
      - Configuring RIPv2 as the Routing Protocol Between the PE and CE Routers 17
      - Configuring Static Routes Between the PE and CE Routers 19
      - Configuring OSPF as the Routing Protocol Between the PE and CE Routers 21
      - Configuring EIGRP as the Routing Protocol Between the PE and CE Routers 23

Configuring EIGRP Redistribution in the MPLS VPN 26
Verifying the VPN Configuration 28
Verifying Connectivity Between MPLS VPN Sites 29
Verifying IP Connectivity from CE Router to CE Router Across the MPLS Core 29
Verifying that the Local and Remote CE Routers Are in the Routing Table 30
Configuration Examples for MPLS VPNs 30
Configuring an MPLS VPN Using BGP Example 30
Configuring an MPLS VPN Using RIP Example 31
Configuring an MPLS VPN Using Static Routes Example 32
Configuring an MPLS VPN Using OSPF Example 33
Configuring an MPLS VPN Using EIGRP Example 34
Additional References 35
Feature Information for MPLS Layer 3 VPNs 37
Configuring Route Maps to Control the Distribution of MPLS Labels Between Routers in an MPLS VPN 39
Finding Feature Information 39
Restrictions for Using Route Maps with MPLS VPNs 39
Prerequisites for Using Route Maps with MPLS VPNs 39
Information About Route Maps in MPLS VPNs 40
How to Configure Route Maps in an MPLS VPN 40
Configuring a Route Map for Incoming Routes 40
Configuring a Route Map for Outgoing Routes 42
Applying the Route Maps to the MPLS VPN Edge Routers 44
Troubleshooting Tips 46
Configuration Examples for Route Maps in MPLS VPNs 46
Using a Route Map in an MPLS VPN Inter-AS Network Example 46
Using a Route Map in an MPLS VPN CSC Network Example 47
Additional References 48
Feature Information for Route Maps in MPLS VPNs 50
Dialing to Destinations with the Same IP Address for MPLS VPNs 53
Finding Feature Information 53
Prerequisites for Dialing to Destinations with the Same IP Address for MPLS VPNs 53
Restrictions for Dialing to Destinations with the Same IP Address for MPLS VPNs 54
Information About Dialing to Destinations with the Same IP Address for MPLS VPNs 55
Introduction to Dialing to Destinations with the Same IP Address for MPLS VPNs 56

Benefits of this Feature 56
How to Enable Dialing to Destinations with the Same IP Address for MPLS VPNs 56
Mapping the VRF and Next-Hop Address to a Dial String 56
Verifying the Configuration 58
Troubleshooting Tips 58
Configuration Examples for Dialing to Destinations with the Same IP Address 59
Additional References 63
Feature Information for Dialing to Destinations with the Same IP Address 65
Configuring Scalable Hub-and-Spoke MPLS VPNs 67
Finding Feature Information 67
Prerequisites for Configuring Scalable Hub-and-Spoke MPLS VPNs 67
Restrictions for Configuring Scalable Hub-and-Spoke MPLS VPNs 68
Information about Configuring Scalable Hub-and-Spoke MPLS VPNs 68
Overview 68
Upstream and Downstream VRFs 69
Reverse Path Forwarding Check 69
How to Ensure that MPLS VPN Clients Use the Hub PE Router 69
Configuring the Upstream and Downstream VRFs on the PE Router or the Spoke PE Router 69
Associating VRFs 71
Configuring the Downstream VRF for an AAA Server 72
Verifying the Configuration 72
Configuration Examples for Configuring Scalable Hub-and-Spoke MPLS VPNs 75
Configuring the Upstream and Downstream VRFs on the PE Router and the Spoke PE Router Example 75
Associating VRFs Example 75
Configuring Scalable Hub-and-Spoke MPLS VPNs--Basic Configuration Example 76
Example 77
Additional References 78
Feature Information for Configuring Scalable Hub-and-Spoke MPLS VPNs 79
Ensuring MPLS VPN Clients Communicate over the Backbone Links 81
Finding Feature Information 81
Prerequisites for Ensuring MPLS VPN Clients Communicate over the Backbone Links 81
Restrictions for Ensuring MPLS VPN Clients Communicate over the Backbone Links 82
Information About Ensuring MPLS VPN Clients Communicate over the Backbone Links 82
Introduction to MPLS VPNs Using OSPF Between PE and CE Routers 82

OSPF Uses Backdoor Paths to Communicate Between VPN Sites 83
Sham-Links Direct Traffic Between VPN Sites over the MPLS VPN Backbone 84
How to Ensure That MPLS VPN Clients Communicate over the MPLS VPN Backbone 85
Configuration Examples for Ensuring MPLS VPN Clients Communicate over the MPLS VPN Backbone 87
Additional References 90
Feature Information for Ensuring MPLS VPN Clients Communicate over the MPLS VPN Backbone 91
Assigning an ID Number to a VPN 93
Finding Feature Information 93
Information About VPN ID 93
Introduction to VPN ID 93
Components of the VPN ID 94
Management Applications That Use VPN IDs 94
Dynamic Host Configuration Protocol 94
Remote Authentication Dial-In User Service 94
How to Configure a VPN ID 95
Specifying a VPN ID 95
Restrictions 95
Verifying the VPN ID Configuration 96
Additional References 97
Feature Information for Assigning an ID Number to a VPN 99
Directing MPLS VPN Traffic Using Policy-Based Routing 101
Finding Feature Information 101
Prerequisites for Directing MPLS VPN Traffic Using Policy-Based Routing 101
Restrictions for Directing MPLS VPN Traffic Using Policy-Based Routing 102
Information About Directing MPLS VPN Traffic Using Policy-Based Routing 102
Directing MPLS VPN Traffic Using Policy-Based Routing Overview 102
VRF Selection Introduces a New PBR Set Clause 103
How to Configure Policy-Based Routing To Direct MPLS VPN Traffic 103
Defining the Match Criteria 103
Prerequisites 104
Defining Match Criteria with a Standard Access List 104
Defining Match Criteria with an Extended Access List 104
Configuring the Route Map and Specifying VRFs 106
Applying a Route Map to an Interface 107

Configuring IP VRF Receive on the Interface 109
Verifying the Configuration 110
Configuration Examples for Directing MPLS VPN Traffic Using Policy-Based Routing 111
Configuring Policy-Based Routing with a Standard Access List Example 111
Verifying Policy-Based Routing Example 111
Additional References 112
Feature Information for Directing MPLS VPN Traffic Using Policy-Based Routing 114
Directing MPLS VPN Traffic Using a Source IP Address 117
Finding Feature Information 117
Prerequisites for Directing MPLS VPN Traffic Using a Source IP Address 117
Restrictions for Directing MPLS VPN Traffic Using a Source IP Address 118
Information About Directing MPLS VPN Traffic Using a Source IP Address 120
Introduction to Directing MPLS VPN Traffic Using a Source IP Address 120
How MPLS VPN Traffic Is Routed Using the Source IP Address 120
Example of MPLS VPN Traffic Being Routed Based on the Source IP Address 121
MPLS VPN Traffic Is Unidirectional 122
Conditions That Cause MPLS VPN Traffic To Become Bidirectional 123
Advantages of Using the Source IP Address over Per-Interface IP VPN Configuration 123
Benefits of Directing MPLS VPN Traffic Using a Source IP Address 124
How to Enable MPLS VPN Traffic To Be Routed Using a Source IP Address 124
Enabling Routing of MPLS VPN Traffic Based on the Source IP Address 124
Establishing IP Static Routes for a VRF Instance 126
Troubleshooting Tips 127
Configuration Examples for Directing MPLS VPN Traffic Using a Source IP Address 128
Enabling MPLS VPN Traffic To Be Routed Based on Source IP Address Example 128
Configuring a VRF to Eliminate Unnecessary Packet Forwarding Example 129
Verifying the Configuration Example 129
Additional References 129
Feature Information for Directing MPLS VPN Traffic Using a Source IP Address 131
MPLS VPN--Show Running VRF 133
Finding Feature Information 133
Prerequisites for MPLS VPN--Show Running VRF 133
Restrictions for MPLS VPN--Show Running VRF 134
Information About MPLS VPN--Show Running VRF 134
Configuration Elements Displayed for the MPLS VPN--Show Running VRF Feature 134

Display of VRF Routing Protocol Configuration 134
Display of Configuration Not Directly Linked to a VRF 135
How to Configure MPLS VPN--Show Running VRF 135
Configuration Examples for MPLS VPN--Show Running VRF 136
Additional References 136
Feature Information for MPLS VPN--Show Running VRF 137
Glossary 138
MPLS VPN Half-Duplex VRF 141
Finding Feature Information 141
Prerequisites for Configuring MPLS VPN Half-Duplex VRF 141
Restrictions for MPLS VPN Half-Duplex VRF 141
Information About Configuring MPLS VPN Half-Duplex VRF 142
MPLS VPN Half-Duplex VRF Overview 142
Upstream and Downstream VRFs 142
Reverse Path Forwarding Check 143
How to Configure MPLS VPN Half-Duplex VRF 143
Configuring the Upstream and Downstream VRFs on the Spoke PE Router 144
Associating a VRF with an Interface 145
Configuring the Downstream VRF for an AAA Server 146
Verifying MPLS VPN Half-Duplex VRF Configuration 147
Configuration Examples for MPLS VPN Half-Duplex VRF 150
Example Configuring the Upstream and Downstream VRFs on the Spoke PE Router 150
Example Associating a VRF with an Interface 151
Example Configuring MPLS VPN Half-Duplex VRF Using Static CE-PE Routing 151
Example Configuring MPLS VPN Half-Duplex VRF Using RADIUS Server and Static CE-PE Routing 152
Example Configuring MPLS VPN Half-Duplex VRF Using Dynamic CE-PE Routing 153
Additional References 155
Feature Information for MPLS VPN Half-Duplex VRF 156
MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs 159
Finding Feature Information 159
Prerequisites for MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs 159
Restrictions for MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs 160
Information About MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs 160
VRF Concepts Similar for IPv4 and IPv6 MPLS VPNs 160

Single-Protocol VRF to Multiprotocol VRF Migration 160
Multiprotocol VRF Configurations Characteristics 161
How to Configure MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs 162
Configuring a VRF for IPv4 and IPv6 MPLS VPNs 162
Associating a Multiprotocol VRF with an Interface 164
Verifying the MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs Configuration 166
Migrating from a Single-Protocol IPv4-Only VRF to a Multiprotocol VRF Configuration 169
Configuration Examples for MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs 170
Example Multiprotocol VRF Configuration Single Protocol with Noncommon Policies 171
Example Multiprotocol VRF Configuration Multiprotocol with Noncommon Policies 171
Example Multiprotocol VRF Configuration Multiprotocol with Common Policies 171
Example Multiprotocol VRF Configuration Multiprotocol with Common and Noncommon Policies 172
Example Configuring a VRF for IPv4 and IPv6 VPNs 172
Example Associating a Multiprotocol VRF with an Interface 173
Example Migrating from a Single-Protocol IPv4-Only VRF Configuration to a Multiprotocol VRF Configuration 173
Additional References 174
Feature Information for MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs 175
Glossary 176
MPLS VPN--Route Target Rewrite 179
Finding Feature Information 179
Prerequisites for MPLS VPN--Route Target Rewrite 179
Restrictions for MPLS VPN--Route Target Rewrite 180
Information About MPLS VPN--Route Target Rewrite 180
Route Target Replacement Policy 180
Route Maps and Route Target Replacement 181
How to Configure MPLS VPN--Route Target Rewrite 181
Configuring a Route Target Replacement Policy 182
Applying the Route Target Replacement Policy 185
Associating Route Maps with Specific BGP Neighbors 185
Refreshing BGP Session to Apply Route Target Replacement Policy 187
Troubleshooting Tips 188
Verifying the Route Target Replacement Policy 189
Troubleshooting Your Route Target Replacement Policy 190

Configuration Examples for MPLS VPN--Route Target Rewrite 192
Configuring Route Target Replacement Policies Examples 192
Applying Route Target Replacement Policies Examples 193
Associating Route Maps with Specific BGP Neighbor Example 193
Refreshing the BGP Session to Apply the Route Target Replacement Policy Example 194
Additional References 194
Feature Information for MPLS VPN--Route Target Rewrite 195
Glossary 196
MPLS VPN VRF Selection Using Policy-Based Routing 199
Finding Feature Information 199
Prerequisites for VRF Selection Using Policy-Based Routing 199
Restrictions for VRF Selection Using Policy-Based Routing 200
Information About VRF Selection Using Policy-Based Routing 200
Introduction to VRF Selection Using Policy-Based Routing 200
Policy-Based Routing Set Clauses Overview 200
How to Configure VRF Selection Using Policy-Based Routing 201
Defining the Match Criteria for PBR VRF Selection Based on Packet Length 201
Prerequisites 201
Configuring PBR VRF Selection with a Standard Access List 201
Configuring PBR VRF Selection with a Named Access List 202
Configuring PBR VRF Selection in a Route Map 203
Configuring PBR on the Interface 205
Configuring IP VRF Receive on the Interface 206
Verifying the Configuration of the VRF Selection Using Policy-Based Routing 208
Configuration Examples for VRF Selection Using Policy-Based Routing 209
Example Defining PBR VRF Selection in Access List 209
Example Verifying VRF Selection Using Policy-Based Routing 209
Verifying Match Criteria 210
Verifying Route-Map Configuration 210
Verifying PBR VRF Selection Policy 210
Additional References 210
Feature Information for VRF Selection Using Policy-Based Routing 212
Glossary 212
MPLS VPN - Interautonomous System Support 215
Finding Feature Information 216

Prerequisites for MPLS VPN - Interautonomous System Support 216
Restrictions for MPLS VPN - Interautonomous System Support 217
Information About MPLS VPN - Interautonomous System Support 217
MPLS VPN Interautonomous System Benefits 218
Interautonomous System Communication with ASBRs 218
Interautonomous System Configurations Supported in an MPLS VPN 218
How Information Is Exchanged in an MPLS VPN Inter-AS with ASBRs 219
Information Sent in an MPLS VPN Inter-AS with ASBRs 219
VPN Routing Information Exchange in an MPLS VPN Inter-AS with ASBRs 220
Packet Forwarding Between MPLS VPN Interautonomous Systems with ASBRs 222
Confederation Configuration for MPLS VPN Inter-AS with ASBRs 224
Load Sharing with MPLS VPN Inter-AS ASBRs 225
How to Configure MPLS VPN - Interautonomous System Support 227
Configuring an eBGP ASBR to Exchange MPLS VPN-IPv4 Addresses 227
Configuring Peering with Directly Connected Interfaces Between ASBRs 227
Configuring Peering of the Loopback Interface of Directly Connected ASBRs 229
Configuring Loopback Interface Addresses for Directly Connected ASBRs 230
Examples 231
Configuring Static Routes to the eBGP Neighbor Loopback 231
Examples 233
Configuring Forwarding on the Directly Connected Interfaces 233
Examples 234
Configuring an eBGP Session Between the Loopbacks 235
Examples 238
Configuring eBGP Routing to Exchange MPLS VPN Routes Between Subautonomous Systems in a Confederation 238
Verifying Inter-AS for ASBRs Exchanging MPLS VPN-IPv4 Addresses 241
Configuring eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs 243
Examples 247
Verifying eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs 248
Configuration Examples for MPLS VPN - Interautonomous System Support 250
Configuring Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses Example 250
Configuration for Autonomous System 1 CE1 Example for Two Autonomous Systems 251
Configuration for Autonomous System 1 PE1 Example for Two Autonomous Systems 251
Configuration for Autonomous System 1 P1 Example for Two Autonomous Systems 252

Configuration for Autonomous System 1 ASBR1 Example for Two Autonomous Systems 253
Configuration for Autonomous System 2 ASBR2 Example for Two Autonomous Systems 253
Configuration for Autonomous System 2 P2 Example for Two Autonomous Systems 254
Configuration for Autonomous System 2 PE2 Example for Two Autonomous Systems 255
Configuration for Autonomous System 2 CE2 Example for Two Autonomous Systems 256
Configuring Inter-AS with ASBRs in a Confederation Example 256
Inter-AS Confederation Configuration for Autonomous System 1 CE1 Example 257
Inter-AS Confederation Configuration for Autonomous System 1 PE1 Example 257
Inter-AS Confederation Configuration for Autonomous System 1 P1 Example 258
Inter-AS Confederation Configuration for Autonomous System 1 ASBR1 Example 259
Inter-AS Confederation Configuration for Autonomous System 2 ASBR2 Example 259
Inter-AS Confederation Configuration for Autonomous System 2 P2 Example 260
Inter-AS Confederation Configuration for Autonomous System 2 PE2 Example 261
Inter-AS Confederation Configuration for Autonomous System 2 CE2 Example 262
Configuring eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs Example 262
Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 CE1 Example 263
Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 PE1 Example 264
Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 P1 Example 265
Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 ASBR1 Example 265
Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 ASBR2 Example 266
Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 ASBR3 Example 267
Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 P2 Example 268
Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 PE2 Example 268
Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 CE2 Example 269
Additional References 270

Feature Information for MPLS VPN - Interautonomous System Support 271
Glossary 274
MPLS VPN--SNMP Notifications 277
Finding Feature Information 278
Prerequisites for MPLS VPN--SNMP Notifications 278
Restrictions for MPLS VPN--SNMP Notifications 278
Information About MPLS VPN--SNMP Notifications 278
Cisco Implementation of MPLS-VPN-MIB 279
Capabilities Supported by MPLS VPN--SNMP Notifications 279
Notification Generation Events for the MPLS-VPN-MIB 279
Notification Specification for MPLS-VPN-MIB 281
Monitoring the MPLS VPN--SNMP Notifications 282
How to Configure the MPLS VPN--SNMP Notifications 282
Configuring an SNMP Communtity 282
Configure the Router to Send SNMP Traps 283
Configure Threshold Values for MPLS VPN--SNMP Notifications 286
Configuration Examples for MPLS VPN--SNMP Notifications 287
Configure the Community Example 288
Configure the Router to Send SNMP Traps Examples 288
Configure Threshold Values for Examples 288
Additional References 288
Command Reference 289
Glossary 290
Multi-VRF Selection Using Policy-Based Routing (PBR) 293
Finding Feature Information 293
Prerequisites for Multi-VRF Selection Using Policy-Based Routing 294
Restrictions for Multi-VRF Selection Using Policy-Based Routing 294
Information About Multi-VRF Selection Using Policy-Based Routing 294
Policy Routing of VPN Traffic Based on Match Criteria 294
Policy-Based Routing set Commands 295
Policy-routing Packets for VRF Instances 295
Change of Normal Routing and Forwarding Behavior 296
Support of Inherit-VRF Inter-VRF and VRF-to-Global Routing 296
How to Configure Multi-VRF Selection Using Policy-Based Routing 297
Defining the Match Criteria for Multi-VRF Selection Using PBR 297

Configuring Multi-VRF Selection Using PBR with a Standard Access List 297
Configuring Multi-VRF Selection Using PBR with a Named Extended Access List 298
Configuring Multi-VRF Selection in a Route Map 299
Configuring Multi-VRF Selection Using PBR and IP VRF Receive on the Interface 302
Verifying the Configuration of Multi-VRF Selection Using PBR 303
Configuration Examples for Multi-VRF Selection Using Policy-Based Routing 305
Defining the Match Criteria for Multi-VRF Selection Using PBR Example 305
Configuring Multi-VRF Selection in a Route Map Example 306
Additional References 306
Feature Information for Multi-VRF Selection Using Policy-Based Routing 307
Glossary 308

Configuring MPLS Layer 3 VPNs

A Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) consists of a set of sites that are interconnected by means of an MPLS provider core network. At each customer site, one or more customer edge (CE) routers attach to one or more provider edge (PE) routers. This module explains how to create an MPLS VPN.

- Finding Feature Information, page 1
- Prerequisites for MPLS Layer 3 VPNs, page 1
- Restrictions for MPLS Layer 3 VPNs, page 2
- Information About MPLS Layer 3 VPNs, page 3
- How to Configure MPLS Layer 3 VPNs, page 9
- Configuration Examples for MPLS VPNs, page 30
- Additional References, page 35
- Feature Information for MPLS Layer 3 VPNs, page 37

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for MPLS Layer 3 VPNs

Before configuring MPLS Layer 3 VPNs, you should have MPLS, Label Distribution Protocol (LDP), and Cisco Express Forwarding installed in your network. All routers in the core, including the PE routers, must be able to support Cisco Express Forwarding and MPLS forwarding. See the Assessing the Needs of MPLS VPN Customers, page 9 for more information.

Cisco Express Forwarding must be enabled all routers in the core, including the PE routers. For information about how to determine if Cisco Express Forwarding is enabled, see Configuring Basic Cisco Express Forwarding--Improving Performance, Scalability, and Resiliency in Dynamic Network .

Restrictions for MPLS Layer 3 VPNs

When configuring static routes in an MPLS or MPLS VPN environment, some variations of the **ip route** and **ip route vrf** commands are not supported. These variations of the commands are not supported in Cisco IOS releases that support the Tag Forwarding Information Base (TFIB), specifically Cisco IOS Releases 12.xT, 12.xM, and 12.0S. The TFIB cannot resolve prefixes when the recursive route over which the prefixes travel disappears and then reappears. However, the command variations are supported in Cisco IOS releases that support the MPLS Forwarding Infrastructure (MFI), specifically Cisco IOS Release 12.2(25)S and later. Use the following guidelines when configuring static routes.

Supported Static Routes in an MPLS Environment

The following **ip route** command is supported when you configure static routes in MPLS environment:

ip route *destination-prefix mask interface next-hop-address*

The following **ip route** commands are supported when you configure static routes in an MPLS environment and configure load sharing with static nonrecursive routes and a specific outbound interface:

ip route *destination-prefix mask* **interface1 next-hop1**

ip route *destination-prefix mask* **interface2 next-hop2**

Unsupported Static Routes in an MPLS Environment that Uses the TFIB

The following **ip route** command is not supported when you configure static routes in an MPLS environment:

ip route *destination-prefix mask next-hop-address*

The following **ip route** command is not supported when you configure static routes in an MPLS environment and enable load sharing where the next hop can be reached through two paths:

ip route *destination-prefix mask next-hop-address*

The following **ip route** command is not supported when you configure static routes in an MPLS environment and enable load sharing where the destination can be reached through two next hops:

ip route *destination-prefix mask* **next-hop1**

ip route *destination-prefix mask* **next-hop2**

Use the *interface* an *next-hop* arguments when specifying static routes.

Supported Static Routes in an MPLS VPN Environment

The following **ip route vrf** commands are supported when you configure static routes in a MPLS VPN environment, and the next hop and interface are in the same VRF:

- ◦ **ip route vrf** *vrf-name destination-prefix mask next-hop-address*
  ◦ **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address*
  ◦ **ip route vrf** *vrf-name destination-prefix mask* **interface1 next-hop1**
  ◦ **ip route vrf** *vrf-name destination-prefix mask* **interface2 next-hop2**

The following **ip route vrf** commands are supported when you configure static routes in a MPLS VPN environment, and the next hop is in the global table in the MPLS cloud in the global routing table. For example, these commands are supported when the next hop is pointing to the Internet Gateway.

- ◦ **ip route vrf** *vrf-name destination-prefix mask next-hop-address* **global**

- **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address* (This command is supported when the next hop and interface are in the core.)

The following **ip route** commands are supported when you configure static routes in a MPLS VPN environment and enable load sharing with static nonrecursive routes and a specific outbound interfaces:

ip route *destination-prefix mask* **interface1 next-hop1**

ip route *destination-prefix mask* **interface2 next-hop2**

Unsupported Static Routes in an MPLS VPN Environment that Uses the TFIB

The following **ip route** command is not supported when you configure static routes in a MPLS VPN environment, the next hop is in the global table in the MPLS cloud within the core, and you enable load sharing where the next hop can be reached through two paths:

ip route vrf *destination-prefix mask next-hop-address* **global**

The following **ip route** commands are not supported when you configure static routes in a MPLS VPN environment, the next hop is in the global table in the MPLS cloud within the core, and you enable load sharing where the destination can be reached through two next hops:

ip route vrf *destination-prefix mask* **next-hop1 global**

ip route vrf *destination-prefix mask* **next-hop2 global**

The following **ip route vrf** commands are not supported when you configure static routes in an MPLS VPN environment, and the next hop and interface are in the same VRF:

ip route vrf *vrf-name destination-prefix mask* **next-hop1** *vrf-name destination-prefix mask* **next-hop1**

ip route vrf *vrf-name destination-prefix mask* **next-hop2**

Supported Static Routes in an MPLS VPN Environment Where the Next Hop Resides in the Global Table on the CE Router

The following **ip route vrf** command is supported when you configure static routes in a MPLS VPN environment, and the next hop is in the global table on the CE side. For example, the following command is supported when the destination-prefix is the CE router's loopback address, as in EBGP multihop cases.

ip route vrf *vrf-name destination-prefix mask interface next-hop-address*

The following **ip route** commands are supported when you configure static routes in a MPLS VPN environment, the next hop is in the global table on the CE side, and you enable load sharing with static non-recursive routes and a specific outbound interfaces:

ip route *destination-prefix mask* **interface1 nexthop1**

ip route *destination-prefix mask* **interface2 nexthop2**

Information About MPLS Layer 3 VPNs

- MPLS VPN Definition, page 4
- How an MPLS VPN Works, page 5
- Major Components of MPLS VPNs, page 6
- Benefits of an MPLS VPN, page 7

MPLS VPN Definition

Before defining an MPLS VPN, you need to define a VPN in general. A VPN is:

- An IP-based network delivering private network services over a public infrastructure
- A set of sites that are allowed to communicate with each other privately over the Internet or other public or private networks

Conventional VPNs are created by configuring a full mesh of tunnels or permanent virtual circuits (PVCs) to all sites in a VPN. This type of VPN is not easy to maintain or expand, because adding a new site requires changing each edge device in the VPN.

MPLS-based VPNs are created in Layer 3 and are based on the peer model. The peer model enables the service provider and the customer to exchange Layer 3 routing information. The service provider relays the data between the customer sites without the customer's involvement.

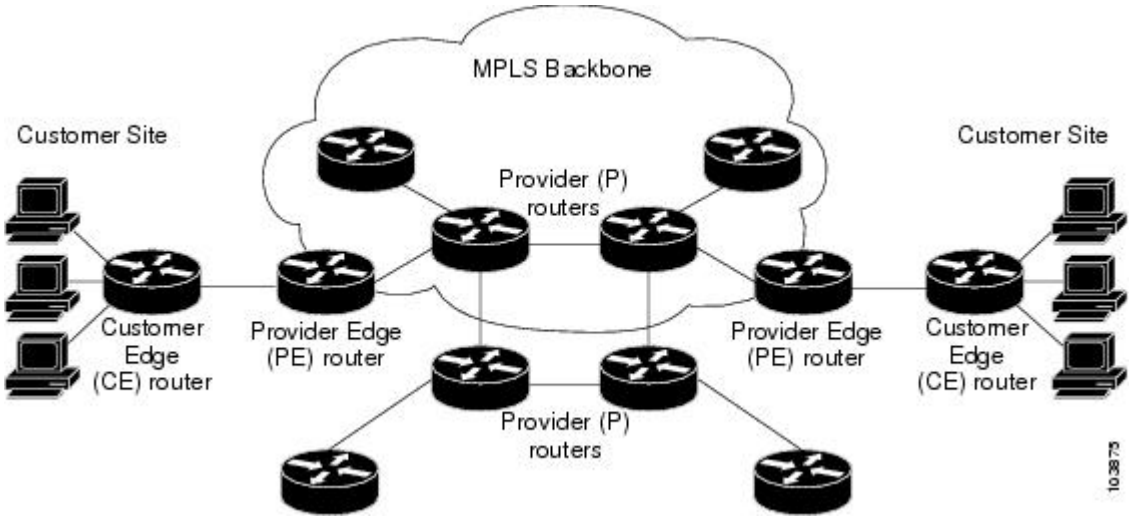
MPLS VPNs are easier to manage and expand than conventional VPNs. When a new site is added to an MPLS VPN, only the service provider’s edge router that provides services to the customer site needs to be updated.

The different parts of the MPLS VPN are described as follows:

- Provider (P) router--Router 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.
- PE router--Router that attaches 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.
- Customer (C) router--Router in the ISP or enterprise network.
- Customer edge router--Edge router on the network of the ISP that connects to the PE router on the network. A CE router must interface with a PE router.

The figure below shows a basic MPLS VPN.

Figure 1 Basic MPLS VPN Terminology



How an MPLS VPN Works

MPLS VPN functionality is enabled at the edge of an MPLS network. The PE router performs the following:

- Exchanges routing updates with the CE router
- Translates the CE routing information into VPNv4 routes
- Exchanges VPNv4 routes with other PE routers through the Multiprotocol Border Gateway Protocol (MP-BGP)

- How Virtual Routing and Forwarding Tables Work in an MPLS VPN, page 5
- How VPN Routing Information Is Distributed in an MPLS VPN, page 5
- BGP Distribution of VPN Routing Information, page 6
- MPLS Forwarding, page 6

How Virtual Routing and Forwarding Tables Work in an MPLS VPN

Each VPN is associated with one or more virtual routing and forwarding (VRF) instances. A VRF defines the VPN membership of a customer site attached to a PE router. A VRF consists of the following components:

- An IP routing table
- A derived Cisco Express Forwarding table
- A set of interfaces that use the forwarding table
- A set of rules and routing protocol parameters that control the information that is included in the routing table

A one-to-one relationship does not necessarily exist between customer sites and VPNs. A site can be a member of multiple VPNs. However, a site can associate with only one VRF. A site's VRF contains all the routes available to the site from the VPNs of which it is a member.

Packet forwarding information is stored in the IP routing table and the Cisco Express Forwarding table for each VRF. A separate set of routing and Cisco Express Forwarding tables is maintained for each VRF. These tables prevent information from being forwarded outside a VPN, and also prevent packets that are outside a VPN from being forwarded to a router within the VPN.

How VPN Routing Information Is Distributed in an MPLS VPN

The distribution of VPN routing information is controlled through the use of VPN route target communities, implemented by BGP extended communities. VPN routing information is distributed as follows:

- When a VPN route that is learned from a CE router is injected into BGP, a list of VPN route target extended community attributes is associated with it. Typically the list of route target community extended values is set from an export list of route targets associated with the VRF from which the route was learned.
- An import list of route target extended communities is associated with each VRF. The import list defines route target extended community attributes that a route must have in order for the route to be imported into the VRF. For example, if the import list for a particular VRF includes route target extended communities A, B, and C, then any VPN route that carries any of those route target extended communities--A, B, *or* C--is imported into the VRF.

BGP Distribution of VPN Routing Information

A PE router can learn an IP prefix from the following sources:

- A CE router by static configuration
- A BGP session with the CE router
- A Routing Information Protocol (RIP) exchange with the CE router

The IP prefix is a member of the IPv4 address family. After the PE router learns the IP prefix, the PE converts it into a VPN-IPv4 prefix by combining it with an 8-byte route distinguisher (RD). The generated prefix is a member of the VPN-IPv4 address family. It uniquely identifies the customer address, even if the customer site is using globally nonunique (unregistered private) IP addresses. The route distinguisher used to generate the VPN-IPv4 prefix is specified by a configuration command associated with the VRF on the PE router.

BGP distributes reachability information for VPN-IPv4 prefixes for each VPN. BGP communication takes place at two levels:

- Within IP domains, known as an autonomous system (interior BGP [IBGP])
- Between autonomous systems (external BGP [EBGP])

PE-PE or PE-RR (route reflector) sessions are IBGP sessions, and PE-CE sessions are EBGP sessions. In an EIGRP PE-CE environment, when an EIGRP internal route is redistributed into BGP by one PE, then back into EIGRP by another PE, the originating router-id for the route is set to the router-id of the second PE, replacing the original internal router-id.

BGP propagates reachability information for VPN-IPv4 prefixes among PE routers by means of the BGP multiprotocol extensions (refer to RFC 2283, *Multiprotocol Extensions for BGP-4*), which define support for address families other than IPv4. Using the extensions ensures that the routes for a given VPN are learned only by other members of that VPN, enabling members of the VPN to communicate with each other.

MPLS Forwarding

Based on routing information stored in the VRF IP routing table and VRF Cisco Express Forwarding table, packets are forwarded to their destination using MPLS.

A PE router binds a label to each customer prefix learned from a CE router and includes the label in the network reachability information for the prefix that it advertises to other PE routers. When a PE router forwards a packet received from a CE router across the provider network, it labels the packet with the label learned from the destination PE router. When the destination PE router receives the labeled packet, it pops the label and uses it to direct the packet to the correct CE router. Label forwarding across the provider backbone is based on either dynamic label switching or traffic engineered paths. A customer data packet carries two levels of labels when traversing the backbone:

- The top label directs the packet to the correct PE router.
- The second label indicates how that PE router should forward the packet to the CE router.

Major Components of MPLS VPNs

An MPLS-based VPN network has three major components:

- VPN route target communities--A VPN route target community is a list of all members of a VPN community. VPN route targets need to be configured for each VPN community member.

- Multiprotocol BGP (MP-BGP) peering of VPN community PE routers--MP-BGP propagates VRF reachability information to all members of a VPN community. MP-BGP peering needs to be configured in all PE routers within a VPN community.
- MPLS forwarding--MPLS transports all traffic between all VPN community members across a VPN service-provider network.

A one-to-one relationship does not necessarily exist between customer sites and VPNs. A given site can be a member of multiple VPNs. However, a site can associate with only one VRF. A customer-site VRF contains all the routes available to the site from the VPNs of which it is a member.

Benefits of an MPLS VPN

MPLS VPNs allow service providers to deploy scalable VPNs and build the foundation to deliver value-added services, such as the following:

Connectionless Service

A significant technical advantage of MPLS VPNs is that they are connectionless. The Internet owes its success to its basic technology, TCP/IP. TCP/IP is built on packet-based, connectionless network paradigm. This means that no prior action is necessary to establish communication between hosts, making it easy for two parties to communicate. To establish privacy in a connectionless IP environment, current VPN solutions impose a connection-oriented, point-to-point overlay on the network. Even if it runs over a connectionless network, a VPN cannot take advantage of the ease of connectivity and multiple services available in connectionless networks. When you create a connectionless VPN, you do not need tunnels and encryption for network privacy, thus eliminating significant complexity.

Centralized Service

Building VPNs in Layer 3 allows delivery of targeted services to a group of users represented by a VPN. A VPN must give service providers more than a mechanism for privately connecting users to intranet services. It must also provide a way to flexibly deliver value-added services to targeted customers. Scalability is critical, because customers want to use services privately in their intranets and extranets. Because MPLS VPNs are seen as private intranets, you may use new IP services such as:

- Multicast
- Quality of service (QoS)
- Telephony support within a VPN
- Centralized services including content and web hosting to a VPN

You can customize several combinations of specialized services for individual customers. For example, a service that combines IP multicast with a low-latency service class enables video conferencing within an intranet.

Scalability

If you create a VPN using connection-oriented, point-to-point overlays, Frame Relay, or ATM virtual connections (VCs), the VPN's key deficiency is scalability. Specifically, connection-oriented VPNs without fully meshed connections between customer sites are not optimal. MPLS-based VPNs instead use the peer model and Layer 3 connectionless architecture to leverage a highly scalable VPN solution. The peer model requires a customer site to peer with only one PE router as opposed to all other customer edge (CE) routers that are members of the VPN. The connectionless architecture allows the creation of VPNs in Layer 3, eliminating the need for tunnels or VCs.

Other scalability issues of MPLS VPNs are due to the partitioning of VPN routes between PE routers and the further partitioning of VPN and IGP routes between PE routers and provider (P) routers in a core network.

- PE routers must maintain VPN routes for those VPNs who are members.
- P routers do not maintain any VPN routes.

This increases the scalability of the provider's core and ensures that no one device is a scalability bottleneck.

Security

MPLS VPNs offer the same level of security as connection-oriented VPNs. Packets from one VPN do not inadvertently go to another VPN.

Security is provided in the following areas:

- At the edge of a provider network, ensuring packets received from a customer are placed on the correct VPN.
- At the backbone, VPN traffic is kept separate. Malicious spoofing (an attempt to gain access to a PE router) is nearly impossible because the packets received from customers are IP packets. These IP packets must be received on a particular interface or subinterface to be uniquely identified with a VPN label.

Easy to Create

To take full advantage of VPNs, customers must be able to easily create new VPNs and user communities. Because MPLS VPNs are connectionless, no specific point-to-point connection maps or topologies are required. You can add sites to intranets and extranets and form closed user groups. Managing VPNs in this manner enables membership of any given site in multiple VPNs, maximizing flexibility in building intranets and extranets.

Flexible Addressing

To make a VPN service more accessible, customers of a service provider can design their own addressing plan, independent of addressing plans for other service provider customers. Many customers use private address spaces, as defined in RFC 1918, and do not want to invest the time and expense of converting to public IP addresses to enable intranet connectivity. MPLS VPNs allow customers to continue to use their present address spaces without network address translation (NAT) by providing a public and private view of the address. A NAT is required only if two VPNs with overlapping address spaces want to communicate. This enables customers to use their own unregistered private addresses, and communicate freely across a public IP network.

Integrated Quality of Service (QoS) Support

QoS is an important requirement for many IP VPN customers. It provides the ability to address two fundamental VPN requirements:

- Predictable performance and policy implementation
- Support for multiple levels of service in an MPLS VPN

Network traffic is classified and labeled at the edge of the network before traffic is aggregated according to policies defined by subscribers and implemented by the provider and transported across the provider core. Traffic at the edge and core of the network can then be differentiated into different classes by drop probability or delay.

Straightforward Migration

For service providers to quickly deploy VPN services, use a straightforward migration path. MPLS VPNs are unique because you can build them over multiple network architectures, including IP, ATM, Frame Relay, and hybrid networks.

Migration for the end customer is simplified because there is no requirement to support MPLS on the CE router and no modifications are required to a customer's intranet.

How to Configure MPLS Layer 3 VPNs

- Configuring the Core Network, page 9
- Connecting the MPLS VPN Customers, page 12
- Verifying the VPN Configuration, page 28
- Verifying Connectivity Between MPLS VPN Sites, page 29

Configuring the Core Network

- Assessing the Needs of MPLS VPN Customers, page 9
- Configuring Routing Protocols in the Core, page 10
- Configuring MPLS in the Core, page 10
- Configuring Multiprotocol BGP on the PE Routers and Route Reflectors, page 10

Assessing the Needs of MPLS VPN Customers

Before you configure an MPLS VPN, you need to identify the core network topology so that it can best serve MPLS VPN customers. Perform this task to identify the core network topology.

SUMMARY STEPS

1. Identify the size of the network.
2. Identify the routing protocols in the core.
3. Determine if you need MPLS VPN High Availability support.
4. Determine if you need BGP load sharing and redundant paths in the MPLS VPN core.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Identify the size of the network.	Identify the following to determine the number of routers and ports you need: • How many customers do you need to support? • How many VPNs are needed per customer? • How many virtual routing and forwarding instances are there for each VPN?
Step 2	Identify the routing protocols in the core.	Determine which routing protocols you need in the core network.

	Command or Action	Purpose
Step 3	Determine if you need MPLS VPN High Availability support.	MPLS VPN Nonstop Forwarding and Graceful Restart are supported on select routers and Cisco software releases. Contact Cisco Support for the exact requirements and hardware support.
Step 4	Determine if you need BGP load sharing and redundant paths in the MPLS VPN core.	See *Load Sharing MPLS VPN Traffic* for configuration steps.

Configuring Routing Protocols in the Core

To configure a routing protocol, such as BGP, OSPF, IS-IS, EIGRP,and static, see the following documents:

- Configuring BGP
- Configuring OSPF
- Configuring IS-IS
- Configuring ERGRP
- Configuring static routes

Configuring MPLS in the Core

To enable MPLS on all routers in the core, you must configure a label distribution protocol. You can use either of the following as a label distribution protocol:

- MPLS Label Distribution Protocol (LDP). For configuration information, see the MPLS Label Distribution Protocol (LDP).
- MPLS Traffic Engineering Resource Reservation Protocol (RSVP). For configuration information, see MPLS Traffic Engineering and Enhancements.

Configuring Multiprotocol BGP on the PE Routers and Route Reflectors

Perform this task to configure multiprotocol BGP (MP-BGP) connectivity on the PE routers and route reflectors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **address-family vpnv4** [**unicast**]
8. **neighbor** {*ip-address* | *peer-group-name*} **send-community extended**
9. **neighbor** {*ip-address* | *peer-group-name*} **activate**
10. **end**

DETAILED STEPS

	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	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**no bgp default ipv4-unicast** **Example:** `Router(config-router)# no bgp default ipv4-unicast`	(Optional) Disables the IPv4 unicast address family on all neighbors. • Use the **no bgp default ipv4-unicast** command if you are using this neighbor for MPLS routes only.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.0.0.1 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.

	Command or Action	Purpose
Step 7	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes. • The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **send-community extended** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 10	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

- Troubleshooting Tips, page 12

Troubleshooting Tips

You can enter a **show ip bgp neighbor** command to verify that the neighbors are up and running. If this command is not successful, enter a **debug ip bgp x.x.x.x events** command, where *x.x.x.x* is the IP address of the neighbor.

Connecting the MPLS VPN Customers

- Defining VRFs on the PE Routers to Enable Customer Connectivity, page 12
- Configuring VRF Interfaces on PE Routers for Each VPN Customer, page 14
- Configuring Routing Protocols Between the PE and CE Routers, page 15

Defining VRFs on the PE Routers to Enable Customer Connectivity

To define VPN routing and forwarding (VRF) instances, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
6. **import map** *route-map*
7. **exit**

DETAILED STEPS

	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 vrf** *vrf-name* **Example:** `Router(config)# ip vrf vpn1`	Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. • The *vrf-name* argument is the name assigned to a VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 100:1`	Creates routing and forwarding tables. • The *route-distinguisher* argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats: ◦ 16-bit AS number: your 32-bit number, for example, 101:3 ◦ 32-bit IP address: your 16-bit number, for example, 10.0.0.1:1

	Command or Action	Purpose
Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf)# route-target import 100:1`	Creates a route-target extended community for a VRF. • The **import** keyword imports routing information from the target VPN extended community. • The **export** keyword exports routing information to the target VPN extended community. • The **both** keyword imports routing information from and exports routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities.
Step 6	**import map** *route-map* **Example:** `Router(config-vrf)# import map vpn1-route-map`	(Optional) Configures an import route map for a VRF. • The *route-map* argument specifies the route map to be used as an import route map for the VRF.
Step 7	**exit** **Example:** `Router(config-vrf)# exit`	(Optional) Exits to global configuration mode.

Configuring VRF Interfaces on PE Routers for Each VPN Customer

To associate a VRF with an interface or subinterface on the PE routers, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip vrf forwarding** *vrf-name*
5. **end**

DETAILED STEPS

	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** *type number* **Example:** `Router(config)# interface Ethernet 5/0`	Specifies the interface to configure and enters interface configuration mode. • The *type* argument specifies the type of interface to be configured. • The *number* argument specifies the port, connector, or interface card number.
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vpn1`	Associates a VRF with the specified interface or subinterface. • The *vrf-name*argument is the name assigned to a VRF.
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.

Configuring Routing Protocols Between the PE and CE Routers

Configure the PE router with the same routing protocol that the CE router uses. You can configure the following routing protocols:

- Configuring BGP as the Routing Protocol Between the PE and CE Routers, page 15
- Configuring RIPv2 as the Routing Protocol Between the PE and CE Routers, page 17
- Configuring Static Routes Between the PE and CE Routers, page 19
- Configuring OSPF as the Routing Protocol Between the PE and CE Routers, page 21
- Configuring EIGRP as the Routing Protocol Between the PE and CE Routers, page 23
- Configuring EIGRP Redistribution in the MPLS VPN, page 26

Configuring BGP as the Routing Protocol Between the PE and CE Routers

To configure PE-to-CE routing sessions using BGP, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **exit-address-family**
8. **end**

DETAILED STEPS

	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	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**address-family ipv4** [**multicast**	**unicast**	**vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 vrf vpn1`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.

	Command or Action	Purpose
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor 10.0.0.1 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 8	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Configuring RIPv2 as the Routing Protocol Between the PE and CE Routers

To configure PE-to-CE routing sessions using RIPv2, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router rip**
4. **version** {**1** | **2**}
5. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
6. **network** *ip-address*
7. **redistribute** *protocol* | [**process-id**] | {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**]
8. **exit-address-family**
9. **end**

DETAILED STEPS

	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	**router rip** **Example:** `Router(config)# router rip`	Enables RIP.
Step 4	**version** {**1** | **2**} **Example:** `Router(config-router)# version 2`	Specifies a Routing Information Protocol (RIP) version used globally by the router.
Step 5	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 vrf vpn1`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specifies the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**network** *ip-address* **Example:** `Router(config-router-af)# network 192.168.7.0`	Enables RIP on the PE-to-CE link.

	Command or Action	Purpose
Step 7	**redistribute** *protocol* | [**process-id**] | {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**] **Example:** `Router(config-router-af)# redistribute bgp 200`	Redistributes routes from one routing domain into another routing domain. • For the RIPv2 routing protocol, use the **redistribute bgp** *as-number* command.
Step 8	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 9	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Configuring Static Routes Between the PE and CE Routers

To configure PE-to-CE routing sessions that use static routes, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route vrf** *vrf-name*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **redistribute** *protocol* | [**process-id**] | {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**]
6. **redistribute** *protocol* | [**process-id**] | {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**]
7. **exit-address-family**
8. **end**

DETAILED STEPS

	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 route vrf** *vrf-name* **Example:** `Router(config)# ip route vrf 200`	Defines static route parameters for every PE-to-CE session.
Step 4	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 vrf vpn1`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specifies the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**redistribute** *protocol* | [**process-id**] | {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**] **Example:** `Router(config-router-af)# redistribute static`	Redistributes routes from one routing domain into another routing domain. • To redistribute VRF static routes into the VRF BGP table, use the **redistribute static** command. See the command for information about other arguments and keywords.

	Command or Action	Purpose
Step 6	**redistribute** *protocol* | [**process-id**] | {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**] **Example:** `Router(config-router-af)# redistribute connected`	Redistributes routes from one routing domain into another routing domain. • To redistribute directly connected networks into the VRF BGP table, use the **redistribute connected** command.
Step 7	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 8	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Configuring OSPF as the Routing Protocol Between the PE and CE Routers

To configure PE-to-CE routing sessions that use OSPF, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id* [**vrf** *vpn-name*]
4. **network** *ip-address wildcard-mask* **area** *area-id*
5. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
6. **redistribute** *protocol* | [**process-id**] | {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**]
7. **exit-address-family**
8. **end**

DETAILED STEPS

	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	**router ospf** *process-id* [**vrf** *vpn-name*] **Example:** `Router(config)# router ospf 1 vrf grc`	Enables OSPF routing and enters router configuration mode. • The *process-id* argument identifies the OSPF process. • The **vrf** *vpn-name* keyword and argument identify a VPN. Create a separate OSPF process for each VRF that will receive VPN routes.
Step 4	**network** *ip-address wildcard-mask* **area** *area-id* **Example:** `Router(config-router)# network 10.0.0.1 0.0.0.3 area 20`	Defines the interfaces on which OSPF runs and to defines the area ID for those interfaces. • The *ip-address* argument identifies the IP address. • The *wildcard-mask* argument identifies the IP-address-type mask that includes “don't care” bits. • The *area-id* argument identifies the area that is to be associated with the OSPF address range. It can be specified as either a decimal value or as an IP address. To associate areas with IP subnets, specify a subnet address as the value of the *area-id*argument.
Step 5	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 vrf vpn1`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.

	Command or Action	Purpose
Step 6	**redistribute** *protocol* | [**process-id**] | {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**] **Example:** `Router(config-router-af)#  redistribute rip metric 1 subnets`	Redistributes routes from one routing domain into another routing domain. You may need to include several protocols to ensure that all IBGP routes are distributed into the VRF.
Step 7	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 8	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Configuring EIGRP as the Routing Protocol Between the PE and CE Routers

Using Enhanced Interior Gateway Routing Protocol (EIGRP) between the PE and CE routers allows you to transparently connect EIGRP customer networks through an MPLS-enabled BGP core network so that EIGRP routes are redistributed through the VPN across the BGP network as internal BGP (iBGP) routes.

To configure PE-to-CE routing sessions that use EIGRP, perform this task.

BGP must be configured in the network core.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no synchronization**
5. **neighbor** *ip-address* **remote-as** *as-number*
6. **neighbor** *ip-address update-source* **loopback** *interface-number*
7. **address-family vpnv4**
8. **neighbor** *ip-address* **activate**
9. **neighbor** *ip-address* **send-community extended**
10. **exit-address-family**
11. **address-family ipv4 vrf** *vrf-name*
12. **redistribute eigrp** *as-number* [**metric** *metric-value*] [**route-map** *map-name*]
13. **no synchronization**
14. **exit-address-family**
15. **end**

DETAILED STEPS

	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	**router bgp** *as-number* **Example:** `Router(config)# router bgp 10`	Enters router configuration mode, and creates a BGP routing process.
Step 4	**no synchronization** **Example:** `Router(config-router)# no synchronization`	Configures BGP to send advertisements without waiting to synchronize with the IGP.

	Command or Action	Purpose
Step 5	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.0.0.1 remote-as 10`	Establishes peering with the specified neighbor or peer-group. • In this step, you are establishing an iBGP session with the PE router that is connected to the CE router at the other CE site.
Step 6	**neighbor** *ip-address update-source* **loopback** *interface-number* **Example:** `Router(config-router)# neighbor 10.0.0.1 update-source loopback 0`	Configures BGP to use any operational interface for TCP connections. • This configuration step is not required. However, the BGP routing process will be less susceptible to the affects of interface or link flapping.
Step 7	**address-family vpnv4** **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions that use standard IPv4 address prefixes, such as BGP, RIP, and static routing sessions.
Step 8	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 activate`	Establishes peering with the specified neighbor or peer-group. • In this step, you are activating the exchange of VPNv4 routing information between the PE routers.
Step 9	**neighbor** *ip-address* **send-community extended** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 send-community extended`	Configures the local router to send extended community attribute information to the specified neighbor. • This step is required for the exchange of EIGRP extended community attributes.
Step 10	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode and enters router configuration mode.

	Command or Action	Purpose
Step 11	**address-family ipv4 vrf** *vrf-name* **Example:** `Router(config-router)# address-family ipv4 vrf RED`	Configures an IPv4 address-family for the EIGRP VRF and enters address family configuration mode. • An address-family VRF needs to be configured for each EIGRP VRF that runs between the PE and CE routers.
Step 12	**redistribute eigrp** *as-number* [**metric** *metric-value*] [**route-map** *map-name*] **Example:** `Router(config-router-af)# redistribute eigrp 101`	Redistributes the EIGRP VRF into BGP. • The autonomous system number from the CE network is configured in this step.
Step 13	**no synchronization** **Example:** `Router(config-router-af)# no synchronization`	Configures BGP to send advertisements without waiting to synchronize with the IGP.
Step 14	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode and enters router configuration mode.
Step 15	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.

Configuring EIGRP Redistribution in the MPLS VPN

Perform this task to every PE router that provides VPN services to enable EIGRP redistribution in the MPLS VPN.

The metric must be configured for routes from external EIGRP autonomous systems and non-EIGRP networks before these routes can be redistributed into an EIGRP CE router. The metric can be configured in the redistribute statement using the redistribute (IP) command or configured with the default-metric (EIGRP) command. If an external route is received from another EIGRP autonomous system or a non-EIGRP network without a configured metric, the route will not be advertised to the CE router.

Note Redistribution between native EIGRP VRFs is not supported. This is designed behavior.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **network** *ip-address wildcard-mask*
6. **redistribute bgp** {*as-number*} [**metric** *bandwidth delay reliability load mtu*] [**route-map** *map-name*]
7. **autonomous-system** *as-number*
8. **exit-address-family**
9. **end**

DETAILED STEPS

	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	**router eigrp** *as-number* **Example:** `Router(config)# router eigrp 1`	Enters router configuration mode and creates an EIGRP routing process. • The EIGRP routing process for the PE router is created in this step.
Step 4	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 vrf RED`	Enters address-family configuration mode and creates a VRF. • The VRF name must match the VRF name that was created in the previous section.

	Command or Action	Purpose
Step 5	**network** *ip-address wildcard-mask* **Example:** `Router(config-router-af)# network 172.16.0.0 0.0.255.255`	Specifies the network for the VRF. • The network statement is used to identify which interfaces to include in EIGRP. The VRF must be configured with addresses that fall within the wildcard-mask range of the network statement.
Step 6	**redistribute bgp** {*as-number*} [**metric** *bandwidth delay reliability load mtu*] [**route-map** *map-name*] **Example:** `Router(config-router-af)# redistribute bgp 10 metric 10000 100 255 1 1500`	Redistributes BGP into the EIGRP. • The autonomous system number and metric of the BGP network is configured in this step. BGP must be redistributed into EIGRP for the CE site to accept the BGP routes that carry the EIGRP information. A metric must also be specified for the BGP network and is configured in this step.
Step 7	**autonomous-system** *as-number* **Example:** `Router(config-router-af)# autonomous-system 101`	Specifies the autonomous system number of the EIGRP network for the customer site.
Step 8	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode and enters router configuration mode.
Step 9	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.

Verifying the VPN Configuration

A route distinguisher must be configured for the VRF, and MPLS must be configured on the interfaces that carry the VRF. Use the **show ip vrf** command to verify the route distinguisher (RD) and interface that are configured for the VRF.

SUMMARY STEPS

1. **show ip vrf**

DETAILED STEPS

show ip vrf
Use this command to display the set of defined VRF instances and associated interfaces. The output also maps the VRF instances to the configured route distinguisher.

Verifying Connectivity Between MPLS VPN Sites

To verify that the local and remote CE routers can communicate across the MPLS core, perform the following tasks:

- Verifying IP Connectivity from CE Router to CE Router Across the MPLS Core, page 29
- Verifying that the Local and Remote CE Routers Are in the Routing Table, page 30

Verifying IP Connectivity from CE Router to CE Router Across the MPLS Core

Perform this task to verify IP connectivity from CE router to CE router across the MPLS VPN.

SUMMARY STEPS

1. **enable**
2. **ping** [*protocol*] {*host-name* | *system-address*}
3. **trace** [*protocol*] [*destination*]
4. **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | *protocol* [*process-id*]] | [**list** [*access-list-name* | *access-list-number*]

DETAILED STEPS

Step 1 **enable**
Use this command to enable privileged EXEC mode.

Step 2 **ping** [*protocol*] {*host-name* | *system-address*}
Use this command to diagnoses basic network connectivity on AppleTalk, CLNS, IP, Novell, Apollo, VINES, DECnet, or XNS networks. Use the **ping** command to verify the connectivity from one CE router to another.

Step 3 **trace** [*protocol*] [*destination*]
Use this command to discover the routes that packets take when traveling to their destination. Use the **trace** command to verify the path that a packet goes through before reaching the final destination. The **trace** command can help isolate a trouble spot if two routers cannot communicate.

Step 4 **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | *protocol* [*process-id*]] | [**list** [*access-list-name* |*access-list-number*]
Use this command to display the current state of the routing table. Use the *ip-address* argument to verify that CE1 has a route to CE2. Verify the routes learned by CE1. Make sure that the route for CE2 is listed.

Verifying that the Local and Remote CE Routers Are in the Routing Table

Perform this task to check that the local and remote CE routers are in the routing table of the PE routers.

SUMMARY STEPS

1. **enable**
2. **show ip route vrf** *vrf-name* [*prefix*]
3. **show ip cef vrf** *vrf-name* [*ip-prefix*]
4. **exit**

DETAILED STEPS

Step 1 **enable**
Use this command to enable privileged EXEC mode.

Step 2 **show ip route vrf** *vrf-name* [*prefix*]
Use this command to display the IP routing table associated with a VRF. Check that the loopback addresses of the local and remote CE routers are in the routing table of the PE routers.

Step 3 **show ip cef vrf** *vrf-name* [*ip-prefix*]
Use this command to display the Cisco Express Forwarding forwarding table associated with a VRF. Check that the prefix of the remote CE router is in the Cisco Express Forwarding table.

Step 4 **exit**

Configuration Examples for MPLS VPNs

- Configuring an MPLS VPN Using BGP Example, page 30
- Configuring an MPLS VPN Using RIP Example, page 31
- Configuring an MPLS VPN Using Static Routes Example, page 32
- Configuring an MPLS VPN Using OSPF Example, page 33
- Configuring an MPLS VPN Using EIGRP Example, page 34

Configuring an MPLS VPN Using BGP Example

This example shows an MPLS VPN that is configured using BGP.

PE Configuration	CE Configuration
ip vrf vpn1 rd 100:1 route-target export 100:1 route-target import 100:1 ! ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.1 255.255.255.255 ! interface Ethernet0/0 ip vrf forwarding vpn1 ip address 34.0.0.2 255.0.0.0 no cdp enable ! interface Ethernet 1/1 ip address 30.0.0.1 255.0.0.0 mpls label protocol ldp mpls ip ! router ospf 100 network 10.0.0. 0.0.0.0 area 100 network 30.0.0.0 0.255.255.255 area 100 ! router bgp 100 no synchronization bgp log-neighbor changes neighbor 10.0.0.3 remote-as 100 neighbor 10.0.0.3 update-source Loopback0 no auto-summary ! address-family vpnv4 neighbor 10.0.0.3 activate neighbor 10.0.0.3 send-community extended bgp scan-time import 5 exit-address-family ! address-family ipv4 vrf vpn1 redistribute connected neighbor 34.0.0.1 remote-as 200 neighbor 34.0.0.1 activate neighbor 34.0.0.1 as-override neighbor 34.0.0.1 advertisement-interval 5 no auto-summary no synchronization exit-address-family	ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.9 255.255.255.255 ! interface Ethernet0/0 ip address 34.0.0.1 255.0.0.0 no cdp enable ! router bgp 200 bgp log-neighbor-changes neighbor 34.0.0.2 remote-as 100 ! address-family ipv4 redistribute connected neighbor 34.0.0.2 activate neighbor 34.0.0.2 advertisement-interval 5 no auto-summary no synchronization exit-address-family

Configuring an MPLS VPN Using RIP Example

This example shows an MPLS VPN that is configured using RIP.

PE Configuration	CE Configuration
ip vrf vpn1 rd 100:1 route-target export 100:1 route-target import 100:1 ! ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.1 255.255.255.255 ! interface Ethernet0/0 ip vrf forwarding vpn1 ip address 34.0.0.2 255.0.0.0 no cdp enable interface Ethernet 1/1 ip address 30.0.0.1 255.0.0.0 mpls label protocol ldp mpls ip ! router rip version 2 timers basic 30 60 60 120 ! address-family ipv4 vrf vpn1 version 2 redistribute bgp 100 metric transparent network 34.0.0.0 distribute-list 20 in no auto-summary exit-address-family ! router bgp 100 no synchronization bgp log-neighbor changes neighbor 10.0.0.3 remote-as 100 neighbor 10.0.0.3 update-source Loopback0 no auto-summary ! address-family vpnv4 neighbor 10.0.0.3 activate neighbor 10.0.0.3 send-community extended bgp scan-time import 5 exit-address-family ! address-family ipv4 vrf vpn1 redistribute connected redistribute rip no auto-summary no synchronization exit-address-family	ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.9 255.255.255.255 ! interface Ethernet0/0 ip address 34.0.0.1 255.0.0.0 no cdp enable router rip version 2 timers basic 30 60 60 120 redistribute connected network 10.0.0.0 network 34.0.0.0 no auto-summary

Configuring an MPLS VPN Using Static Routes Example

This example shows an MPLS VPN that is configured using static routes.

PE Configuration	CE Configuration
ip vrf vpn1 rd 100:1 route-target export 100:1 route-target import 100:1 ! ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.1 255.255.255.255 ! interface Ethernet0/0 ip vrf forwarding vpn1 ip address 34.0.0.2 255.0.0.0 no cdp enable ! interface Ethernet 1/1 ip address 30.0.0.1 255.0.0.0 mpls label protocol ldp mpls ip ! router ospf 100 network 10.0.0. 0.0.0.0 area 100 network 30.0.0.0 0.255.255.255 area 100 ! router bgp 100 no synchronization bgp log-neighbor changes neighbor 10.0.0.3 remote-as 100 neighbor 10.0.0.3 update-source Loopback0 no auto-summary ! address-family vpnv4 neighbor 10.0.0.3 activate neighbor 10.0.0.3 send-community extended bgp scan-time import 5 exit-address-family ! address-family ipv4 vrf vpn1 redistribute connected redistribute static no auto-summary no synchronization exit-address-family ! ip route vrf vpn1 10.0.0.9 255.255.255.255 34.0.0.1 ip route vrf vpn1 34.0.0.0 255.0.0.0 34.0.0.1	ip cef ! interface Loopback0 ip address 10.0.0.9 255.255.255.255 ! interface Ethernet0/0 ip address 34.0.0.1 255.0.0.0 no cdp enable ! ip route 10.0.0.9 255.255.255.255 34.0.0.2 3 ip route 31.0.0.0 255.0.0.0 34.0.0.2 3

Configuring an MPLS VPN Using OSPF Example

This example shows an MPLS VPN that is configured using OSPF.

PE Configuration	CE Configuration
ip vrf vpn1 rd 100:1 route-target export 100:1 route-target import 100:1 ! ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.1 255.255.255.255 ! interface Ethernet0/0 ip vrf forwarding vpn1 ip address 34.0.0.2 255.0.0.0 no cdp enable ! router ospf 1000 vrf vpn1 log-adjacency-changes redistribute bgp 100 metric-type 1 subnets network 10.0.0.13 0.0.0.0 area 10000 network 34.0.0.0 0.255.255.255 area 10000 ! router bgp 100 no synchronization bgp log-neighbor changes neighbor 10.0.0.3 remote-as 100 neighbor 10.0.0.3 update-source Loopback0 no auto-summary ! address-family vpnv4 neighbor 10.0.0.3 activate neighbor 10.0.0.3 send-community extended bgp scan-time import 5 exit-address-family ! address-family ipv4 vrf vpn1 redistribute connected redistribute ospf 1000 match internal external 1 external 2 no auto-summary no synchronization exit-address-family	ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.9 255.255.255.255 ! interface Ethernet0/0 ip address 34.0.0.1 255.0.0.0 no cdp enable ! router ospf 1000 log-adjacency-changes auto-cost reference-bandwidth 1000 redistribute connected subnets network 34.0.0.0 0.255.255.255 area 1000 network 10.0.0.0 0.0.0.0 area 1000

Configuring an MPLS VPN Using EIGRP Example

This example shows an MPLS VPN that is configured using EIGRP.

PE Configuration	CE Configuration
ip vrf vpn1 rd 100:1 route-target export 100:1 route-target import 100:1 ! ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.1 255.255.255.255 interface Ethernet0/0 ip vrf forwarding vpn1 ip address 34.0.0.2 255.0.0.0 no cdp enable interface Ethernet 1/1 ip address 30.0.0.1 255.0.0.0 mpls label protocol ldp mpls ip router eigrp 1000 auto-summary ! address-family ipv4 vrf vpn1 redistribute bgp 100 metric 10000 100 255 1 1500 network 34.0.0.0 distribute-list 20 in no auto-summary autonomous-system 1000 exit-address-family ! router bgp 100 no synchronization bgp log-neighbor changes neighbor 10.0.0.3 remote-as 100 neighbor 10.0.0.3 update-source Loopback0 no auto-summary ! address-family vpnv4 neighbor 10.0.0.3 activate neighbor 10.0.0.3 send-community extended bgp scan-time import 5 exit-address-family ! address-family ipv4 vrf vpn1 redistribute connected redistribute eigrp no auto-summary no synchronization exit-address-family	ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.9 255.255.255.255 ! interface Ethernet0/0 ip address 34.0.0.1 255.0.0.0 no cdp enable ! router eigrp 1000 network 34.0.0.0 auto-summary

Additional References

Related Documents

Related Topic	Document Title
MPLS VPN Carrier Supporting Carrier	• MPLS VPN Carrier Supporting Carrier Using LDP and an IGP • MPLS VPN Carrier Supporting Carrier with BGP
MPLS VPN InterAutonomous Systems	• MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels • MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

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 software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2547	BGP/MPLS VPNs

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 MPLS Layer 3 VPNs

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 1 Feature Information for MPLS Layer 3 VPNs

Feature Name	Releases	Feature Configuration Information
MPLS Virtual Private Networks	12.0(5)T 12.0(21)ST 12.0(22)S 12.0(23)S 12.2(13)T 12.2(14)S 12.0(26)S	This feature allows a set of sites that to be interconnected by means of an MPLS provider core network. At each customer site, one or more customer edge (CE) routers attach to one or more provider edge (PE) routers.
MPLS VPN Support for EIGRP Between Provider Edge and Customer Edge	12.0(22)S 12.2(15)T 12.2(18)S 12.0(27)S	This feature allows you to connect customers running EIGRP to an MPLS VPN.

Configuring Route Maps to Control the Distribution of MPLS Labels Between Routers in an MPLS VPN

Route maps enable you to specify which routes are distributed with Multiprotocol Label Switching (MPLS) labels. Route maps also enable you to specify which routes with MPLS labels a router receives and adds to its Border Gateway Protocol (BGP) table.

- Finding Feature Information, page 39
- Restrictions for Using Route Maps with MPLS VPNs, page 39
- Prerequisites for Using Route Maps with MPLS VPNs, page 39
- Information About Route Maps in MPLS VPNs, page 40
- How to Configure Route Maps in an MPLS VPN, page 40
- Configuration Examples for Route Maps in MPLS VPNs, page 46
- Additional References, page 48
- Feature Information for Route Maps in MPLS VPNs, page 50

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Using Route Maps with MPLS VPNs

You can use route maps with MPLS VPN Inter-AS with Autonomous System Boundary Routers (ASBRs) exchanging IPv4 routes with MPLS labels. You cannot use route maps with MPLS VPN Inter-AS with ASBRs exchanging VPN-IPv4 addresses.

Prerequisites for Using Route Maps with MPLS VPNs

Before you configure and apply route maps, you need to create an access control list (ACL) and specify the routes that the router should distribute with MPLS labels.

Information About Route Maps in MPLS VPNs

When routers are configured to distribute routes with MPLS labels, all the routes are encoded with the multiprotocol extensions and contain MPLS labels. You can use a route map to control the distribution of MPLS labels between routers.

Route maps enable you to specify which routes are distributed with MPLS labels. Route maps also enable you to specify which routes with MPLS labels a router receives and adds to its BGP table. Route maps enable you to specify the following:

- For a router distributing MPLS labels, you can specify which routes are distributed with an MPLS label.
- For a router receiving MPLS labels, you can specify which routes are accepted and installed in the BGP table.

Route maps work with ACLs. You enter the routes into an ACL and then specify the ACL when you configure the route map. You can configure a router to accept only routes that are specified in the route map. The router checks the routes listed in the BGP update message against the list of routes in the specified ACL. If a route in the BGP update message matches a route in the ACL, the route is accepted and added to the BGP table.

How to Configure Route Maps in an MPLS VPN

Perform the following tasks to enable routers to send MPLS labels with the routes specified in the route maps:

- Configuring a Route Map for Incoming Routes, page 40
- Configuring a Route Map for Outgoing Routes, page 42
- Applying the Route Maps to the MPLS VPN Edge Routers, page 44

Configuring a Route Map for Incoming Routes

Perform this task to create a route map to filter arriving routes. You create an ACL and specify the routes that the router should accept and add to the BGP table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **route-map** *map-name* [**permit** | **deny**] *sequence-number*
5. **match ip address** {*access-list-number* [*access-list-number...*| *access-list-name...*] *access-list-name* [*access-list-number...*| *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name....*]]
6. **match mpls-label**
7. **exit**

DETAILED STEPS

	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	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**route-map** *map-name* [**permit** | **deny**] *sequence-number* **Example:** `Router(config-router)# route-map csc-mpls-routes-in permit`	Enters route map configuration mode and creates a route map with the name you specify. • The *map-name* argument identifies the name of the route map. • The **permit** keyword allows the actions to happen if all conditions are met. • A **deny** keyword prevents any actions from happening if all conditions are met. • The *sequence-number* argument allows you to prioritize route maps. If you have multiple route maps and want to prioritize them, assign each one a number. The route map with the lowest number is implemented first, followed by the route map with the second lowest number, and so on.
Step 5	**match ip address** {*access-list-number* [*access-list-number...*| *access-list-name...*] *access-list-name* [*access-list-number...*| *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name....*]] **Example:** `Router(config-route-map)# match ip address acl-in`	Distributes any routes that have a destination network number address that is permitted by a standard access list, an extended access list, or a prefix list, or performs policy routing on packets. • The *access-list-number...* argument is a number of a standard or extended access list. It can be an integer from 1 to 199. The ellipsis indicates that multiple values can be entered. • The *access-list-name...* argument is a name of a standard or extended access list. It can be an integer from 1 to 199. The ellipsis indicates that multiple values can be entered. • The **prefix-list** keyword distributes routes based on a prefix list. • The *prefix-list-name...* argument is a name of a specific prefix list. The ellipsis indicates that multiple values can be entered.

	Command or Action	Purpose
Step 6	**match mpls-label** **Example:** `Router(config-route-map)# match mpls-label`	Redistributes routes that include MPLS labels if the routes meet the conditions specified in the route map.
Step 7	**exit** **Example:** `Router(config-route-map)# exit`	Exits route map configuration mode and returns to global configuration mode.

Configuring a Route Map for Outgoing Routes

This configuration is optional.

Perform this task to create a route map to filter departing routes. You create an access list and specify the routes that the router should distribute with MPLS labels.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **route-map** *map-name* [**permit** | **deny**] *sequence-number*
5. **match ip address** {*access-list-number* [*access-list-number*...| *access-list-name*...}] | *access-list-name* [*access-list-number*...| *access-list-name* | **prefix-list** *prefix-list-name* [*prefix-list-name*....]]
6. **set mpls-label**
7. **exit**

DETAILED STEPS

	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	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**route-map** *map-name* [**permit** | **deny**] *sequence-number* **Example:** `Router(config-router)# route-map csc-mpls-routes-out permit`	Enters route map configuration mode and creates a route map with the name you specify. • The *map-name* argument identifies the name of the route map. • The **permit** keyword allows the actions to happen if all conditions are met. • A **deny** keyword prevents any actions from happening if all conditions are met. • The *sequence-number* argument allows you to prioritize route maps. If you have multiple route maps and want to prioritize them, assign each one a number. The route map with the lowest number is implemented first, followed by the route map with the second lowest number, and so on.
Step 5	**match ip address** {*access-list-number* [*access-list-number...*| *access-list-name...*}] | *access-list-name* [*access-list-number...*| *access-list-name* | **prefix-list** *prefix-list-name* [*prefix-list-name....*]] **Example:** `Router(config-route-map)# match ip address acl-out`	Distributes any routes that have a destination network number address that is permitted by a standard access list, an extended access list, or a prefix list, or performs policy routing on packets. • The *access-list-number...* argument is a number of a standard or extended access list. It can be an integer from 1 to 199. The ellipsis indicates that multiple values can be entered. • The *access-list-name...* argument is a name of a standard or extended access list. It can be an integer from 1 to 199. The ellipsis indicates that multiple values can be entered. • The **prefix-list** keyword distributes routes based on a prefix list. • The *prefix-list-name...* argument is a name of a specific prefix list. The ellipsis indicates that multiple values can be entered.
Step 6	**set mpls-label** **Example:** `Router(config-route-map)# set mpls-label`	Enables a route to be distributed with an MPLS label if the route matches the conditions specified in the route map.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config-route-map)# exit`	Exits route map configuration mode and returns to global configuration mode.

Applying the Route Maps to the MPLS VPN Edge Routers

This configuration is optional.

Perform this task to enable the edge routers to use the route maps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **neighbor** *ip-address* **route-map** *map-name* **in**
6. **neighbor** *ip-address* **route-map** *map-name* **out**
7. **neighbor** *ip-address* **send-label**
8. **exit-address-family**
9. **end**

DETAILED STEPS

	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	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 vrf vpn1`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**neighbor** *ip-address* **route-map** *map-name* **in** **Example:** `Router(config-router-af)# neighbor pp. 0.0.1 route-map csc-mpls-routes-in in`	Applies a route map to incoming routes. • The *ip-address* argument specifies the router to which the route map is to be applied. • The *map-name* argument specifies the name of the route map. • The **in** keyword applies the route map to incoming routes.
Step 6	**neighbor** *ip-address* **route-map** *map-name* **out** **Example:** `Router(config-router-af)# neighbor pp. 0.0.1 route-map csc-mpls-route-out out`	Applies a route map to outgoing routes. • The *ip-address* argument specifies the router to which the route map is to be applied. • The *map-name* argument specifies the name of the route map. • The **out** keyword applies the route map to outgoing routes.
Step 7	**neighbor** *ip-address* **send-label** **Example:** `Router(config-router-af)# neighbor pp. 0.0.1 send-label`	Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighboring router.
Step 8	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits from address family configuration mode.

	Command or Action	Purpose
Step 9	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

- Troubleshooting Tips, page 46

Troubleshooting Tips

You can enter a **show route-map** *map-name* command to verify that the route map is applied to the PE routers.

Note After you make any changes to a route map, you need to reset the BGP connection for the changes to take effect.

Configuration Examples for Route Maps in MPLS VPNs

- Using a Route Map in an MPLS VPN Inter-AS Network Example, page 46
- Using a Route Map in an MPLS VPN CSC Network Example, page 47

Using a Route Map in an MPLS VPN Inter-AS Network Example

In this example, a route map is applied to an autonomous system border router (ASBR) that exchanges IPv4 routes and MPLS labels with another ASBR.

- A route map called OUT specifies that the ASBR should distribute the PE1 route (ee.ee) with labels and the RR1 route (aa.aa) without labels.
- A route map called IN specifies that the ASBR should accept the PE2 route (ff.ff) with labels and the RR2 route (bb.bb) without labels.

```
ip subnet-zero
mpls label protocol tdp
!
interface Loopback0
 ip address ww.ww.ww.ww 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Ethernet0/2
 ip address hh.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
interface Ethernet0/3
 ip address dd.0.0.1 255.0.0.0
 no ip directed-broadcast
```

```
 no ip mroute-cache
 mpls label protocol ldp
 tag-switching ip
!
router ospf 10
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Ethernet0/2
 network ww.ww.ww.ww 0.0.0.0 area 100
 network dd.0.0.0 0.255.255.255 area 100

router bgp 100
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor aa.aa.aa.aa remote-as 100
 neighbor aa.aa.aa.aa update-source Loopback0
 neighbor hh.0.0.1 remote-as 200
 no auto-summary
 !
!
address-family ipv4                      ! Redistributing IGP into BGP
 redistribute ospf 10                    ! so that PE1 & RR1 loopbacks
 neighbor aa.aa.aa.aa activate           ! get into the BGP table
 neighbor aa.aa.aa.aa send-label
 neighbor hh.0.0.1 activate
 neighbor hh.0.0.1 advertisement-interval 5
 neighbor hh.0.0.1 send-label
 neighbor hh.0.0.1 route-map IN in       ! accepting routes in route map IN.
 neighbor hh.0.0.1 route-map OUT out     ! distributing routes in route map OUT.
 neighbor kk.0.0.1 activate
 neighbor kk.0.0.1 advertisement-interval 5
 neighbor kk.0.0.1 send-label
 neighbor kk.0.0.1 route-map IN in       ! accepting routes in route map IN.
 neighbor kk.0.0.1 route-map OUT out     ! distributing routes in route map OUT.
 no auto-summary
 no synchronization
 exit-address-family
!
ip default-gateway 3.3.0.1
ip classless
!
access-list 1 permit ee.ee.ee.ee log               !Setting up the access lists
access-list 2 permit ff.ff.ff.ff log
access-list 3 permit aa.aa.aa.aa log
access-list 4 permit bb.bb.bb.bb log
route-map IN permit 10                             !Setting up the route maps
 match ip address 2
 match mpls-label
!
route-map IN permit 11
 match ip address 4
!
route-map OUT permit 12
 match ip address 3
!
route-map OUT permit 13
 match ip address 1
 set mpls-label
!
end
```

Using a Route Map in an MPLS VPN CSC Network Example

The following example creates two route maps, which are named:

- IN for incoming routes
- OUT for outgoing routes

The route maps specify the following:

- If an IP address in an incoming BGP update message matches an IP address in access list 99, the route is added to the BGP table.
- If an IP address in an outbound BGP update message matches an IP address in access list 88, the router distributes that route.

The route maps are applied to the CSC-PE router with the address qq.0.0.1.

```
address-family ipv4 vrf vpn2
 neighbor qq.0.0.1 remote-as 200
 neighbor qq.0.0.1 activate
 neighbor qq.0.0.1 as-override
 neighbor qq.0.0.1 advertisement-interval 5
 neighbor qq.0.0.1 route-map IN in
 neighbor qq.0.0.1 route-map OUT out
 neighbor qq.0.0.1 send-label
!
access-list 88 permit rr.rr.rr.rr
access-list 88 permit ss.ss.ss.ss
access-list 88 permit tt.tt.tt.tt
access-list 99 permit uu.uu.uu.uu
access-list 99 permit vv.vv.vv.vv
access-list 99 permit ww.ww.ww.ww
!
route-map IN permit 1
 match ip address 99
!
route-map OUT permit 1
 match ip address 88
 set mpls-label
!
```

Additional References

Related Documents

Related Topic	Document Title
Basic MPLS VPNs	Configuring MPLS Layer 3 VPNs
MPLS VPN Carrier Supporting Carrier	• MPLS VPN Carrier Supporting Carrier Using LDP and an IGP • MPLS VPN Carrier Supporting Carrier with BGP
MPLS VPN InterAutonomous Systems	• MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels • MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

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 software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2547	BGP/MPLS VPNs

Technical Assistance

Description	Link
The Cisco Technical Support 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
The Cisco Technical Support 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.	
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.	

Feature Information for Route Maps in MPLS VPNs

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 2 Feature Information for Route Maps in MPLS VPNs

Feature Name	Releases	Feature Configuration Information
This feature was included as part of the following features: • MPLS VPN Inter-Autonomous Systems - IPv4 BGP Label Distribution • MPLS VPN Carrier Supporting Carrier with IPv4 BGP Label Distribution	12.0(21)ST 12.0(22)S 12.0(23)S 12.2(13)T 12.0(24)S 12.2(14)S 12.0(27)S 12.0(29)S	Route maps enable you to specify which routes are distributed with MPLS labels. Route maps also enable you to specify which routes with MPLS labels a router receives and adds to its BGP table.

Dialing to Destinations with the Same IP Address for MPLS VPNs

The dialer software in Cisco IOS prior to Release 12.2(8)T had no way to dial two different destinations with the same IP address. More specifically, in networks where a network access server (NAS) supports dialing clients with overlapping addresses, dial-out attempts fail. This module explains how to dial to more than one destination with the same IP address.

- Finding Feature Information, page 53
- Prerequisites for Dialing to Destinations with the Same IP Address for MPLS VPNs, page 53
- Restrictions for Dialing to Destinations with the Same IP Address for MPLS VPNs, page 54
- Information About Dialing to Destinations with the Same IP Address for MPLS VPNs, page 55
- How to Enable Dialing to Destinations with the Same IP Address for MPLS VPNs, page 56
- Configuration Examples for Dialing to Destinations with the Same IP Address, page 59
- Additional References, page 63
- Feature Information for Dialing to Destinations with the Same IP Address, page 65

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Dialing to Destinations with the Same IP Address for MPLS VPNs

Before configuring this feature, you should understand how to configure the following network features:

- Virtual profiles with two-way AAA authentication
- MPLS VPNs

Refer to the documents listed in the Additional References, page 63 section for information about configuring these features.

Restrictions for Dialing to Destinations with the Same IP Address for MPLS VPNs

When configuring static routes in an MPLS or MPLS VPN environment, some variations of the **ip route** and **ip route vrf** commands are not supported. These variations of the commands are not supported in Cisco IOS releases that support the Tag Forwarding Information Base (TFIB), specifically Cisco IOS Releases 12.xT, 12.xM, and 12.0S. The TFIB cannot resolve prefixes when the recursive route over which the prefixes travel disappears and then reappears. However, the command variations are supported in Cisco IOS releases that support the MPLS Forwarding Infrastructure (MFI), specifically Cisco IOS Release 12.2(25)S and later. Use the following guidelines when configuring static routes.

Supported Static Routes in an MPLS Environment

The following **ip route** command is supported when you configure static routes in MPLS environment:

ip route *destination-prefix mask interface next-hop-address*

The following **ip route** commands are supported when you configure static routes in an MPLS environment and configure load sharing with static nonrecursive routes and a specific outbound interface:

ip route *destination-prefix mask* **interface1 next-hop1**

ip route *destination-prefix mask* **interface2 next-hop2**

Unsupported Static Routes in an MPLS Environment that Uses the TFIB

The following **ip route** command is not supported when you configure static routes in an MPLS environment:

ip route *destination-prefix mask next-hop-address*

The following **ip route** command is not supported when you configure static routes in an MPLS environment and enable load sharing where the next hop can be reached through two paths:

ip route *destination-prefix mask next-hop-address*

The following **ip route** command is not supported when you configure static routes in an MPLS environment and enable load sharing where the destination can be reached through two next hops:

ip route *destination-prefix mask* **next-hop1**

ip route *destination-prefix mask* **next-hop2**

Use the *interface* an *next-hop* arguments when specifying static routes.

Supported Static Routes in an MPLS VPN Environment

The following **ip route vrf** commands are supported when you configure static routes in a MPLS VPN environment, and the next hop and interface are in the same VRF:

- ◦ **ip route vrf** *vrf-name destination-prefix mask next-hop-address*
  ◦ **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address*
  ◦ **ip route vrf** *vrf-name destination-prefix mask* **interface1 next-hop1**
  ◦ **ip route vrf** *vrf-name destination-prefix mask* **interface2 next-hop2**

The following **ip route vrf** commands are supported when you configure static routes in a MPLS VPN environment, and the next hop is in the global table in the MPLS cloud in the global routing table. For example, these commands are supported when the next hop is pointing to the Internet Gateway.

- **ip route vrf** *vrf-name destination-prefix mask next-hop-address* **global**
- **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address* (This command is supported when the next hop and interface are in the core.)

The following **ip route** commands are supported when you configure static routes in a MPLS VPN environment and enable load sharing with static nonrecursive routes and a specific outbound interfaces:

ip route *destination-prefix mask* **interface1 next-hop1**

ip route *destination-prefix mask* **interface2 next-hop2**

Unsupported Static Routes in an MPLS VPN Environment that Uses the TFIB

The following **ip route** command is not supported when you configure static routes in a MPLS VPN environment, the next hop is in the global table in the MPLS cloud within the core, and you enable load sharing where the next hop can be reached through two paths:

ip route vrf *vrf-name destination-prefix mask next-hop-address* **global**

The following **ip route** commands are not supported when you configure static routes in a MPLS VPN environment, the next hop is in the global table in the MPLS cloud within the core, and you enable load sharing where the destination can be reached through two next hops:

ip route vrf *destination-prefix mask* **next-hop1 global**

ip route vrf *destination-prefix mask* **next-hop2 global**

The following **ip route vrf** commands are not supported when you configure static routes in an MPLS VPN environment, and the next hop and interface are in the same VRF:

ip route vrf *vrf-name destination-prefix mask* **next-hop1**

ip route vrf *vrf-name destination-prefix mask* **next-hop2**

Supported Static Routes in an MPLS VPN Environment Where the Next Hop Resides in the Global Table on the CE Router

The following **ip route vrf** command is supported when you configure static routes in a MPLS VPN environment, and the next hop is in the global table on the CE side. For example, the following command is supported when the destination-prefix is the CE router's loopback address, as in EBGP multihop cases.

ip route vrf *vrf-name destination-prefix mask interface next-hop-address*

The following **ip route** commands are supported when you configure static routes in a MPLS VPN environment, the next hop is in the global table on the CE side, and you enable load sharing with static non-recursive routes and a specific outbound interfaces:

ip route *destination-prefix mask* **interface1 nexthop1**

ip route *destination-prefix mask* **interface2 nexthop2**

Information About Dialing to Destinations with the Same IP Address for MPLS VPNs

- Introduction to Dialing to Destinations with the Same IP Address for MPLS VPNs, page 56
- Benefits of this Feature, page 56

Introduction to Dialing to Destinations with the Same IP Address for MPLS VPNs

The Cisco IOS dialer software can distinguish between two destinations with the same IP address using information stored in the VRF. This capability is provided to the dialer software by two existing Cisco IOS commands, **dialer map** and **ip route**, which have been enhanced to include VPN routing and forwarding (VRF) information.

In previous Cisco IOS releases, the dialer software obtained the telephone number for dial-out based on the destination IP address configured in the **dialer map** command. Now, the enhanced **dialer map** command supplies the name of the VRF so that the telephone number to be dialed is based on the VRF name and the destination IP address. The VRF is identified based on the incoming interface of the packet, and is used with the destination IP address defined in the **dialer map** command to determine the telephone number to be dialed.

The **ip route** configuration command also includes the VRF information. When a packet arrives in an incoming interface that belongs to a particular VRF, only those **ip route** commands that correspond to that particular VRF are used to determine the destination interface.

Benefits of this Feature

This feature allows the dialer software to dial out in an MPLS-based VPN. The MPLS VPN model simplifies network routing. For example, rather than needing to manage routing over a complex virtual network backbone composed of many virtual circuits, an MPLS VPN user can employ the backbone of the service provider as the default route in communicating with all other VPN sites.

This default route capability allows several sites to transparently interconnect through the service provider network. One service provider network can support several different IP VPNs, each of which appears to its users as a separate, private network. Within a VPN, each site can send IP packets to any other site in the same VPN, because each VPN is associated with one or more VRFs. The VRF is a key element in the VPN technology, because it maintains the routing information that defines a customer VPN site.

How to Enable Dialing to Destinations with the Same IP Address for MPLS VPNs

- Mapping the VRF and Next-Hop Address to a Dial String, page 56
- Verifying the Configuration, page 58

Mapping the VRF and Next-Hop Address to a Dial String

Use the following procedure to map a VRF and next-hop address combination to a dial string and thereby allow the dialer software to be VRF-aware for an MPLS VPN.

These commands are only part of the required configuration and show how to map a VRF and next-hop address combination to a dial string. Refer to the documents listed in the Additional References, page 63 section and the example in the Configuration Examples for Dialing to Destinations with the Same IP Address, page 59 section for details on where to include these commands in the network configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface dialer** *number*
4. **dialer map ip** *protocol-next-hop-address* **vrf** *vrf-name* **name** *host-name dial-string*
5. **end**
6. **ip route vrf** *vrf-name ip-address mask interface-type interface-number*

DETAILED STEPS

	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 dialer** *number* **Example:** `Router(config)# interface dialer 1`	Enters interface configuration mode and begins dialer configuration.
Step 4	**dialer map ip** *protocol-next-hop-address* **vrf** *vrf-name* **name** *host-name dial-string* **Example:** `Router(config-if)# dialer map ip 60.0.0.12 vrf yellow name rubbertree02 5552171`	Maps a VRF and next-hop address combination to a dial string (telephone number).
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode.

Command or Action	Purpose
Step 6 **ip route vrf** *vrf-name ip-address mask interface-type interface-number* **Example:** `Router(config)# ip route vrf blue 10.0.0.1 255.255.255.255 Dialer0`	Configures a VRF and next hop address combination that points to the interface where the dialer software should make the connection.

Verifying the Configuration

To verify the configuration, use the following procedure.

SUMMARY STEPS

1. **ping**
2. **show adjacency**

DETAILED STEPS

Step 1 **ping**
Use this command on the customer edge NAS to place a call to a peer. The expected result is that the NAS successfully dials out to that peer.

Step 2 **show adjacency**
Use this command if the call fails to check Cisco Express Forwarding (CEF) adjacency table information.

- Troubleshooting Tips, page 58

Troubleshooting Tips

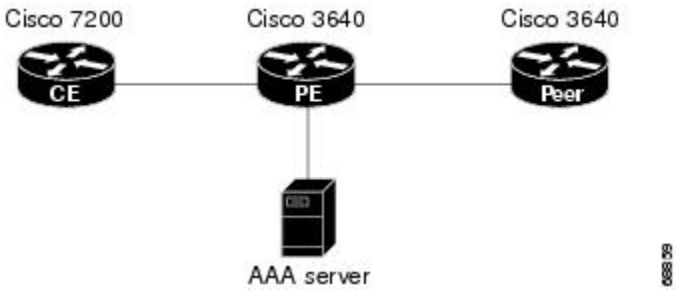
If you encounter problems with the feature, use the following **debug** privileged EXEC commands on the NAS to help you determine where the problem lies:

- **debug aaa authentication**
- **debug aaa authorization**
- **debug dialer**
- **debug ppp authentication**
- **debug ppp negotiation**
- **debug radius**

Configuration Examples for Dialing to Destinations with the Same IP Address

This section provides a configuration example of the feature for a simple network topology shown in the figure below.

Figure 2 MPLS VPN Topology



Note The network addresses and telephone numbers used in the following configuration are examples only and will not work in an actual network configuration.

Customer Edge (CE) Router

```
!
hostname oaktree02
enable secret 5 !1!35Fg$Ep4.D8JGpg7rKxQa49BF9/
!
ip subnet-zero
no ip domain-lookup
!
controller T1 5/0
!
controller T1 5/1
!
interface FastEthernet0/0
 no ip address
 no ip mroute-cache
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 no ip mroute-cache
 shutdown
 duplex auto
 speed auto
!
interface Ethernet1/0
 ip address 10.0.58.11 255.255.255.0
 no ip mroute-cache
 half-duplex
!
interface Ethernet1/1
 ip address 50.0.0.2 255.0.0.0
 no ip mroute-cache
 half-duplex
```

```
!
interface Ethernet1/2
 no ip address
 no ip mroute-cache
 shutdown
 half-duplex
!
interface Ethernet1/3
 no ip address
 no ip mroute-cache
 shutdown
 half-duplex
!
interface Serial2/0
 no ip address
 no ip mroute-cache
 shutdown
 no fair-queue
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 no ip mroute-cache
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 no ip mroute-cache
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 no ip mroute-cache
 shutdown
 serial restart-delay 0
!
interface FastEthernet4/0
 no ip address
 no ip mroute-cache
 shutdown
 duplex auto
 speed auto
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.0.58.1
ip route 60.0.0.0 255.0.0.0 50.0.0.1
no ip http server
!
!
snmp-server manager
banner motd ^C  AV-8B    OAKTREE^C
alias exec r sh run
!
line con 0
 exec-timeout 0 0
line aux 0
 login
line vty 0 4
 no login
!
end
```

Provider Edge (PE) Router

```
hostname pinetree02
!
aaa new-model
!
!
aaa authentication login con-log none
```

```
aaa authentication ppp default group radius
aaa authorization network default group radius
aaa session-id common
enable secret 5 $1$7KlA$xpC8l4dJCZogbzZvGUtFl/
!
username rubbertree02 password 0 Hello
ip subnet-zero
!
no ip domain-lookup
!
ip vrf yellow
 rd 100:1
ip cef
virtual-profile aaa
isdn switch-type primary-5ess
!
controller T1 3/0
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 3/1
 framing esf
 linecode b8zs
!
controller T1 3/2
 framing esf
 linecode b8zs
!
controller T1 3/3
 framing esf
 linecode b8zs
!
controller T1 3/4
 framing esf
 linecode b8zs
!
controller T1 3/5
 framing esf
 linecode b8zs
!
controller T1 3/6
 framing esf
 linecode b8zs
!
controller T1 3/7
 framing esf
 linecode b8zs
!
interface Loopback0
 ip vrf forwarding yellow
 ip address 70.0.0.1 255.0.0.0
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex half
!
interface Ethernet2/0
 ip address 10.0.58.3 255.255.255.0
 duplex full
!
interface Ethernet2/1
 ip vrf forwarding yellow
 ip address 50.0.0.1 255.0.0.0
 duplex half
!
interface Ethernet2/2
 no ip address
 shutdown
 duplex half
!
interface Ethernet2/3
```

```
 no ip address
 shutdown
 duplex half
!
interface Serial3/0:23
 description phone# 555-3123
 no ip address
 encapsulation ppp
 dialer rotary-group 0
 dialer-group 1
 isdn switch-type primary-5ess
 ppp authentication chap
!
interface Serial4/0
 no ip address
 shutdown
 no fair-queue
!
interface Dialer0
 ip address negotiated
 encapsulation ppp
 dialer in-band
 dialer map ip 60.0.0.12 vrf yellow name rubbertree02 5552171
 dialer map ip 60.0.0.2 5552172
 dialer-group 1
 ppp authentication chap
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.0.58.1
ip route 60.0.0.2 255.255.255.255 Dialer0
ip route vrf yellow 60.0.0.0 255.0.0.0 Dialer0 permanent
no ip http server
ip pim bidir-enable
!
ip director cache time 60
dialer-list 1 protocol ip permit
!
radius-server host 172.19.192.89 auth-port 1645 acct-port 1646 key rad123
radius-server retransmit 3
call rsvp-sync
!
no mgcp timer receive-rtcp
!
mgcp profile default
!
dial-peer cor custom
!
gatekeeper
 shutdown
!
banner motd ^C   F/A-18    PINETREE ^C
!
line con 0
 exec-timeout 0 0
 login authentication con-log

line aux 0
line vty 5 15
!
end
```

Peer Router

```
hostname rubbertree02
!
logging buffered 32000 debugging
enable secret 5 $1$RCKC$scgtdlaDzjSyUVAi7KK5Q.
enable password Windy
!
username pinetree02 password 0 Hello
!
ip subnet-zero
```

```
no ip domain-lookup
!
isdn switch-type basic-5ess
!
interface Ethernet0
 ip address 10.0.58.9 255.255.255.0
 no ip route-cache
!
interface BRI0
 description phone# 555-2171
 ip address 60.0.0.12 255.0.0.0
 encapsulation ppp
 no ip route-cache
 dialer map ip 60.0.0.11 5553123
 dialer map ip 60.0.0.2 5552172
 dialer-group 1
 isdn switch-type basic-5ess
 isdn fast-rollover-delay 45
!
ip default-gateway 10.0.58.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.0.58.1
ip route 50.0.0.0 255.0.0.0 70.0.0.1
no ip http server
!
dialer-list 1 protocol ip permit
no cdp run
banner motd ^C   F-4B   RUBBERTREE^C
!
line con 0
 exec-timeout 0 0
line vty 0 4
 password Windy
 login
!
end
```

AAA Server User File

```
[aaa-serv]/usr/testing/bin> ./radiusd_1.16 -d . -a . -x
greentree-16        Password = "Hello", Expiration = "Dec 31 2005"
        Service-Type = Framed-User,
        Framed-Protocol = PPP
        cisco-avpair = "lcp:interface-config=ip vrf forwarding yellow \nip
unnumbered Loopback0"
```

Additional References

Related Documents

Related Topic	Document Title
MPLS	MPLS Product Literature

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 software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1164	Application of the Border Gateway Protocol in the Internet
RFC 1171	A Border Gateway Protocol 4
RFC 1700	Assigned Numbers
RFC 1966	BGP Route Reflection: An Alternative to Full Mesh IBGP
RFC 2283	Multiprotocol Extensions for BGP-4
RFC 2547	BGP/MPLS VPNs
RFC 2842	Capabilities Advertisement with BGP-4
RFC 2858	Multiprotocol Extensions for BGP-4

Technical Assistance

Description	Link
The Cisco Technical Support 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. 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 Dialing to Destinations with the Same IP Address

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 3 Feature Information for Dialing to Destinations with the Same IP Address

Feature Name	Releases	Feature Configuration Information
Dialer Map VRF-Aware for MPLS VPNs	12.2(8)T	The Cisco IOS dialer software is "VRF-aware for an MPLS VPN," which means that it can distinguish between two destinations with the same IP address using information stored in the VRF.

Configuring Scalable Hub-and-Spoke MPLS VPNs

This module explains how to ensure that virtual private network (VPN) clients that connect to the same provider edge (PE) router at the edge of the Mutliprotocol (MPLS) Virtual Private Network (VPN) use the hub site. This feature prevents the VPN clients from communicating directly with each other, bypassing the hub site. This feature also provides scalable hub-and-spoke connectivity for subscribers of an MPLS VPN service by removing the requirement of one VRF per spoke.

- Finding Feature Information, page 67
- Prerequisites for Configuring Scalable Hub-and-Spoke MPLS VPNs, page 67
- Restrictions for Configuring Scalable Hub-and-Spoke MPLS VPNs, page 68
- Information about Configuring Scalable Hub-and-Spoke MPLS VPNs, page 68
- How to Ensure that MPLS VPN Clients Use the Hub PE Router, page 69
- Configuration Examples for Configuring Scalable Hub-and-Spoke MPLS VPNs, page 75
- Additional References, page 78
- Feature Information for Configuring Scalable Hub-and-Spoke MPLS VPNs, page 79

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring Scalable Hub-and-Spoke MPLS VPNs

You must have a working MPLS core network.

Restrictions for Configuring Scalable Hub-and-Spoke MPLS VPNs

- In both the upstream and downstream VRFs, routing protocols are not supported on interfaces configured with this feature. Interfaces that are not configured with this feature, however, do not have this restriction for the upstream or downstream VRFs.
- You can configure this feature only on virtual access interfaces (VAIs) and virtual template interfaces (VTIs).
- Only unnumbered interfaces are supported.
- Multicast is not supported on interfaces configured for hub-and-spoke MPLS VPNs.

Information about Configuring Scalable Hub-and-Spoke MPLS VPNs

- Overview, page 68
- Upstream and Downstream VRFs, page 69
- Reverse Path Forwarding Check, page 69

Overview

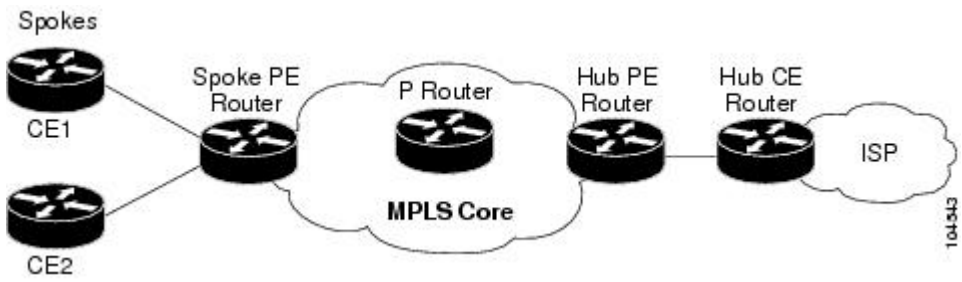
This feature prevents local connectivity between subscribers at the spoke provider edge (PE) router and ensures that a hub site provides subscriber connectivity. Any sites that connect to the same PE router must forward intersite traffic using the hub site. This ensures that the routing done at the spoke site moves from the access-side interface to the network-side interface or from the network-side interface to the access-side interface, but never from the access-side interface to the access-side interface.

This feature prevents situations where the PE router locally switches the spokes without passing the traffic through the hub site. This prevents subscribers from directly connecting to each other.

This feature eases configuration by removing the requirement of one VRF per spoke. In prior releases, when spokes connected to the same PE router, each spoke was configured in a separate VRF to ensure that the traffic between the spokes traversed the central link between the wholesale service provider and the ISP. However, this solution was not scalable. When many spokes connected to the same PE router, configuration of VRFs for each spoke became quite complex and greatly increased memory usage. This was especially true in large-scale environments that supported high-density remote access to Layer 3 VPNs.

The figure below shows a sample hub-and-spoke topology.

Figure 3 Hub-and-Spoke Topology



Upstream and Downstream VRFs

This feature uses two unidirectional VRFs to forward IP traffic between the spokes and the hub PE router:

- The upstream VRF forwards the IP traffic from the spokes toward the hub PE router. This VRF typically contains only a default route but might also contain summary routes and multiple default routes. The default route points to the interface on the hub PE router that connects to the upstream ISP. The router dynamically learns about the default route from the routing updates that the hub PE router or home gateway sends. The upstream VRF also contains the VAIs that connect the spokes, but it contains no other local interfaces.
- The downstream VRF forwards traffic from the hub PE router back to the spokes. This VRF contains Point-to-Point Protocol (PPP) peer routes for the spokes and per-user static routes received from the Authentication, Authorization, and Accounting (AAA) server. It also contains the routes imported from the hub PE router.

The router redistributes routes from the downstream VRF into Multiprotocol Border Gateway Protocol (MP-BGP). The spoke PE router typically advertises a summary route across the MPLS core for the connected spokes. The VRF configured on the hub PE router imports the advertised summary route.

Reverse Path Forwarding Check

The unicast Reverse Path Forwarding (RPF) check ensures that an IP packet that enters a router uses the correct inbound interface. This feature supports unicast RPF check on the spoke-side interfaces. Because different VRFs are used for downstream and upstream forwarding, the RPF mechanism ensures that source address checks occur in the downstream VRF.

How to Ensure that MPLS VPN Clients Use the Hub PE Router

- Configuring the Upstream and Downstream VRFs on the PE Router or the Spoke PE Router, page 69
- Associating VRFs, page 71
- Configuring the Downstream VRF for an AAA Server, page 72
- Verifying the Configuration, page 72

Configuring the Upstream and Downstream VRFs on the PE Router or the Spoke PE Router

To configure the upstream and downstream VRFs on the PE router or on the spoke PE router, use the following procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
6. **exit**

DETAILED STEPS

	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 vrf** *vrf-name* **Example:** `Router(config)# ip vrf U`	Enters VRF configuration mode and defines the VRF instance by assigning a VRF name.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 1:0`	Creates routing and forwarding tables.
Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf)# route-target import 1:0`	Creates a list of import and export route target communities for the specified VRF. • The **import** keyword is required to create an upstream VRF. The upstream VRF is used to import the default route from the hub PE router. • The **export** keyword is required to create a downstream VRF. The downstream VRF is used to export the routes of all subscribers of a given service that the VRF serves.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-vrf)# exit`	Returns to global configuration mode.

Associating VRFs

The virtual template interface is used to create and configure a virtual access interface (VAI). After you define and configure the VRFs on the PE routers, associate each VRF with the following:

- Interface or subinterface
- Virtual template interface

To associate a VRF, enter the following commands on the PE router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number*
4. **ip vrf forwarding** *vrf-name1* [**downstream** *vrf-name2*]
5. **ip unnumbered** *type number*
6. **exit**

DETAILED STEPS

	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 virtual-template** *number* **Example:** `Router(config)# interface virtual-template 1`	Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces. Enters interface configuration mode.
Step 4	**ip vrf forwarding** *vrf-name1* [**downstream** *vrf-name2*] **Example:** `Router(config-if)# ip vrf forwarding vpn1 downstream D`	Associates a virtual template interface with the VRF you specify. • The *vrf-name1* argument is the name of the VRF associated with the virtual template interface. • The *vrf-name2* argument is the name of the downstream VRF into which the PPP peer route and all of the per-user routes from the AAA server are installed. If an AAA server is used, it provides the VRF membership; you do not need to configure the VRF members on the virtual templates.
Step 5	**ip unnumbered** *type number* **Example:** `Router(config-if)# ip unnumbered Loopback1`	Enables IP processing on an interface without assigning an explicit IP address to the interface. The *type* and *number* arguments are the type and number of another interface on which the router has an assigned IP address. It cannot be another unnumbered interface.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.

Configuring the Downstream VRF for an AAA Server

To configure the downstream VRF for an AAA server, enter the following Cisco attribute value:

lcp:interface-config=ip vrf forwarding U downstream D

For more information about configuring a RADIUS server, see Configuring Virtual Template Interfaces.

Verifying the Configuration

To verify the configuration, perform the following steps.

SUMMARY STEPS

1. **show ip vrf** [**brief** | **detail** | **interfaces** | **id**] [*vrf-name*]
2. **show ip route vrf** *vrf-name*
3. **show running-config** [**interface** *type number*]

DETAILED STEPS

Step 1 **show ip vrf** [**brief** | **detail** | **interfaces** | **id**] [*vrf-name*]

Use this command to display information about all of the VRFs configured on the router, including the downstream VRF for each associated VAI.

Example:

```
Router# show ip vrf

  Name   Default RD   Interface
  D      2:0          Loopback2
                      Virtual-Access3 [D]

                      Virtual-Access4 [D]


  U      2:1          Virtual-Access3
                      Virtual-Access4
```

show ip vrf detail *vrf-name*

Use this command to display detailed information about the VRF you specify, including all of the VAIs associated with the VRF.

If you do not specify a value for *vrf-name*, detailed information about all of the VRFs configured on the router appears, including all of the VAIs associated with each VRF.

The following example shows how to display detailed information for the VRF called vrf1.

Example:

```
Router# show ip vrf detail vrf1
VRF D; default RD 2:0; default VPNID <not set>
  Interfaces:
         Loopback2           Virtual-Access3 [D]  Virtual-Access4 [D]
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:2:0
  Import VPN route-target communities
    RT:2:1
  No import route-map
  No export route-map
VRF U; default RD 2:1; default VPNID <not set>
  Interfaces:
    Virtual-Access3          Virtual-Access4
  Connected addresses are not in global routing table
  No Export VPN route-target communities
  Import VPN route-target communities
    RT:2:1
  No import route-map
  No export route-map
```

Step 2 **show ip route vrf** *vrf-name*

Use this command to display the IP routing table for the VRF you specify, and information about the per-user static routes installed in the downstream VRF.

The following example shows how to display the routing table for the downstream VRF named D.

Example:

```
Router# show ip route vrf D

Routing Table: D
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is not set
     2.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
U       2.0.0.2/32 [1/0] via 2.8.1.1
S       2.0.0.0/8 is directly connected, Null0
U       2.0.0.5/32 [1/0] via 2.8.1.2
C       2.8.1.2/32 is directly connected, Virtual-Access4
C       2.8.1.1/32 is directly connected, Virtual-Access3
```

The following example shows how to display the routing table for the upstream VRF named U.

Example:

```
Router# show ip route vrf U
Routing Table: U
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS interarea
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is 100.0.0.20 to network 0.0.0.0
     2.0.0.0/32 is subnetted, 1 subnets
C       2.0.0.8 is directly connected, Loopback2
B*   0.0.0.0/0 [200/0] via 100.0.0.20, 1w5d
```

Step 3 **show running-config** [**interface** *type number*]

Use this command to display information about the virtual access interface you specify, including information about the upstream and downstream VRFs.

The following example shows how to display information about the interface named virtual-access 3.

Example:

```
Router# show running-config interface virtual-access 3
Building configuration...
Current configuration : 92 bytes
!
interface Virtual-Access3
 ip vrf forwarding U downstream D
 ip unnumbered Loopback2
end
```

The following example shows how to display information about the interface named virtual-access 4.

Example:

```
Router# show running-config interface virtual-access 4
Building configuration...
Current configuration : 92 bytes
!
```

```
interface Virtual-Access4
 ip vrf forwarding U downstream D
 ip unnumbered Loopback2
end
```

Configuration Examples for Configuring Scalable Hub-and-Spoke MPLS VPNs

- Configuring the Upstream and Downstream VRFs on the PE Router and the Spoke PE Router Example, page 75
- Associating VRFs Example, page 75
- Configuring Scalable Hub-and-Spoke MPLS VPNs--Basic Configuration Example, page 76
- Example, page 77

Configuring the Upstream and Downstream VRFs on the PE Router and the Spoke PE Router Example

The following example configures an upstream VRF named U:

```
Router> enable
Router# configure terminal
Router(config)# ip vrf U
Router(config-vrf)# rd 1:0
Router(config-vrf)# route-target import 1:0
```

The following example configures a downstream VRF named D:

```
Router> enable
Router# configure terminal
Router(config)# ip vrf D
Router(config-vrf)# rd 1:8

Router(config-vrf)# route-target export 1:100
```

Associating VRFs Example

The following example associates the VRF named U with the virtual-template 1 interface and specifies the downstream VRF named D:

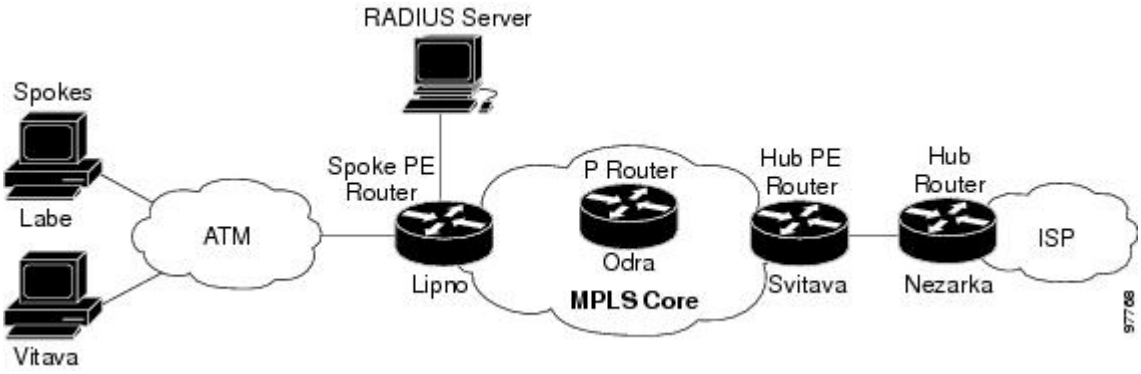
```
Router> enable
Router# configure terminal
Router(config)# interface virtual-template 1
Router(config-if)# ip vrf forwarding U downstream D
Router(config-if)# ip unnumbered Loopback1
```

Configuring Scalable Hub-and-Spoke MPLS VPNs--Basic Configuration Example

In this example, local authentication is used; that is, the RADIUS server is not used.

This example uses the hub-and-spoke topology shown in the figure below.

Figure 4 Sample Topology




```
ip vrf D
 rd 1:8
 route-target export 1:100
!
ip vrf U
 rd 1:0
 route-target import 1:0
!
ip cef
vpdn enable
!
vpdn-group U
 accept-dialin
  protocol pppoe
  virtual-template 1
!
interface Loopback2
 ip vrf forwarding U
 ip address 2.0.0.8 255.255.255.255
!
interface ATM2/0
 description Mze ATM3/1/2
 no ip address
 no atm ilmi-keepalive
 pvc 0/16 ilmi
!
 pvc 3/100
  protocol pppoe
!
pvc 3/101
  protocol pppoe
!
interface Virtual-Template1
 ip vrf forwarding U downstream D
 ip unnumbered Loopback2
 peer default ip address pool U-pool
 ppp authentication chap
```

Example

The following example shows how to connect two Point-to-Point Protocol over Ethernet (PPPoE) clients to a single VRF pair on the spoke PE router named Lipno. Although both PPPoE clients are configured in the same VRF, all communication occurs using the hub PE router. Half-duplex VRFs are configured on the spoke PE. The client configuration is downloaded to the spoke PE from the RADIUS server.

This example uses the hub-and-spoke topology shown in the figure above.

Note The wholesale provider can forward the user authentication request to the corresponding ISP. If the ISP authenticates the user, the wholesale provider appends the VRF information to the request that goes back to the PE router.

```
aaa new-model
!
aaa group server radius R
 server 22.0.20.26 auth-port 1812 acct-port 1813
!
aaa authentication ppp default group radius
aaa authorization network default group radius
!
ip vrf D
 description Downstream VRF - to spokes
 rd 1:8
 route-target export 1:100
!
ip vrf U
 description Upstream VRF - to hub
 rd 1:0
 route-target import 1:0
!
ip cef
vpdn enable
!
vpdn-group U
 accept-dialin
  protocol pppoe
  virtual-template 1
!
interface Loopback2
 ip vrf forwarding U
 ip address 2.0.0.8 255.255.255.255
!
interface ATM2/0
  pvc 3/100
  protocol pppoe
 !
pvc 3/101
  protocol pppoe
 !
interface virtual-template 1
 no ip address
 ppp authentication chap
!
router bgp 1
 no synchronization
 neighbor 100.0.0.34 remote-as 1
 neighbor 100.0.0.34 update-source Loopback0
 no auto-summary
 !
address-family vpnv4
  neighbor 100.0.0.34 activate
  neighbor 100.0.0.34 send-community extended
  auto-summary
  exit-address-family
```

```
 !
address-family ipv4 vrf U
  no auto-summary
  no synchronization
  exit-address-family
!
address-family ipv4 vrf D
  redistribute static
  no auto-summary
  no synchronization
  exit-address-family
!
ip local pool U-pool 2.8.1.1 2.8.1.100
ip route vrf D 2.0.0.0 255.0.0.0 Null0
!
radius-server host 22.0.20.26 auth-port 1812 acct-port 1813
radius-server key cisco
```

Additional References

Related Documents

Related Topic	Document Title
Basic MPLS VPNs	Configuring MPLS Layer 3 VPNs
MPLS VPN Carrier Supporting Carrier	• MPLS VPN Carrier Supporting Carrier Using LDP and an IGP • MPLS VPN Carrier Supporting Carrier with BGP
MPLS VPN InterAutonomous Systems	• MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels • MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

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 2547	BGP/MPLS VPNs

Technical Assistance

Description	Link
The Cisco Technical Support 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. 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 Configuring Scalable Hub-and-Spoke MPLS VPNs

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 4 Feature Information for Configuring Scalable Hub-and-Spoke MPLS VPNs

Feature Name	Releases	Feature Configuration Information
MPLS VPN: Half Duplex VRF Support	12.3(6) 12.3(11)T	This feature ensures that VPN clients that connect to the same PE router at the edge of the MPLS VPN use the hub site to communicate.

Ensuring MPLS VPN Clients Communicate over the Backbone Links

This module describes how to configure a sham-link that ensures traffic travels between Virtual Private Network (VPN) client sites over the Multiprotocol Label Switching (MPLS) VPN backbone. This feature is for VPNs that run Open Shortest Path First (OSPF) between the provider edge (PE) and customer edge (CE) routers. By default, OSPF uses backdoor paths between VPN sites, not the MPLS VPN backbone.

- Finding Feature Information, page 81
- Prerequisites for Ensuring MPLS VPN Clients Communicate over the Backbone Links, page 81
- Restrictions for Ensuring MPLS VPN Clients Communicate over the Backbone Links, page 82
- Information About Ensuring MPLS VPN Clients Communicate over the Backbone Links, page 82
- How to Ensure That MPLS VPN Clients Communicate over the MPLS VPN Backbone, page 85
- Configuration Examples for Ensuring MPLS VPN Clients Communicate over the MPLS VPN Backbone, page 87
- Additional References, page 90
- Feature Information for Ensuring MPLS VPN Clients Communicate over the MPLS VPN Backbone, page 91

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Ensuring MPLS VPN Clients Communicate over the Backbone Links

Before you can configure a sham-link in an MPLS VPN, you must first enable OSPF as follows:

- Create an OSPF routing process.
- Specify the range of IP addresses to be associated with the routing process.
- Assign area IDs to be associated with the range of IP addresses.

Restrictions for Ensuring MPLS VPN Clients Communicate over the Backbone Links

When OSPF is used as a protocol between PE and CE routers, the OSPF metric is preserved when routes are advertised over the VPN backbone. The metric is used on the remote PE routers to select the correct route. For this reason, you should not modify the metric value when OSPF is redistributed to Border Gateway Protocol (BGP), and when BGP is redistributed to OSPF. If you modify the metric value, routing loops may occur.

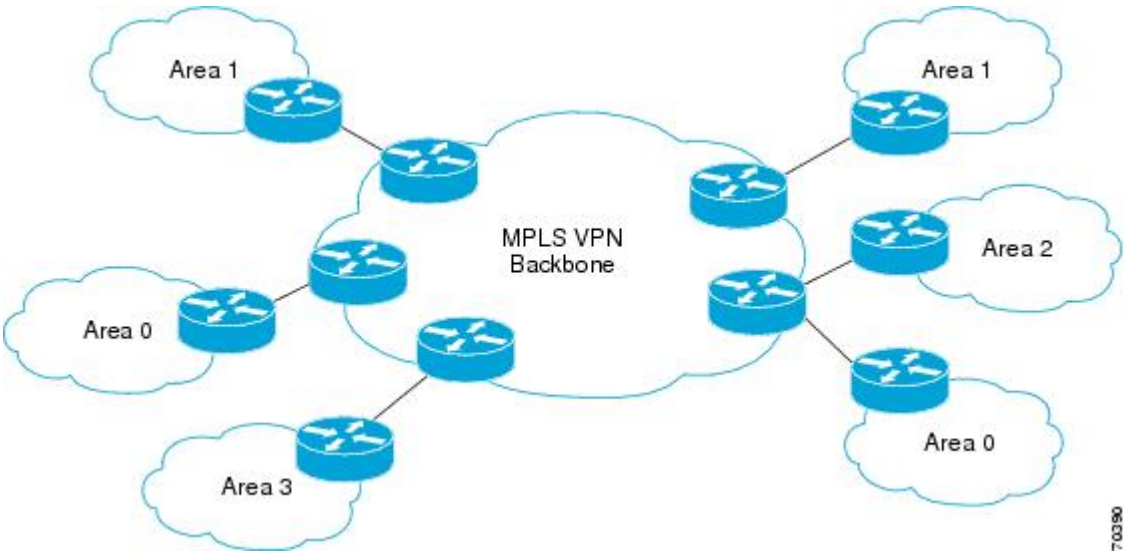
Information About Ensuring MPLS VPN Clients Communicate over the Backbone Links

- Introduction to MPLS VPNs Using OSPF Between PE and CE Routers, page 82
- OSPF Uses Backdoor Paths to Communicate Between VPN Sites, page 83
- Sham-Links Direct Traffic Between VPN Sites over the MPLS VPN Backbone, page 84

Introduction to MPLS VPNs Using OSPF Between PE and CE Routers

In an MPLS VPN configuration, the OSPF protocol is one way you can connect CE routers to PE routers in the VPN backbone. OSPF is often used by customers that run OSPF as their intrasite routing protocol, subscribe to a VPN service, and want to exchange routing information between their sites using OSPF (during migration or on a permanent basis) over an MPLS VPN backbone.

The figure below shows an example of how VPN client sites (areas 0, 1, 2, and 3) that run OSPF can connect over an MPLS VPN backbone.



When OSPF is used to connect PE and CE routers, all routing information learned from a VPN site is placed in the VPN routing and forwarding (VRF) instance associated with the incoming interface. The PE

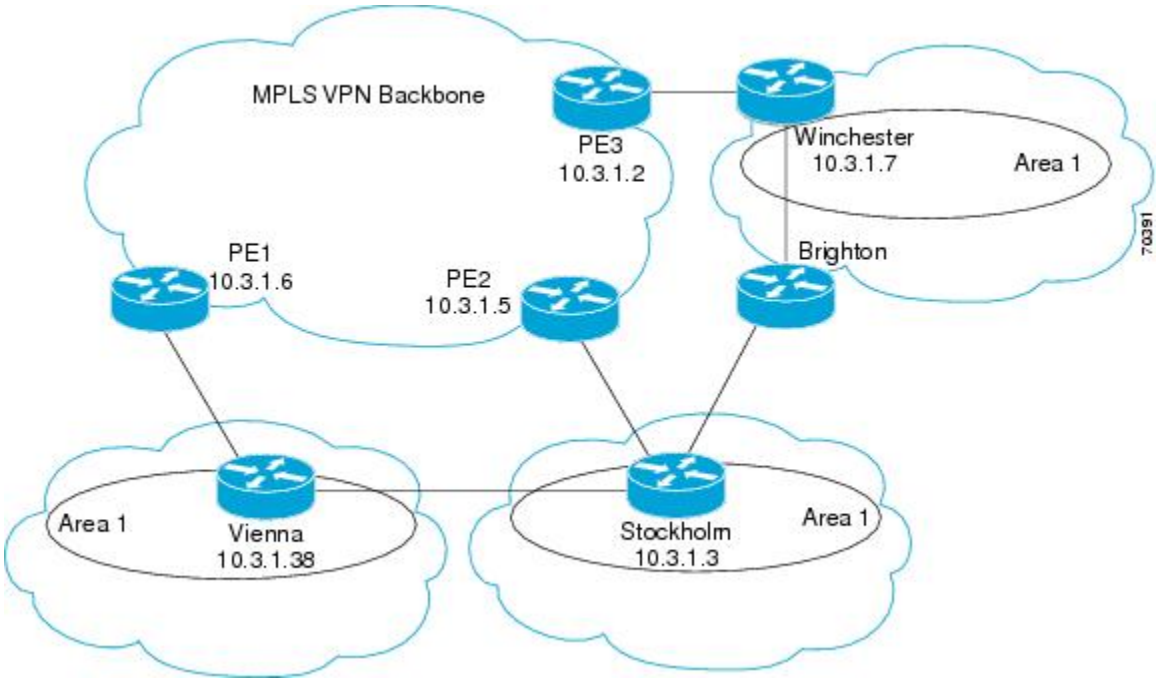
routers that attach to the VPN use the BGP to distribute VPN routes to each other. A CE router can then learn the routes to other sites in the VPN by peering with its attached PE router. The MPLS VPN backbone provides an additional level of routing hierarchy to interconnect the VPN sites running OSPF.

When OSPF routes are propagated over the MPLS VPN backbone, additional information about the prefix in the form of BGP extended communities (route type, domain ID extended communities) is appended to the BGP update. This community information is used by the receiving PE router to decide the type of link-state advertisement (LSA) to be generated when the BGP route is redistributed to the OSPF PECE process. In this way, internal OSPF routes that belong to the same VPN and are advertised over the VPN backbone are seen as interarea routes on the remote sites.

OSPF Uses Backdoor Paths to Communicate Between VPN Sites

Although OSPF PECE connections assume that the only path between two client sites is across the MPLS VPN backbone, backdoor paths between VPN sites may exist. For instance, in the figure below, Vienna, Stockholm, Brighton, and Winchester can communicate through backdoor paths instead of using the MPLS VPN backbone.

If the sites belong to the same OSPF area, the backdoor path will always be selected, because OSPF prefers intra-area paths to interarea paths. (PE routers advertise OSPF routes learned over the VPN backbone as interarea paths.) For this reason, OSPF backdoor paths between VPN sites must be taken into account so that routing is performed based on policy.



For example, the figure above shows three client sites, each with backdoor links. Because each site runs OSPF within the same Area 1 configuration, all routing between the three sites uses the backdoor paths, rather than the MPLS VPN backbone.

The following example shows BGP routing table entries for the Winchester router (prefix 10.3.1.7/32) from the standpoint of the PE1 router in the figure. Prefix 10.3.1.7 is the loopback interface of the Winchester

CE router. As shown in bold in this example, the loopback interface is learned via BGP from PE2 and PE3. It is also generated through redistribution into BGP on PE1.

```
PE1# show ip bgp vpnv4 all 10.3.1.7
BGP routing table entry for 100:251:10.3.1.7/32, version 58
Paths: (3 available, best #2)
  Advertised to non peer-group peers:
  10.3.1.2 10.3.1.5
  Local
    10.3.1.5 (metric 30) from 10.3.1.5 (10.3.1.5)
      Origin incomplete, metric 22, localpref 100, valid, internal
      Extended Community: RT:1:793 OSPF DOMAIN ID:0.0.0.100 OSPF
      RT:1:2:0 OSPF 2
  Local
    10.2.1.38 from 0.0.0.0 (10.3.1.6)
      Origin incomplete, metric 86, localpref 100, weight 32768,
      valid, sourced, best
      Extended Community: RT:1:793 OSPF DOMAIN ID:0.0.0.100 OSPF
      RT:1:2:0 OSPF 2
  Local
    10.3.1.2 (metric 30) from 10.3.1.2 (10.3.1.2)
      Origin incomplete, metric 11, localpref 100, valid, internal
      Extended Community: RT:1:793 OSPF DOMAIN ID:0.0.0.100 OSPF
      RT:1:2:0 OSPF 2
```

Within BGP, the locally generated route (10.2.1.38) is considered to be the best route.

However, as shown in bold in the next example, the VRF routing table shows that the selected path is learned via OSPF with a next hop of 10.2.1.38, which is the Vienna CE router.

```
PE1# show ip route vrf ospf 10.3.1.7
Routing entry for 10.3.1.7/32
  Known via "ospf 100", distance 110, metric 86, type intra area
  Redistributing via bgp 215
  Advertised by bgp 215
  Last update from 10.2.1.38 on Serial0/0/0, 00:00:17 ago
  Routing Descriptor Blocks:
  * 10.2.1.38
, from 10.3.1.7, 00:00:17 ago, via Serial0/0/0
      Route metric is 86, traffic share count is 1
```

This path is selected because:

- The OSPF backdoor path is preferred over the interarea path (over the MPLS VPN backbone) generated by the PE1 router.
- OSPF has a lower administrative distance (AD) than internal BGP (BGP running between routers in the same autonomous system).

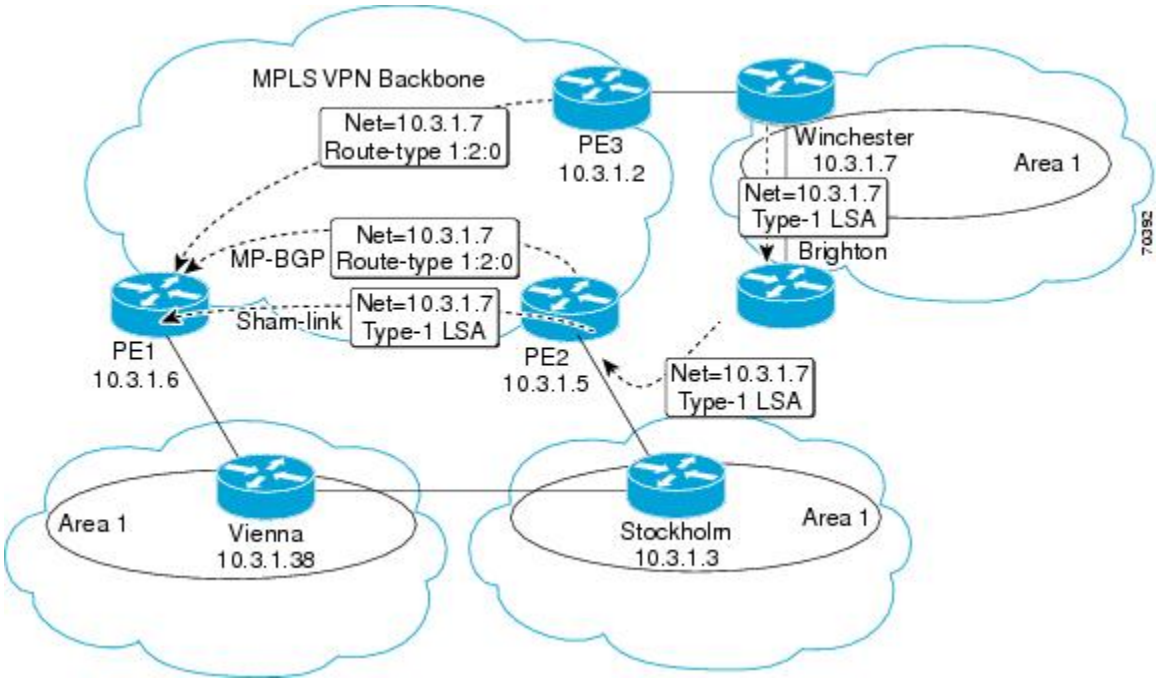
If the backdoor paths between sites are used only for backup purposes and do not participate in the VPN service, then the default route selection is acceptable. You can set up the OSPF cost configured with a sham-link to send VPN site traffic over a backdoor path.

Sham-Links Direct Traffic Between VPN Sites over the MPLS VPN Backbone

To ensure that VPN sites that belong to the same OSPF area and share an OSPF backdoor path communicate with each other using the MPLS VPN backbone, you must create a sham-link. (If no backdoor path exists between the sites, no sham-link is required.) A sham-link is an additional OSPF intra-area (logical) link between ingress and egress VRFs on the PE routers that connect to the CE routers of the VPN sites.

The figure below shows a sample sham-link between PE1 and PE2. You associate a cost with each sham-link to force traffic to use the sham-link rather than the backdoor path. When a sham-link is configured

between PE routers, the PE routers can populate the VRF routing table with the OSPF routes learned over the sham-link.



Because the sham-link is seen as an intra-area link between PE routers, an OSPF adjacency is created and database exchange (for the particular OSPF process) occurs across the link. The PE router can then flood LSAs between sites from across the MPLS VPN backbone. As a result, the desired intra-area connectivity is created.

How to Ensure That MPLS VPN Clients Communicate over the MPLS VPN Backbone

This section explains how to create a sham-link on an MPLS VPN PE router. Perform this task on both PE routers that share the sham-link.

Before you create a sham-link between PE routers in an MPLS VPN, you must:

- Configure a separate /32 address on the remote PE so that OSPF packets can be sent over the VPN backbone to the remote end of the sham-link. The /32 address must meet the following criteria:
  - Belong to a VRF.
  - Not be advertised by OSPF.
  - Be advertised by BGP.

You can use the /32 address for other sham-links.

- Associate the sham-link with an existing OSPF area.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *interface-number*
4. **ip vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask*
6. **end**
7. **router ospf** *process-id***vrf** *vrf-name*
8. **area** *area-id* **sham-link** *source-address destination-address* **cost** *number*
9. **show ip ospf sham-links**

DETAILED STEPS

	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 loopback** *interface-number* **Example:** `Router(config)# interface loopback 1`	Creates a loopback interface to be used as an endpoint of the sham-link on the PE router and enters interface configuration mode.
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding ospf`	Associates the loopback interface with a VRF. Removes the IP address.
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.2.1.2 255.255.255.255`	Reconfigures the IP address of the loopback interface on the PE router.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config-if)# end`	Returns to global configuration mode.
Step 7	**router ospf** *process-id***vrf** *vrf-name* **Example:** `Router(config)# router ospf 100 vrf ospf`	Configures the specified OSPF process with the VRF associated with the sham-link interface on the PE router and enters interface configuration mode.
Step 8	**area** *area-id* **sham-link** *source-address destination-address* **cost** *number* **Example:** `Router(config-if)# area 1 sham-link 10.2.1.2 10.2.1.1 cost 40`	Configures the sham-link on the PE router interface within a specified OSPF area and with the loopback interfaces specified by the IP addresses as endpoints. • **cost** *number* configures the OSPF cost for sending an IP packet over the PE sham-link interface.
Step 9	**show ip ospf sham-links** **Example:**	Verifies that the sham-link was successfully created and is operational.

Example

The following is sample output from the **show ip ospf sham-links** command:

```
Router# show ip ospf sham-links
Sham Link OSPF_SL0 to address 10.2.1.2 is up
Area 1 source address 10.2.1.1
  Run as demand circuit
  DoNotAge LSA allowed.
  Cost of using 40 State POINT_TO_POINT,
  Timer intervals configured,
  Hello 10, Dead 40, Wait 40,
    Hello due in 00:00:04
    Adjacency State FULL (Hello suppressed)
    Index 2/2, retransmission queue length 4,     number of retransmission 0
    First 0x63311F3C(205)/0x63311FE4(59) Next
    0x63311F3C(205)/0x63311FE4(59)
    Last retransmission scan length is 0,     maximum is 0
    Last retransmission scan time is 0 msec,     maximum is 0 msec
    Link State retransmission due in 360 msec
```

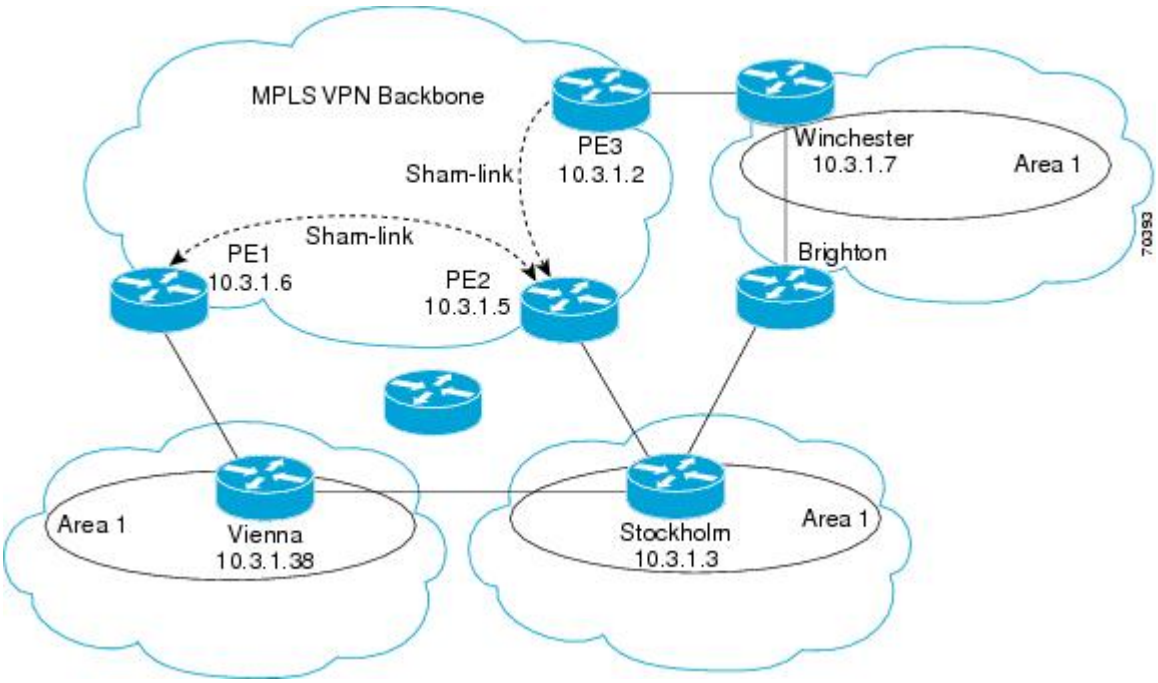
Configuration Examples for Ensuring MPLS VPN Clients Communicate over the MPLS VPN Backbone

The following example shows how to configure a sham-link between two PE routers:

```
Router1(config)# interface loopback 1
Router1(config-if)# ip vrf forwarding ospf
Router1(config-if)# ip address 10.2.1.1 255.255.255.255
!
Router2(config)# interface loopback 1
Router2(config-if)# ip vrf forwarding ospf
Router2(config-if)# ip address 10.2.1.2 255.255.255.255
!
Router1(config)# router ospf 100 vrf ospf
Router1(config-if)# area 1 sham-link 10.2.1.1 10.2.1.2 cost 40
!
Router2(config)# router ospf 100 vrf ospf
Router2(config-if)# area 1 sham-link 10.2.1.2 10.2.1.1 cost 40
```

This example shows how a sham-link is used only to affect the OSPF intra-area path selection of the PE and CE routers. The PE router also uses the information received from Multiprotocol BGP (MP-BGP) to set the outgoing label stack of incoming packets, and to decide to which egress PE router to label-switch the packets.

The figure below shows a sample MPLS VPN topology in which a sham-link configuration is necessary. A VPN client has three sites, each with a backdoor path. Two sham-links have been configured, one between PE1 and PE2, and another between PE2 and PE3. A sham-link between PE1 and PE3 is not necessary in this configuration, because the Vienna and Winchester sites do not share a backdoor path.



The following example shows the forwarding that occurs between sites from the standpoint of how PE1 views the 10.3.1.7/32 prefix, the loopback1 interface of the Winchester CE router in the figure above.

```
PE1# show ip bgp vpnv4 all 10.3.1.7
BGP routing table entry for 100:251:10.3.1.7/32, version 124
Paths: (1 available, best #1)
  Local
    10.3.1.2 (metric 30) from 10.3.1.2
 (10.3.1.2)
       Origin incomplete, metric 11, localpref 100, valid, internal,
```

```
       best
       Extended Community: RT:1:793 OSPF DOMAIN ID:0.0.0.100 OSPF
       RT:1:2:0 OSPF 2
PE1# show ip route vrf ospf 10.3.1.7
Routing entry for 10.3.1.7/32
  Known via "ospf 100
", distance 110, metric 13, type intra area
  Redistributing via bgp 215
  Last update from 10.3.1.2 00:12:59 ago
  Routing Descriptor Blocks:
10.3.1.2 (Default-IP-Routing-Table), from 10.3.1.7, 00:12:59 ago
```

The next example shows forwarding information in which the next hop for the route, 10.3.1.2, is the PE3 router rather than the PE2 router (which is the best path according to OSPF). The OSPF route is not redistributed to BGP on the PE, because the other end of the sham-link already redistributed the route to BGP and there is no need for duplication. The OSPF sham-link is used only to influence intra-area path selection. When sending traffic to a particular destination, the PE router uses the MP-BGP forwarding information.

```
PE1# show ip bgp vpnv4 all tag | begin 10.3.1.7
   10.3.1.7/32      10.3.1.2
        notag/38

PE1# show mpls forwarding 10.3.1.2
Local  Outgoing      Prefix           Bytes label    Outgoing   Next Hop
label  label or VC   or Tunnel Id     switched       interface
31     42            10.3.1.2/32
      0              PO3/0/0    point2point
PE1# show ip cef vrf ospf 10.3.1.7
10.3.1.7/32, version 73, epoch 0, cached adjacency to POS3/0/0
0 packets, 0 bytes
  tag information set
   local tag: VPN-route-head
   fast tag rewrite with PO3/0/0, point2point, tags imposed: {42 38
}
  via 10.3.1.2
, 0 dependencies, recursive
    next hop 10.1.1.17, POS3/0/0 via 10.3.1.2/32
    valid cached adjacency
    tag rewrite with PO3/0/0, point2point, tags imposed: {42 38}
```

If a prefix is learned across the sham-link and the path via the sham-link is selected as the best, the PE router does not generate an MP-BGP update for the prefix. It is not possible to route traffic from one sham-link over another sham-link.

In the following example, PE2 shows how an MP-BGP update for the prefix is not generated. Although 10.3.1.7/32 has been learned via OSPF across the sham-link as shown in bold, no local generation of a route into BGP is performed. The only entry within the BGP table is the MP-BGP update received from PE3 (the egress PE router for the 10.3.1.7/32 prefix).

```
PE2# show ip route vrf ospf 10.3.1.7
Routing entry for 10.3.1.7/32
  Known via "ospf 100
", distance 110, metric 12, type intra area
  Redistributing via bgp 215
  Last update from 10.3.1.2 00:00:10 ago
  Routing Descriptor Blocks:
  * 10.3.1.2 (Default-IP-Routing-Table), from 10.3.1.7, 00:00:10 ago
      Route metric is 12, traffic share count is 1
PE2# show ip bgp vpnv4 all 10.3.1.7
BGP routing table entry for 100:251:10.3.1.7/32, version 166
Paths: (1 available, best #1)
  Not advertised to any peer
  Local
    10.3.1.2 (metric 30) from 10.3.1.2 (10.3.1.2)
      Origin incomplete, metric 11, localpref 100, valid, internal,
      best
```

```
Extended Community: RT:1:793 OSPF DOMAIN ID:0.0.0.100 OSPF
RT:1:2:0 OSPF 2
```

The PE router uses the information received from MP-BGP to set the ongoing label stack of incoming packets, and to decide to which egress PE router to label-switch the packets.

Additional References

Related Documents

Related Topic	Document Title
Basic MPLS VPNs	Configuring MPLS Layer 3 VPNs
MPLS VPN Carrier Supporting Carrier	• MPLS VPN Carrier Supporting Carrier Using LDP and an IGP • MPLS VPN Carrier Supporting Carrier with BGP
MPLS VPN InterAutonomous Systems	• MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels • MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

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 software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1164	Application of the Border Gateway Protocol in the Internet

RFC	Title
RFC 1171	A Border Gateway Protocol 4
RFC 1700	Assigned Numbers
RFC 1966	BGP Route Reflection: An Alternative to Full Mesh IBGP
RFC 2283	Multiprotocol Extensions for BGP-4
RFC 2328	Open Shortest Path First, Version 2
RFC 2547	BGP/MPLS VPNs
RFC 2842	Capabilities Advertisement with BGP-4
RFC 2858	Multiprotocol Extensions for BGP-4
RFC 3107	Carrying Label Information in BGP-4

Technical Assistance

Description	Link
The Cisco Technical Support 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. 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 Ensuring MPLS VPN Clients Communicate over the MPLS VPN Backbone

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 5 Feature Information for Ensuring MPLS VPN Clients Communicate over the MPLS VPN Backbone

Feature Name	Releases	Feature Configuration Information
Ensuring MPLS VPN Clients Communicate over the MPLS VPN Backbone	12.2(8)T 12.0(21)ST 12.0(22)S	This feature allows you to configure a sham-link that directs traffic between Virtual Private Network (VPN) client sites over the Multiprotocol Label Switching (MPLS) VPN backbone.

Assigning an ID Number to a VPN

You can identify Virtual Private Networks (VPNs) by a VPN identification number, as described in RFC 2685. This implementation of the VPN ID feature is used for identifying a VPN.

- Finding Feature Information, page 93
- Information About VPN ID, page 93
- How to Configure a VPN ID, page 95
- Additional References, page 97
- Feature Information for Assigning an ID Number to a VPN, page 99

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About VPN ID

- Introduction to VPN ID, page 93
- Components of the VPN ID, page 94
- Management Applications That Use VPN IDs, page 94

Introduction to VPN ID

You can identify VPNs by a VPN identification number, as described in RFC 2685. This implementation of the VPN ID feature is used for identifying a VPN. The VPN ID feature is not used to control the distribution of routing information or to associate IP addresses with VPN ID numbers in the MP-BGP VPNv4 routing updates.

Multiple VPNs can be configured in a router. A VPN is private and uses a private address space that might also be used by another VPN or by the Internet. The IP address used in a VPN is only significant to the VPN in which it exists. You can use a VPN name (a unique ASCII string) to reference a specific VPN configured in the router. Alternately, you can use a VPN ID to identify a particular VPN in the router. The

VPN ID follows a standard specification (RFC 2685). To ensure that the VPN has a consistent VPN ID, assign the same VPN ID to all the routers in the service provider network that services that VPN.

Note Configuration of a VPN ID for a VPN is optional. You can still use a VPN name to identify configured VPNs in the router. The VPN name is not affected by the VPN ID configuration. These are two independent mechanisms to identify VPNs.

Components of the VPN ID

Each VPN ID defined by RFC 2685 consists of the following elements:

- An Organizational Unique Identifier (OUI), a three-octet hex number: The IEEE Registration Authority assigns OUIs to any company that manufactures components under the ISO/IEC 8802 standard. The OUI is used to generate universal LAN MAC addresses and protocol identifiers for use in local and metropolitan area network applications. For example, an OUI for Cisco Systems is 00-03-6B (hex).
- A VPN index: a four-octet hex number, which identifies the VPN within the company.

Use the following **vpn id** command and specify the VPN ID:

vpn id *oui*:*vpn-index*

A colon separates the OUI from the VPN index.

Management Applications That Use VPN IDs

You can use several applications to manage VPNs by VPN ID. Remote access applications, such as the Remote Authentication Dial-In User Service (RADIUS) and Dynamic Host Configuration Protocol (DHCP), can use the VPN ID feature to identify a VPN. RADIUS can use the VPN ID to assign dial-in users to the proper VPN, based on each user's authentication information.

- Dynamic Host Configuration Protocol, page 94
- Remote Authentication Dial-In User Service, page 94

Dynamic Host Configuration Protocol

Using DHCP network administrators can centrally manage and automate the assignment of IP addresses in an organization's network. The DHCP application uses the VPN ID as follows:

1. A VPN DHCP client requests a connection to a provider edge (PE) router from a VRF interface.
2. The PE router determines the VPN ID associated with that interface.
3. The PE router sends a request with the VPN ID and other information for assigning an IP address to the DHCP server.
4. The DHCP server uses the VPN ID and IP address information to process the request.
5. The DHCP server sends a response back to the PE router, allowing the VPN DHCP client access to the VPN.

Remote Authentication Dial-In User Service

A RADIUS server (or daemon) provides authentication and accounting services to one or more client network access servers (NASs). RADIUS servers authenticate users and return all configuration information necessary for the client to deliver service to the users.

Typically, a user login consists of a query (Access-Request) from the NAS to the RADIUS server and a corresponding response (Access-Accept or Access-Reject) from the server.

- The Access-Request packet contains the username, encrypted password, NAS IP address, VPN ID, and port. The format of the request also provides information on the type of session that the user wants to initiate.
- The RADIUS server returns an Access-Accept response if it finds the username and verifies the password. The response includes a list of attribute-value pairs that describe the parameters to be used for this session. If the user is not authenticated, an Access-Reject is sent by the RADIUS server and access is denied.

How to Configure a VPN ID

- Specifying a VPN ID, page 95
- Verifying the VPN ID Configuration, page 96

Specifying a VPN ID

Use this procedure to specify a VPN ID.

- Restrictions, page 95

Restrictions

The VPN ID feature is not used to control the distribution of routing information or to associate IP addresses with VPN ID numbers in the MP-BGP VPNv4 routing updates.

Each VRF configured on a PE router can have a VPN ID configured. Configure all the PE routers that belong to the same VPN with the same VPN ID. Make sure the VPN ID is unique to the service provider network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **vpn id** *oui*:*vpn-index* :

DETAILED STEPS

	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 vrf** *vrf-name* **Example:** `Router(config)# ip vrf vrf1`	Creates a VRF routing table and a CEF forwarding table and enters VRF configuration mode. • *vrf-name* --Name assigned to a VRF.
Step 4	**vpn id** *oui*:*vpn-index* : **Example:** `Router(config-vrf)# vpn id a1:3f6c`	Assigns the VPN ID to the VRF. • *oui* :--An organizationally unique identifier. The IEEE organization assigns this identifier to companies. The OUI is restricted to three octets. • *vpn-index*--This value identifies the VPN within the company. This VPN index is restricted to four octets.

Verifying the VPN ID Configuration

To verify the VPN ID configuration, perform the following steps.

SUMMARY STEPS

1. **show ip vrf**
2. **show ip vrf id**
3. **show ip vrf detail**

DETAILED STEPS

Step 1 **show ip vrf**

Use this command to display information about the VRF tables on the PE router. This example displays three VRF tables called vpn1, vpn2, and vpn5.

Example:

```
Router# show ip vrf
```

```
  Name                             Default RD          Interfaces
  vpn1                             100:1               Ethernet1/1
                                                       Ethernet1/4
  vpn2                             <not set>
  vpn5                             500:1               Loopback2
```

Step 2 **show ip vrf id**

Use this command to ensure that the PE router contains the VPN ID you specified. The following example shows that only VRF tables vpn1 and vpn2 have VPN IDs assigned. The VRF table called vpn5 is not displayed, because it does not have a VPN ID.

Example:

```
Router# show ip vrf id
VPN Id          Name                             RD
2:3             vpn2                             <not set>
A1:3F6C         vpn1                             100:1
```

Step 3 **show ip vrf detail**

Use this command to see all the VRFs on a PE router. This command displays all the VPN IDs that are configured on the router, their associated VRF names, and VRF route distinguishers (RDs). If a VRF table in the PE router has not been assigned a VPN ID, that VRF entry is not included in the output.

Example:

```
Router# show ip vrf detail
VRF vpn1; default RD 100:1; default VPNID A1:3F6C
  Interfaces:
    Ethernet1/1             Ethernet1/4
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:100:1
  Import VPN route-target communities
    RT:100:1                RT:500:1
  No import route-map
  No export route-map
VRF vpn2; default RD <not set>; default VPNID 2:3
  No interfaces
  Connected addresses are not in global routing table
  No Export VPN route-target communities
  No Import VPN route-target communities
  No import route-map
  No export route-map
VRF vpn5; default RD 500:1; default VPNID <not set>
  Interfaces:
```

Additional References

Related Documents

Related Topic	Document Title
Basic MPLS VPNs	Configuring MPLS Layer 3 VPNs

Related Topic	Document Title
MPLS VPN Carrier Supporting Carrier	• MPLS VPN Carrier Supporting Carrier Using LDP and an IGP • MPLS VPN Carrier Supporting Carrier with BGP
MPLS VPN InterAutonomous Systems	• MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels • MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

Standards

Standard	Title
IEEE Std 802-1990	IEEE Local and Metropolitan Area Networks: Overview and Architecture

MIBs

MIB	MIBs Link
None	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 2685	Virtual Private Networks Identifier

Technical Assistance

Description	Link
The Cisco Technical Support 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. 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 Assigning an ID Number to a VPN

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 6 Feature Information for Assigning an ID Number to a VPN

Feature Name	Releases	Feature Configuration Information
VPN ID	12.0(17)ST 12.2(4)B 12.2(8)T 12.2(14)S	This feature lets you you identify VPNs by a VPN identification number, as described in RFC 2685.

Directing MPLS VPN Traffic Using Policy-Based Routing

This module explains how to configure policy-based routing (PBR) to classify and forward Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) traffic based on multiple VPN routing and forwarding (VRF) selection match criteria.

- Finding Feature Information, page 101
- Prerequisites for Directing MPLS VPN Traffic Using Policy-Based Routing, page 101
- Restrictions for Directing MPLS VPN Traffic Using Policy-Based Routing, page 102
- Information About Directing MPLS VPN Traffic Using Policy-Based Routing, page 102
- How to Configure Policy-Based Routing To Direct MPLS VPN Traffic, page 103
- Configuration Examples for Directing MPLS VPN Traffic Using Policy-Based Routing, page 111
- Additional References, page 112
- Feature Information for Directing MPLS VPN Traffic Using Policy-Based Routing, page 114

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Directing MPLS VPN Traffic Using Policy-Based Routing

- Multiprotocol BGP (MP-BGP), Multiprotocol Label Switching (MPLS), Cisco Express Forwarding (CEF), and MPLS VPNs must be enabled in your network.
- The router must be running Cisco IOS software that supports policy-based routing (PBR).
- A VRF must be defined prior to the configuration of this feature. An error message is displayed in the console if no VRF exists.

Restrictions for Directing MPLS VPN Traffic Using Policy-Based Routing

- VRF Select is supported only in Service Provider (-p-) images.
- This feature can coexist with features that use VRF selection based on the source IP address, but these features cannot be configured together on the same interface. This is designed behavior to prevent VRF table selection conflicts that could occur if these features were misconfigured together. The console returns an error message if you attempt to configure the **ip vrf select source** and the **ip vrf policy-map** commands on the same interface.
- Protocol Independent Multicast (PIM) and multicast packets do not support PBR and cannot be configured for a source IP address that is match criteria for this feature.
- The **set vrf** command cannot be configured with the following commands in the same route map sequence:
  - **set ip default interface**
  - **set interface**
  - **set ip default next-hop**
  - **set ip next-hop**

A packet cannot be set to an interface or to a next hop when the **set vrf** command is specified. This is designed behavior. An error message is displayed if you attempt to configure the **set vrf** command with any of the above four set clauses.

- The VRF Selection using Policy Based Routing feature cannot be configured with IP prefix lists.
- If an interface is associated with a VRF by configuring the **ip vrf forwarding** interface configuration command, you cannot also configure the same interface to use PBR with the **set vrf** route map configuration command.
- PBR can be configured on an interface where a VRF is defined. However, the console displays the following warning messages if you attempt to configure both PBR and a VRF on the same interface:

```
%% Policy Based Routing is NOT supported for VRF" interfaces
%% IP-Policy can be used ONLY for marking "(set/clear DF bit) on
```

Information About Directing MPLS VPN Traffic Using Policy-Based Routing

- Directing MPLS VPN Traffic Using Policy-Based Routing Overview, page 102
- VRF Selection Introduces a New PBR Set Clause, page 103

Directing MPLS VPN Traffic Using Policy-Based Routing Overview

This feature allows you to route VPN traffic based on the following match criteria:

- IP Access Lists -- IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. All IP access list configuration options in Cisco IOS software can be used to define match criteria.

- Packet Lengths-- Length of a packet in bytes. The packet length filter is defined in a route map with the **match length** route map configuration command.

Policy routing is defined in the route map. The route map is applied to the incoming interface with the **ip policy route-map** interface configuration command. IP access list match criteria is applied to the route map with the **match ip address** route map configuration command. Packet length match criteria is applied to the route map with the **match length** route map configuration command. The set action is defined with the **set vrf** route map configuration command. The match criteria is evaluated, and the appropriate VRF is selected by the set clause. This combination allows you to define match criteria for incoming VPN traffic and policy route VPN packets out to the appropriate VRF.

VRF Selection Introduces a New PBR Set Clause

When configuring PBR, the following four set clauses can be used to change normal routing and forwarding behavior:

- **set ip default interface**
- **set ip interface**
- **set ip default next-hop**
- **set ip next-hop**

Configuring any of the above set clauses will overwrite normal routing forwarding behavior of a packet.

This feature introduces the fifth set clause that can be used to change normal routing and forwarding behavior. You can use the **set vrf** command to select the appropriate VRF after the successful match occurs in the route map. However, the **set vrf** command cannot be configured with the above four PBR set clauses. This is designed behavior, because a packet cannot be set to an interface or a specific next hop when it is configured within a VRF. An error message will be displayed in the console if you attempt to configure the **set vrf** command with any of the above four PBR set clauses within the same route map.

How to Configure Policy-Based Routing To Direct MPLS VPN Traffic

- Defining the Match Criteria, page 103
- Prerequisites, page 104
- Configuring the Route Map and Specifying VRFs, page 106
- Applying a Route Map to an Interface, page 107
- Configuring IP VRF Receive on the Interface, page 109
- Verifying the Configuration, page 110

Defining the Match Criteria

The match criteria is defined in an access list. Standard and extended access lists are supported. The following sections show how to configure each type of access list:

Match criteria can also be defined based on the packet length by configuring the **match length** route-map configuration command. You use a route map to configure VRF selection based on packet length. See the Configuring the Route Map and Specifying VRFs, page 106 for more information.

Prerequisites

The following tasks assume that the VRF and associated IP address are already defined.

- Defining Match Criteria with a Standard Access List, page 104
- Defining Match Criteria with an Extended Access List, page 104

Defining Match Criteria with a Standard Access List

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**]

DETAILED STEPS

	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* {**deny** | **permit**} *source* [*source-wildcard*] [**log**] **Example:** `Router(config)# access-list 40 192.168.1.0 0.0.0.255 permit`	Creates an access list and defines the match criteria for the route map. • Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. All IP access list configuration options in Cisco IOS software can be used to define match criteria. • The example creates a standard access list numbered 40. This filter will permit traffic from any host with an IP address in the 192.168.1.0/24 subnet.

Defining Match Criteria with an Extended Access List

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*]
4. [*sequence-number*] **permit** |**deny** *protocol source source-wildcard* **destination** *destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
5. **exit**

DETAILED STEPS

	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** {**standard** | **extended**} [*access-list-name* | *access-list-number*] **Example:** `Router(config)# ip access-list extended NAMEDACL`	Specifies the IP access list type, and enters the corresponding access list configuration mode. • A standard, extended, or named access list can be used.
Step 4	[*sequence-number*] **permit** |**deny** *protocol source source-wildcard* **destination** *destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-ext-nacl)# permit ip any any option any-options`	Defines the criteria for which the access list will permit or deny packets. • Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. All IP access list configuration options in Cisco IOS software can be used to define match criteria. • The example creates a named access list that permits any configured IP option.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-ext-nacl)# exit`	Exits named access list configuration mode, and enters global configuration mode.

Configuring the Route Map and Specifying VRFs

You define a route map then assign an access list to it. Then you specify a VRF for the traffic that matches the criteria in the route map. Use the **set vrf** command to specify the VRF through which the outbound VPN packets are routed.

Define the VRF before configuring the route map; otherwise the console displays an error.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. Do one of the following:
   - **match ip address***acl-number* [*acl-number*... | *acl-name*...] | *acl-name* [*acl-name*... | *acl-number*]
5. **set vrf***vrf-name*
6. **exit**

DETAILED STEPS

	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	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map RED permit 10`	Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing. Enters route map configuration mode.

Step 4	Do one of the following:
	• **match ip address***acl-number* [*acl-number...* | *acl-name...*] | *acl-name* [*acl-name...* | *acl-number*]
	
	Example:
	`Router(config-route-map)# match ip address 1`
	
	Example:
	`match length minimum-length maximum-length`
	
	Example:
	`Router(config-route-map)# match length 3 200`
		Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets.
		• IP access lists are supported.
		• The example configures the route map to use standard access list 1 to define match criteria.
		or
		Specifies the Layer 3 packet length in the IP header as a match criteria in a class map.
		• The example configures the route map to match packets that are between 3 and 200 bytes in size.

Step 5	**set vrf***vrf-name* **Example:** `Router(config-route-map)# set vrf RED`	Defines which VRF to send VPN packets that are successfully matched. • The example policy routes matched packets out to the VRF named RED.
Step 6	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.

Applying a Route Map to an Interface

You apply a route map to the incoming interface with the ip policy route-map global configuration command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip policy route-map** [*map-tag*]
5. **ip vrf receive***vrf-name*
6. **end**

DETAILED STEPS

	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 FastEthernet 0/1`	Configures an interface and enters interface configuration mode.
Step 4	**ip policy route-map** [*map-tag*] **Example:** `Router(config-if)# ip policy route-map RED`	Identifies a route map to use for policy routing on an interface.
Step 5	**ip vrf receive***vrf-name* **Example:** `Router(config-if)# ip vrf receive VRF_1`	Adds the IP addresses that are associated with an interface into the VRF table. • This command can be configured so that the receiving packets can be received by the router after being set to a specific VRF.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and enters global configuration mode.

Configuring IP VRF Receive on the Interface

You must add the source IP address to the VRF selection table. VRF Selection is a one-way (unidirectional) feature. It is applied to the incoming interface. If a match and set operation occurs in the route map but there is no receive entry in the local VRF table, the packet will be dropped if the packet destination is local.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip policy route-map** [*map-tag*]
5. **ip vrf receive** *vrf-name*
6. **end**

DETAILED STEPS

	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 FastEthernet 0/1`	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip policy route-map** [*map-tag*] **Example:** `Router(config-if)# ip policy route-map RED`	Identifies a route map to use for policy routing on an interface.
Step 5	**ip vrf receive** *vrf-name* **Example:** `Router(config-if)# ip vrf receive VRF_1`	Adds the IP addresses that are associated with an interface into the VRF table. • This command must be configured for each VRF that will be used for VRF selection.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Verifying the Configuration

To verify that the configuration is correct, perform the steps in this section.

SUMMARY STEPS

1. **enable**
2. **show ip access-list** [*access-list-number* | *access-list-name*]
3. **show route-map** [*map-name*]
4. **show ip policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip access-list** [*access-list-number*	*access-list-name*] **Example:** `Router# show ip access-list`	Displays the contents of all current IP access lists. • This command is used to verify the match criteria that is defined in the access list. Both named and numbered access lists are supported.

	Command or Action	Purpose
Step 3	**show route-map** [*map-name*] **Example:** `Router# show route-map`	Displays all route maps configured or only the one specified. • This command is used to verify match and set clauses within the route map.
Step 4	**show ip policy** **Example:** `Router# show ip policy`	Displays the route map used for policy routing. • This command can be used to display the route map and the associated interface.

Configuration Examples for Directing MPLS VPN Traffic Using Policy-Based Routing

- Configuring Policy-Based Routing with a Standard Access List Example, page 111
- Verifying Policy-Based Routing Example, page 111

Configuring Policy-Based Routing with a Standard Access List Example

In the following example, three standard access lists are created to define match criteria for three different subnets. A route map called PBR-VRF-Selection is assigned to interface Ethernet 0/1. If interface Ethernet 0/1 receives a packet whose source IP address is part of the 10.1.0.0/24 subnet, that packet is sent to VRF_1.

```
access-list 40 permit 10.1.0.0 0.0.255.255
access-list 50 permit 10.2.0.0 0.0.255.255
access-list 60 permit 10.3.0.0 0.0.255.255
route-map PBR-VRF-Selection permit 10
 match ip address 40
 set vrf VRF_1
 !
route-map PBR-VRF-Selection permit 20
 match ip address 50
 set vrf VRF_2
 !
route-map PBR-VRF-Selection permit 30
 match ip address 60
 set vrf VRF_3
 !
interface Ethernet0/1
 ip address 192.168.1.6 255.255.255.252
 ip policy route-map PBR-VRF-Selection
 ip vrf receive VRF_1
 ip vrf receive VRF_2
 ip vrf receive VRF_3
```

Verifying Policy-Based Routing Example

The following verification examples show defined match criteria and route-map policy configuration.

Verifying Match Criteria

To verify the configuration of match criteria for PBR VRF selection, use the **show ip access-lists** command. The following **show ip access-lists** command output displays three subnet ranges defined as match criteria in three standard access-lists:

```
Router# show ip access-lists

Standard IP access list 40
    10 permit 10.1.0.0, wildcard bits 0.0.255.255
Standard IP access list 50
    10 permit 10.2.0.0, wildcard bits 0.0.255.255
Standard IP access list 60
    10 permit 10.3.0.0, wildcard bits 0.0.255.255
```

Verifying Route-Map Configuration

To verify route-map configuration, use the **show route-map** command. The output displays the match criteria and set action for each route-map sequence. The output also displays the number of packets and bytes that have been policy routed per each route-map sequence.

```
Router# show route-map
route-map PBR-VRF-Selection, permit, sequence 10
  Match clauses:
    ip address (access-lists): 40
  Set clauses:
    vrf VRF_1
  Policy routing matches: 0 packets, 0 bytes
route-map PBR-VRF-Selection, permit, sequence 20
  Match clauses:
    ip address (access-lists): 50
  Set clauses:
    vrf VRF_2
  Policy routing matches: 0 packets, 0 bytes
route-map PBR-VRF-Selection, permit, sequence 30
  Match clauses:
    ip address (access-lists): 60
  Set clauses:
    vrf VRF_3
  Policy routing matches: 0 packets, 0 bytes
```

Verifying PBR VRF Selection Policy

The following **show ip policy** command output displays the interface and associated route map that is configured for policy routing.

```
Router# show ip policy
Interface      Route map
Ethernet0/1    PBR-VRF-Selection
```

Additional References

Related Documents

Related Topic	Document Title
Basic MPLS VPNs	Configuring MPLS Layer 3 VPNs

Related Topic	Document Title
MPLS VPN Carrier Supporting Carrier	• MPLS VPN Carrier Supporting Carrier Using LDP and an IGP • MPLS VPN Carrier Supporting Carrier with BGP
MPLS VPN InterAutonomous Systems	• MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels • MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	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 obtain lists of supported MIBs by platform and Cisco software release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	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 Technical Support 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. 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 Directing MPLS VPN Traffic Using Policy-Based Routing

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 7 Feature Information for Directing MPLS VPN Traffic Using Policy-Based Routing

Feature Name	Releases	Feature Configuration Information
MPLS VPN--VRF Selection using Policy-Based Routing	12.3(7)T 12.2(25)S	This feature allows you to classify and forward VPN traffic based on match criteria, such as IP access lists and packet length.

Directing MPLS VPN Traffic Using a Source IP Address

This module explains how to set up an interface on a provider edge (PE) router to route packets to different Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) based on the source IP address of the packet.

- Finding Feature Information, page 117
- Prerequisites for Directing MPLS VPN Traffic Using a Source IP Address, page 117
- Restrictions for Directing MPLS VPN Traffic Using a Source IP Address, page 118
- Information About Directing MPLS VPN Traffic Using a Source IP Address, page 120
- How to Enable MPLS VPN Traffic To Be Routed Using a Source IP Address, page 124
- Configuration Examples for Directing MPLS VPN Traffic Using a Source IP Address, page 128
- Additional References, page 129
- Feature Information for Directing MPLS VPN Traffic Using a Source IP Address, page 131

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Directing MPLS VPN Traffic Using a Source IP Address

- MPLS VPNs must be enabled in the provider network.
- Cisco Express Forwarding (CEF) must be enabled on any interfaces that have this feature enabled.
- The Cisco IOS software must support MPLS VPNs, and the provider network must have MPLS Label Distribution Protocol (LDP) installed and running.
- This feature is supported on the Cisco 7200 series, 7500 series, and 12000 series router platforms.

Restrictions for Directing MPLS VPN Traffic Using a Source IP Address

VRF Select is supported only in Service Provider (-p-) images.

Unidirectional Traffic

This is a unidirectional feature and can only be used from a customer (IP-based) network into a provider (MPLS-based) network. This feature cannot be used from a provider network to a customer network.

Subnet Masks

Subnet masks should be kept as short as possible for Engine 2 line cards. Performance can degrade with longer subnet masks (/24 or /32, for example).

traceroute Command

An IP **traceroute** command from a customer edge (CE) router that has this featue enabled to a typical MPLS VPN VRF CE router works as expected. However, an IP **traceroute** command from a typical MPLS VPN VRF CE router to a CE router that has this feature enabled may fail to show all the relevant hop information across the core.

Supported Static Route Configurations

When configuring static routes in an MPLS or MPLS VPN environment, some variations of the **ip route** and **ip route vrf** commands are not supported. These variations of the commands are not supported in Cisco IOS releases that support the Tag Forwarding Information Base (TFIB), specifically Cisco IOS Releases 12.xT, 12.xM, and 12.0S. The TFIB cannot resolve prefixes when the recursive route over which the prefixes travel disappears and then reappears. However, the command variations are supported in Cisco IOS releases that support the MPLS Forwarding Infrastructure (MFI), specifically Cisco IOS Release 12.2(25)S and later. Use the following guidelines when configuring static routes.

Supported Static Routes in an MPLS Environment

The following **ip route** command is supported when you configure static routes in MPLS environment:

ip route *destination-prefix mask interface next-hop-address*

The following **ip route** commands are supported when you configure static routes in an MPLS environment and configure load sharing with static nonrecursive routes and a specific outbound interface:

ip route *destination-prefix mask* **interface1 next-hop1**

ip route *destination-prefix mask* **interface2 next-hop2**

Unsupported Static Routes in an MPLS Environment that Uses the TFIB

The following **ip route** command is not supported when you configure static routes in an MPLS environment:

ip route *destination-prefix mask next-hop-address*

The following **ip route** command is not supported when you configure static routes in an MPLS environment and enable load sharing where the next hop can be reached through two paths:

ip route *destination-prefix mask next-hop-address*

The following **ip route** command is not supported when you configure static routes in an MPLS environment and enable load sharing where the destination can be reached through two next hops:

ip route *destination-prefix mask* **next-hop1**

ip route *destination-prefix mask* **next-hop2**

Use the *interface* and *next-hop* arguments when specifying static routes.

Supported Static Routes in an MPLS VPN Environment

The following **ip route vrf** commands are supported when you configure static routes in a MPLS VPN environment, and the next hop and interface are in the same VRF:

- ◦ **ip route vrf** *vrf-name destination-prefix mask next-hop-address*
  ◦ **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address*
  ◦ **ip route vrf** *vrf-name destination-prefix mask* **interface1 next-hop1**
  ◦ **ip route vrf** *vrf-name destination-prefix mask* **interface2 next-hop2**

The following **ip route vrf** commands are supported when you configure static routes in a MPLS VPN environment, and the next hop is in the global table in the MPLS cloud in the global routing table. For example, these commands are supported when the next hop is pointing to the Internet Gateway.

- ◦ **ip route vrf** *vrf-name destination-prefix mask next-hop-address* **global**
  ◦ **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address* (This command is supported when the next hop and interface are in the core.)

The following **ip route** commands are supported when you configure static routes in a MPLS VPN environment and enable load sharing with static nonrecursive routes and a specific outbound interfaces:

ip route *destination-prefix mask* **interface1 next-hop1**

ip route *destination-prefix mask* **interface2 next-hop2**

Unsupported Static Routes in an MPLS VPN Environment that Uses the TFIB

The following **ip route** command is not supported when you configure static routes in a MPLS VPN environment, the next hop is in the global table in the MPLS cloud within the core, and you enable load sharing where the next hop can be reached through two paths:

ip route vrf *destination-prefix mask next-hop-address* **global**

The following **ip route** commands are not supported when you configure static routes in a MPLS VPN environment, the next hop is in the global table in the MPLS cloud within the core, and you enable load sharing where the destination can be reached through two next hops:

ip route vrf *destination-prefix mask* **next-hop1 global**

ip route vrf *destination-prefix mask* **next-hop2 global**

The following **ip route vrf** commands are not supported when you configure static routes in an MPLS VPN environment, and the next hop and interface are in the same VRF:

ip route vrf *vrf-name destination-prefix mask* **next-hop1**

ip route vrf *vrf-name destination-prefix mask* **next-hop2**

Supported Static Routes in an MPLS VPN Environment Where the Next Hop Resides in the Global Table on the CE Router

The following **ip route vrf** command is supported when you configure static routes in a MPLS VPN environment, and the next hop is in the global table on the CE side. For example, the following command is supported when the destination-prefix is the CE router's loopback address, as in EBGP multihop cases.

ip route vrf *vrf-name destination-prefix mask interface next-hop-address*

The following **ip route** commands are supported when you configure static routes in a MPLS VPN environment, the next hop is in the global table on the CE side, and you enable load sharing with static non-recursive routes and a specific outbound interfaces:

ip route *destination-prefix mask* **interface1 nexthop1**

ip route *destination-prefix mask* **interface2 nexthop2**

Information About Directing MPLS VPN Traffic Using a Source IP Address

- Introduction to Directing MPLS VPN Traffic Using a Source IP Address, page 120
- How MPLS VPN Traffic Is Routed Using the Source IP Address, page 120
- Example of MPLS VPN Traffic Being Routed Based on the Source IP Address, page 121
- MPLS VPN Traffic Is Unidirectional, page 122
- Advantages of Using the Source IP Address over Per-Interface IP VPN Configuration, page 123
- Benefits of Directing MPLS VPN Traffic Using a Source IP Address, page 124

Introduction to Directing MPLS VPN Traffic Using a Source IP Address

This feature allows packets arriving on an interface to be switched into the appropriate VRF table based upon the source IP address of the packets. Once the packets have been "selected" into the correct VRF routing table, they are processed normally based upon the destination address and forwarded through the rest of the MPLS VPN.

In most cases, this is a "one way" feature; it works on packets coming from the end users to the PE router.

How MPLS VPN Traffic Is Routed Using the Source IP Address

This feature uses the following process to route packets from the customer networks to the PE router and into the provider network.

A two-table lookup mechanism is used at the ingress interface of the PE router to determine the routing and forwarding of packets coming from the customer networks, which use IP protocols, to the MPLS VPN networks, which use MPLS protocols.

- The first table, the VRF Selection table, is used to compare the source IP address of the packet with a list of IP addresses in the table. Each IP address in the table is associated with an MPLS VPN. If a match is found between the source IP address of the packet and an IP address in the VRF Selection table, the packet is routed to the second table (the VRF table) or the routing table for the appropriate VPN.

If no match is found in the table for the source IP address of the packet, the packet will either be routed via the global routing table used by the PE router (this is the default behavior), or will be dropped. See the

Configuring a VRF to Eliminate Unnecessary Packet Forwarding Example, page 129 for more information.

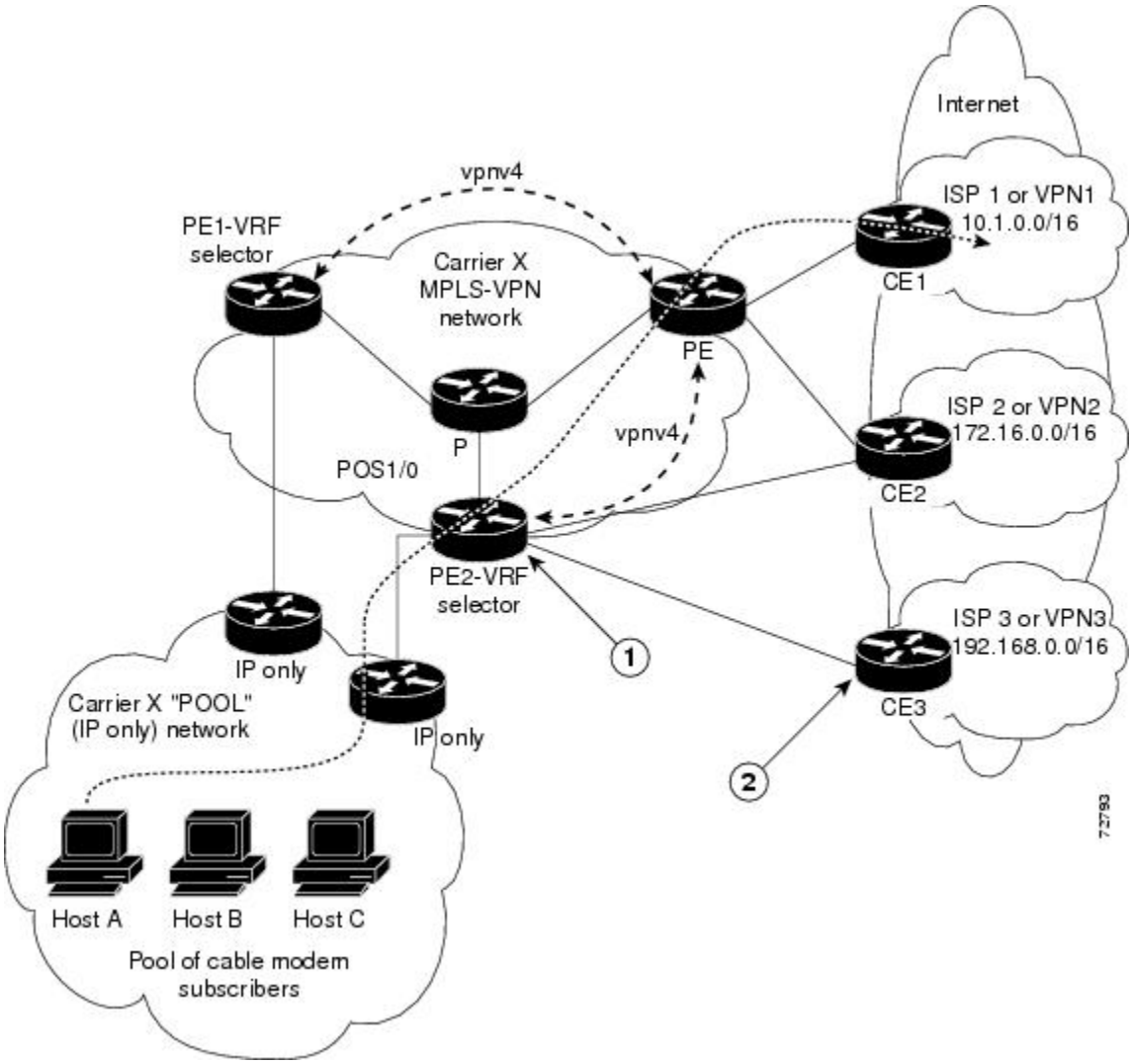
- The second table, the VRF table (also known as the VPN routing and forwarding table), contains the virtual routing and forwarding information for the specified VPN and is used to forward the selected VPN traffic to the correct MPLS label switched path (LSP) based upon the destination IP address of the packet.

The VRF Selection process removes the association between the VRF and the interface and allows more than one MPLS VPN VRF to be associated with the interface.

Example of MPLS VPN Traffic Being Routed Based on the Source IP Address

An example of this feature is a network carrier that allows subscribers to the carrier to choose from multiple Internet service providers (ISPs) for Internet access. The figure below provides an example of this feature with an IP-based Host network, an MPLS VPN network, and three ISPs connected to the MPLS VPN network.

Figure 5 Implementation Example



In the figure above, Carrier X represents the network carrier; Host A, Host B and Host C represent the carrier subscribers; and ISP 1, ISP 2 and ISP 3 represent the ISPs.

- PE2 acts as both a VRF selector and a typical MPLS VPN PE router to CE2 and CE3.
- ISPs 1 through 3 provide a list of IP addresses to Carrier X so that each host in the “POOL” network can be properly addressed. This host addressing would most likely be done by using the DHCP or DNS services of Carrier X.

A dashed line represents the path of a packet traveling from Host A to ISP 1. Host A chooses ISP 1 to use as its ISP. Carrier X provides an IP address to Host A that falls within the range of the ISP 1 registered network addresses (1.1.0.0/16). Based upon this IP address allocation, the VRF Selection criteria is set.

By using default routes, hosts on the POOL network (such as Host A), forward traffic from the Carrier X IP-based (POOL) network to the Carrier X MPLS-based VPN network. PE2 has been configured with this feature. Therefore, the MPLS VPN network forwards the traffic from Host A to ISP 1.

This is a one-way (unidirectional) feature in most implementations; it only works on packets coming from the customer networks to a PE router. Traffic coming from the ISPs to the hosts (in the example, traffic traveling from the ISPs on the right to the hosts on the left) is not affected by this feature and does not have to be returned via an MPLS path. This traffic can return via the shortest available IP path.

Another example is a Cable Modem Termination System (CMTS). If the owner of the CMTS wants to allow cable modem subscribers to choose their ISP from a group of ISPs, this feature provides a fast and scalable solution.

MPLS VPN Traffic Is Unidirectional

In the figure above, the end users are typical Internet home users. If this were a two-way (bidirectional) feature, traffic coming from the ISPs to the hosts would be required to use only the PE routers that have this feature enabled, which might cause performance issues.

When traffic from the POOL network goes through the Carrier network to the ISP networks for Internet access, the traffic in the Carrier network must be forwarded using MPLS VPN paths, because the router has “selected” the traffic into the correct MPLS VPN.

Traffic from the ISP networks to the POOL network does not have to use MPLS VPN paths in the Carrier network and can use any path that is most efficient to return to the POOL network. This traffic can use a path that uses either MPLS or IP for routing and forwarding and does not have to travel via an MPLS VPN.

Traffic from the ISP networks to the POOL networks can be forwarded using the global routing table used by every interface. One way to accomplish this is to enter VRF static routes on the PE router interfaces connected to the ISPs. The VRF static routes would route traffic from the ISPs to the Carrier network. See the Establishing IP Static Routes for a VRF Instance, page 126 for information on placing a default VRF static route onto an interface.

Establishing static VRF routes allows traffic from the ISPs to enter the Carrier network as traffic that can only be routed by using the global routing table toward the POOL network.

If the ISPs do not provide global host address space, or this feature is not being used to route Internet traffic, the PE interfaces connected to the ISPs must be placed into a VRF. If the PE interfaces are using VRFs for routing traffic from the ISPs, all traffic from the ISPs to the hosts through the Carrier network would be forwarded using MPLS VPN paths, and performance would not be as optimal as if IP forwarding was used.

Normal IP-based VPN operations, such as populating the Routing Information Base (RIB) and Forwarding Information Base (FIB) from a routing protocol such as Border Gateway Protocol (BGP), are used to route and forward packets within the various VPNs in the customer networks. The provider network uses MPLS-based routing protocols to perform VPN routing and forwarding inside the provider network.

• Conditions That Cause MPLS VPN Traffic To Become Bidirectional, page 123

Conditions That Cause MPLS VPN Traffic To Become Bidirectional

Forwarding of traffic from the Carrier network to the POOL network by using the global routing table is only possible if the ISPs have provided registered IP address space for all of the subscribed users within the POOL network from the global routing table.

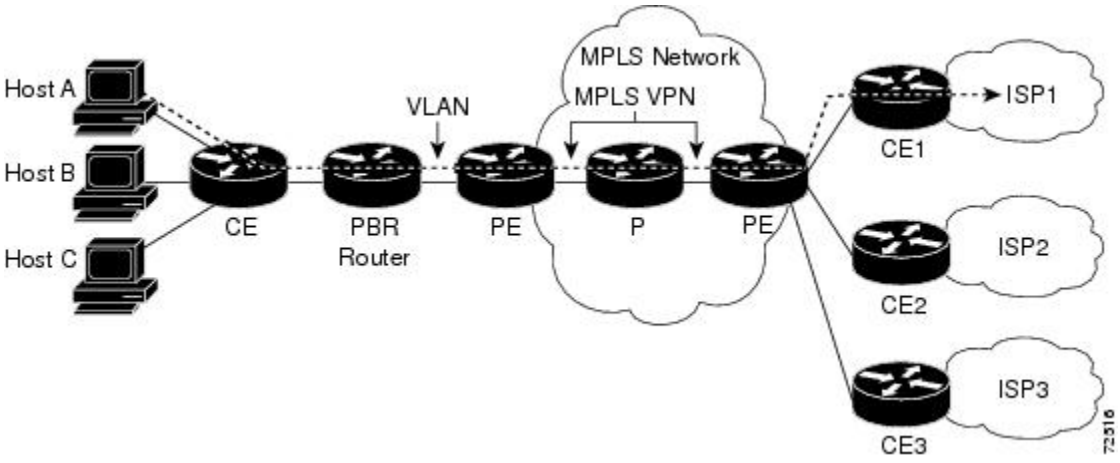
If the POOL network uses IP addresses that are not globally routeable and are designed for a nonconnected enterprise (defined by RFC 1918), this feature becomes bidirectional. All traffic being sent and received by the host would have to travel via a router that has this feature enabled. The POOL network cannot be addressed with overlapping address space, regardless of the type of address space being used.

Advantages of Using the Source IP Address over Per-Interface IP VPN Configuration

This feature removes the association between a VPN and an interface. Before this feature was introduced, the following implementation was used to route outgoing MPLS VPN packets to different destinations:

- A policy-based router (PBR) is attached to the CE router.
- The egress side of the PBR router side has VLANs connected to a PE.
- The PBR router uses a policy-based route map to select the correct output (VLAN) interface and each VLAN is under a specific VRF. The figure below illustrates a sample configuration of using a PBR router for routing MPLS packets to different destinations.

Figure 6 Implementation of Multiple VPNs



The following limitations apply to PBR-based solutions that use this implementation:

- Policy routing and MPLS VPN functions cannot be performed on the same platform. Integration into a single platform is critical for manageability and support.
- VRF is limited to one VPN per interface, which limits scalability.
- The Cisco 7500 series router is used for the PBR, which can limit network performance.
- There is no network redundancy.
- The PBR is the only point of connection for all the networks attached to the PBR. The capacity and the performance capabilities of the PBR router are critical.

- There is no diversity in the connectivity to the networks.
- Every network is required to connect to every PBR. If every network is not connected to every PBR, packets from the end user to the PBR would be dropped because the PBR would have no way of switching the IP traffic properly.
- Adding multiple PBRs that are interconnected introduces more network policy-routed hops.

This feature addresses the limitations of and problems with using a PBR for packet routing and forwarding.

Benefits of Directing MPLS VPN Traffic Using a Source IP Address

Association of VPN to interface is removed

This feature removes the association between a VPN and an interface, thus allowing packets from the Host network to the provider network to have more than one VPN available per interface.

Access to every customer network is possible from every PE router in the provider network

Access points to each network can be established at any MPLS PE router, and can be made redundant by connections to multiple PE routers (for example, the CE2 router in the figure above).

Multiple points in the provider network can be used for VPN routing and forwarding

MPLS VPNs, like IP, are connectionless. Any PE router can carry MPLS VPN traffic from the MPLS network out to the CE routers.

How to Enable MPLS VPN Traffic To Be Routed Using a Source IP Address

- Enabling Routing of MPLS VPN Traffic Based on the Source IP Address, page 124
- Establishing IP Static Routes for a VRF Instance, page 126

Enabling Routing of MPLS VPN Traffic Based on the Source IP Address

Perform the following steps to enable MPLS VPN traffic to be routed based on the source IP address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf selection source** *source-IP-address source-IP-mask* **vrf** *vrf-name*
4. **interface** *type number*
5. **ip vrf select source**
6. **ip vrf receive** *vrf_name***vrf**
7. **end**
8. **show ip route vrf**
9. **show ip vrf select**

DETAILED STEPS

	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	**vrf selection source** *source-IP-address source-IP-mask* **vrf** *vrf-name* **Example:** `Router(config)# vrf selection source 16.16.0.0 255.255.0.0 vrf vpn1`	Populates a source IP address to a VRF selection table.
Step 4	**interface** *type number* **Example:** `Router(config)# interface FastEthernet 0/1`	Configures an interface and enters interface configuration mode.
Step 5	**ip vrf select source** **Example:** `Router(config-if)# ip vrf select source`	Enables an interface to direct MPLS VPN traffic based on the source IP address of the packet.
Step 6	**ip vrf receive** *vrf_name***vrf** **Example:** `Router(config-if)# ip vrf receive vpn1`	Adds all the IP addresses that are associated with an interface into a VRF table.
Step 7	**end** **Example:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.

	Command or Action	Purpose
Step 8	**show ip route vrf** **Example:** `Router# show ip route vrf`	Displays the IP routing table associated wtih a VRF instance. Use this command to verify the configuration.
Step 9	**show ip vrf select** **Example:** `Router# show ip vrf select`	Displays information about the VRF selection.

Establishing IP Static Routes for a VRF Instance

Traffic coming from the ISPs to the hosts does not require the use of the MPLS VPN paths; this traffic can use the shortest IP route back to the host.

VPN static routes for traffic returning to the customer networks are only necessary if VPN traffic returning to the customer networks is being forwarded back from the enabled interface. The remote PE router could also be configured to route return traffic to the customer networks directly by using the global routing table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip route vrf** *vrf_name prefix mask* [*next-hop-address*] [**interface** {*interface-number*}] [**global**] [**distance**] [**permanent**] [**tag** *tag*]

DETAILED STEPS

	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 FastEthernet 0/1`	Configures an interface and enters interface configuration mode.
Step 4	**ip route vrf** *vrf_name prefix mask* [*next-hop-address*] [**interface** {*interface-number*}] [**global**] [**distance**] [**permanent**] [**tag** *tag*] **Example:** `Router(config-if)# ip route vrf vpn1 16.16.0.0 255.255.0.0 POS1/0`	Establishes static routes for a VRF.

- Troubleshooting Tips, page 127

Troubleshooting Tips

- Enter the **debug vrf select** command to enable debugging for this feature.

Note The **debug vrf select** command can cause many messages to be logged when you change the configuration and when switching occurs.

- The following error messages appear if problems occur while configuring this feature:
  - If you attempt to configure a nonexisting VRF Selection table:

```
Router(config)# vrf selection source 2.0.0.0o 255.255.0.0 vrf VRF_NOEXIST
VRF Selection: VRF table VRF_NOEXIST does not exist.
```

- If you attempt to remove a VRF Selection entry that does not exist:

```
Router(config)# no vrf selection source 2.0.0.0 255.255.0.0 vrf VRF1
VRF Selection: Can't find the node to remove.
```

- If you attempt to configure a duplicate IP address and subnet mask for a VRF Selection entry:

```
Router(config)# vrf selection source 2.0.0.0 255.0.0.0 vrf VRF_AOL
Router(config)# vrf selection source 2.0.0.0 255.0.0.0 vrf VRF_AOL
VRF Selection: duplicate address and mask configured.
```

- If an inconsistent IP address and mask are used for a VRF Selection entry:

```
Router(config)# vrf selection source 170.1.2.1 255.255.255.0 vrf red
% Inconsistent address and mask

Router(config)# vrf selection source 170.1.2.1 255.255.255.255 vrf red
```

- If you attempt to configure a VRF instance on an interface that has this feature already configured:

```
Router(config-if)# ip vrf select source

Router(config-if)# ip vrf forward red
% Can not configure VRF if VRF Select is already configured
To enable VRF, first remove VRF Select from the interface
```

- If you attempt to configure an entry on an interface that has this feature already configured:

```
Router(config-if)# ip vrf forward red
Router(config-if)# ip vrf select source
% Can not configure VRF Select if interface is under a non-global VRF
To enable VRF Select, first remove VRF from the interface
```

Configuration Examples for Directing MPLS VPN Traffic Using a Source IP Address

- Enabling MPLS VPN Traffic To Be Routed Based on Source IP Address Example, page 128
- Configuring a VRF to Eliminate Unnecessary Packet Forwarding Example, page 129
- Verifying the Configuration Example, page 129

Enabling MPLS VPN Traffic To Be Routed Based on Source IP Address Example

The following example defines two entries (vpn1 and vpn2) in the VRF Selection table. In this example, packets with the source address of 16.16.0.0 will be routed to the VRF vpn1, and packets with the source address of 17.17.0.0 will be routed to the VRF vpn2:

```
Router(config)# vrf selection source 16.16.0.0 255.255.0.0 vrf vpn1
Router(config)# vrf selection source 17.17.0.0 255.255.0.0 vrf vpn2
```

The following example creates IP static routes for two VRFs (vpn1 and vpn2) for the POS1/0 interface:

```
Router(config)# ip route vrf vpn1 16.16.0.0 255.255.0.0 POS1/0
Router(config)# ip route vrf vpn1 17.17.0.0 255.255.0.0 POS1/0
```

The following example configures the POS1/0 interface for this feature and adds the configured IP address (31.0.0.1) to the VRFs vpn1 and vpn2 as connected routes.

```
Router(config)# interface POS1/0
Router(config-if)# description Link to CE1 POS1/0 (eng2)
Router(config-if)# ip vrf select source
Router(config-if)# ip vrf receive vpn1
Router(config-if)# ip vrf receive vpn2
Router(config-if)# ip address 31.0.0.1 255.0.0.0
Router(config-if)# no ip directed broadcast
Router(config-if)# load-interval 30
Router(config-if)# crc 32
Router(config-if)# end
```

Configuring a VRF to Eliminate Unnecessary Packet Forwarding Example

If a packet arrives at an interface that has VRF Select enabled, and its source IP address does not match any VRF Select definition, that packet will be forwarded via the global routing table. This default behavior could cause problems if IP address spoofing is being implemented. Unnecessary traffic could be forwarded via the global routing table. To eliminate this unnecessary routing of packets, create a VRF Selection definition that will forward all unknown incoming traffic to a null interface.

The following configuration causes all traffic not matching a more specific VRF Selection definition to be routed to the Null0 interface, thus dropping the packets.

```
Router(config)# ip vrf VRF_DROP
Router(config-vrf)# rd 999:99
Router(config-vrf)# route-target export 999:99
Router(config-vrf)# route-target import 999:99
Router(config-vrf)# exit
Router(config)# vrf selection source 0.0.0.0 0.0.0.0 vrf VRF_DROP
Router(config)# ip route vrf VRF_DROP 0.0.0.0 0.0.0.0 Null0
```

Verifying the Configuration Example

This example shows the IP routing table associated with the VRF vrf1:

```
Router# show ip route vrf vpn1
Routing Table: vpn1
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
Gateway of last resort is not set
B    33.0.0.0/8 [200/0] via 10.10.10.10, 00:00:37
5.0.0.0/16 is subnetted, 1 subnets
B       5.19.0.0 [200/0] via 10.10.10.10, 00:00:37
14.0.0.0/32 is subnetted, 1 subnets
B       14.14.14.14 [200/0] via 10.10.10.10, 00:00:37
15.0.0.0/32 is subnetted, 1 subnets
S       15.15.15.15 [1/0] via 34.0.0.1, POS1/1
```

Additional References

Related Documents

Related Topic	Document Title
Basic MPLS VPNs	Configuring MPLS Layer 3 VPNs
MPLS VPN Carrier Supporting Carrier	• MPLS VPN Carrier Supporting Carrier Using LDP and an IGP • MPLS VPN Carrier Supporting Carrier with BGP

Related Topic	Document Title
MPLS VPN InterAutonomous Systems	• MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels • MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

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 obtain lists of supported MIBs by platform and Cisco software release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

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 Technical Support 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. 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 Directing MPLS VPN Traffic Using a Source IP Address

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 8 Feature Information for Directing MPLS VPN Traffic Using a Source IP Address

Feature Name	Releases	Feature Configuration Information
VRF Selection Based on Source IP Address	12.0(22)S 12.0(23)S 12.0(24)S 12.0(26)S	This feature lets you direct MPLS VPN traffic based on the source IP address of the packet.

MPLS VPN--Show Running VRF

The MPLS VPN--Show Running VRF feature provides a Cisco IOS command-line interface (CLI) option to display a subset of the running configuration on a router that is linked to a Virtual Private Network (VPN) routing and forwarding (VRF) instance. You can display the configuration of a specific VRF or of all VRFs configured on a router.

On heavily loaded routers, the display of the configuration file might require several pages or screens. As the configuration increases in size and complexity, the possibility of misconfiguration also increases. You might find it difficult to trace a problem on a router where you have several VRFs configured. A command that displays all the elements of the configuration linked to a VRF allows for easier troubleshooting on a per-VRF basis and facilitates comparisons among configurations of different VRFs on the same router.

- Finding Feature Information, page 133
- Prerequisites for MPLS VPN--Show Running VRF, page 133
- Restrictions for MPLS VPN--Show Running VRF, page 134
- Information About MPLS VPN--Show Running VRF, page 134
- How to Configure MPLS VPN--Show Running VRF, page 135
- Configuration Examples for MPLS VPN--Show Running VRF, page 136
- Additional References, page 136
- Feature Information for MPLS VPN--Show Running VRF, page 137
- Glossary, page 138

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for MPLS VPN--Show Running VRF

- A Cisco IOS image that supports VRFs installed on the router
- At least one VRF configured on the router
- Cisco Express Forwarding for MPLS VPN routing and forwarding

Restrictions for MPLS VPN--Show Running VRF

Any element of the running configuration of the router that is not linked directly to a VRF is not displayed. For example, a route map associated with a Border Gateway Protocol (BGP) neighbor in a VRF address-family configuration is not displayed. The VRF address-family configuration under BGP is displayed, but the route-map configuration is not. An exception to this general rule is the display of a controller configuration (for more information, see the Display of Configuration Not Directly Linked to a VRF, page 135).

Information About MPLS VPN--Show Running VRF

- Configuration Elements Displayed for the MPLS VPN--Show Running VRF Feature, page 134
- Display of VRF Routing Protocol Configuration, page 134
- Display of Configuration Not Directly Linked to a VRF, page 135

Configuration Elements Displayed for the MPLS VPN--Show Running VRF Feature

You can display the running configuration associated with a specific VRF or all VRFs on the router by entering the **show running-config vrf** command. To display the running configuration of a specific VRF, enter the name of the VRF as an argument to the **show running-config vrf** command. For example, for a VRF named vpn3, you enter:

```
Router# show running-config vrf vpn3
```

The **show running-config vrf** command displays the following elements of the running configuration on a router:

- The VRF configuration

This includes any configuration that is applied in the VRF submode.

- The configuration of each interface in the VRF

Entering a **show run vrf** *vpn-name* command is the same as executing a **show running-config interface** *type number* for each interface that you display by use of the **show ip vrf** *vpn-name* command. The interfaces display in the same sorted order that you would expect from the **show ip interface** command.

For a channelized interface, the configuration of the controller is displayed (as shown by the **show run controller** *controller-name* command).

For a subinterface, the configuration of the main interface is displayed.

Display of VRF Routing Protocol Configuration

Open Shortest Path First (OSPF), Routing Information Protocol (RIP), Border Gateway Protocol (BGP), Enhanced Interior Gateway Routing Protocol (EIGRP), and static routing are routing protocols that support VRF configuration.

OSPF has one process per VRF. The **show running-config vrf** command display includes the complete configuration of any OSPF process associated with the VRF. For example, the following shows the sample display for OSPF process 101, which is associated with the VRF named vpn3:

```
router ospf 101 vrf vpn3
 log-adjacency-changes
 area 1 sham-link 10.43.43.43 10.23.23.23 cost 10
 network 172.17.0.0 0.255.255.255 area 1
```

RIP, BGP, and EIGRP support VRF address-family configuration. If a VRF address family for the VRF exists for any of these routing protocols, a configuration in the following format is displayed:

```
router
protocol
 {AS
 | PID
}
!
address-family ipv4 vrf
 vrf-name
.
.
.
```

Where the *protocol* argument is one of the following: **rip**, **bgp** or **eigrp**; the *AS* argument is an autonomous system number; the *PID* argument is a process identifier; and the *vrf-name* argument is the name of the associated VRF.

The following shows a sample display for a BGP with autonomous system number 100 associated with a VRF named vpn3:

```
!
router bgp 100
!
address-family ipv4 vrf vpn3
 redistribute connected
 redistribute ospf 101 match external 1 external 2
 no auto-summary
 no synchronization
 exit-address-family
 !
```

The **show running-config vrf** command also includes the configuration of any static routes configured in the VRF. For example:

```
ip route vrf vpn1 10.1.1.0 255.255.255.0 10.30.1.1 global
ip route vrf vpn1 10.1.2.0 255.255.255.0 10.125.1.2
```

Display of Configuration Not Directly Linked to a VRF

Any element of a configuration that is not linked directly to a VRF is not displayed. In some instances, the display of the configuration of an element that is not directly linked to a VRF is required.

For example, the **show running-config vrf** command displays the configuration of an E1 controller whose serial subinterfaces are in a VRF. The command displays the controller configuration and the subinterface configuration.

How to Configure MPLS VPN--Show Running VRF

There are no tasks for the MPLS VPN--Show Running VRF feature.

Configuration Examples for MPLS VPN--Show Running VRF

Additional References

Related Documents

Related Topic	Document Title
MPLS command descriptions	*Cisco IOS Multiprotocol Label Switching Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	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 software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	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 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 MPLS VPN--Show Running VRF

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 9 Feature Information for MPLS VPN--Show Running VRF

Feature Name	Releases	Feature Information
MPLS VPN--Show Running VRF	12.2(28)SB 12.0(32)SY 12.2(33)SRB 12.2(33)SXH 12.4(20)T	The MPLS VPN--Show Running VRF feature provides a CLI option to display a subset of the running configuration on a router that is linked to a VRF. You can display the configuration of a specific VRF or of all VRFs configured on a router. A command that displays all the elements of the configuration linked to a VRF allows for easier troubleshooting on a per-VRF basis and facilitates comparisons among configurations of different VRFs on the same router. In 12.2(28)SB, this feature was introduced. In 12.0(32)SY, support was added for a Cisco IOS 12.0SY release. In 12.2(33)SRB, support was added for a Cisco IOS 12.2SR release. In 122(33)SXH, support was added for a Cisco IOS 12.2SX release. In 12.4(20)T, support was added for a Cisco IOS 12.4T release.
		The following commands were introduced or modified: **show policy-map interface brief**, **show running-config vrf**.

Glossary

BGP --Border Gateway Protocol. An interdomain routing protocol that replaces External Gateway Protocol (EGP). BGP systems exchange reachability information with other BGP systems. BGP is defined by RFC 1163.

EGP --External Gateway Protocol. An internet protocol for exchanging routing information between autonomous systems. EGP is documented in RFC 904. Not to be confused with the general term exterior gateway protocol. EGP is an obsolete protocol that was replaced by Border Gateway Protocol (BGP).

EIGRP --Enhanced Interior Gateway Routing Protocol. Advanced version of Interior Gateway Routing Protocol (IGRP) developed by Cisco. Provides superior convergence properties and operating efficiency, and combines the advantages of link state protocols with those of distance vector protocols.

IGP --Interior Gateway Protocol. An internet protocol used to exchange routing information within an autonomous system. Examples of common Internet IGPs include Interior Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).

IGRP --Interior Gateway Routing Protocol. An Interior Gateway Protocol (IGP) developed by Cisco to address the issues associated with routing in large, heterogeneous networks.

MPLS --Multiprotocol Label Switching. A switching method that forwards IP traffic through the use of a label. This label instructs the routers and the switches in the network where to forward each packet based on preestablished IP routing information.

OSPF --Open Shortest Path First. A link-state, hierarchical, Interior Gateway Protocol (IGP) routing algorithm and routing protocol proposed as a successor to Routing Information Protocol (RIP) in the Internet community. OSPF features include least-cost routing, multipath routing, and load balancing. OSPF was derived from an early version of the Intermediate System-to-Intermediate System (IS-IS) protocol.

RIP --Routing Information Protocol. Internal Gateway Protocol (IGP) supplied with UNIX Berkeley Software Distribution (BSD) systems. RIP is the most common IGP in the Internet. It uses hop count as a routing metric.

VPN --Virtual Private Network. The result of a router configuration that enables IP traffic to use tunneling to travel securely over a public TCP/IP network.

MPLS VPN Half-Duplex VRF

The MPLS VPN Half-Duplex VRF feature provides scalable hub-and-spoke connectivity for subscribers of an Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) service. This feature addresses the limitations of hub-and-spoke topologies by removing the requirement of one virtual routing and forwarding (VRF) instance per spoke. This feature also ensures that subscriber traffic always traverses the central link between the wholesale service provider and the Internet service provider (ISP), whether the subscriber traffic is being routed to a remote network by way of the upstream ISP or to another locally or remotely connected subscriber.

- Finding Feature Information, page 141
- Prerequisites for Configuring MPLS VPN Half-Duplex VRF, page 141
- Restrictions for MPLS VPN Half-Duplex VRF, page 141
- Information About Configuring MPLS VPN Half-Duplex VRF, page 142
- How to Configure MPLS VPN Half-Duplex VRF, page 143
- Configuration Examples for MPLS VPN Half-Duplex VRF, page 150
- Additional References, page 155
- Feature Information for MPLS VPN Half-Duplex VRF, page 156

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring MPLS VPN Half-Duplex VRF

You must have a working MPLS core network.

Restrictions for MPLS VPN Half-Duplex VRF

The following features are not supported on interfaces configured with the MPLS VPN Half-Duplex VRF feature:

- Multicast

- MPLS VPN Carrier Supporting Carrier
- MPLS VPN Interautonomous Systems

Information About Configuring MPLS VPN Half-Duplex VRF

- MPLS VPN Half-Duplex VRF Overview, page 142
- Upstream and Downstream VRFs, page 142
- Reverse Path Forwarding Check, page 143

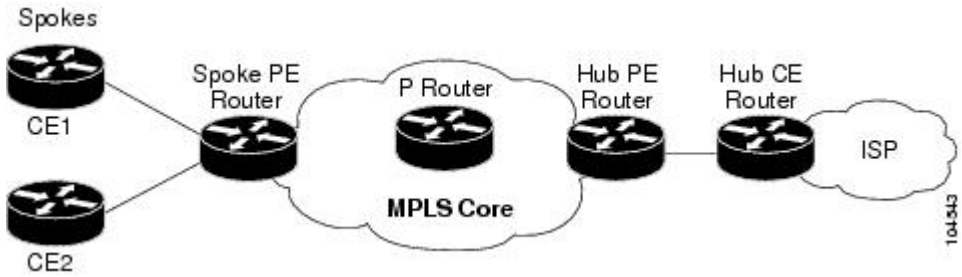
MPLS VPN Half-Duplex VRF Overview

The MPLS VPN Half-Duplex VRF feature provides:

- The MPLS VPN Half-Duplex VRF feature prevents local connectivity between subscribers at the spoke provider edge (PE) router and ensures that a hub site provides subscriber connectivity. Any sites that connect to the same PE router must forward intersite traffic using the hub site. This ensures that the routing done at the spoke site moves from the access-side interface to the network-side interface or from the network-side interface to the access-side interface, but never from the access-side interface to the access-side interface.
- The MPLS VPN Half-Duplex VRF feature prevents situations where the PE router locally switches the spokes without passing the traffic through the upstream ISP. This prevents subscribers from directly connecting to each other, which causes the wholesale service provider to lose revenue.
- The MPLS VPN Half-Duplex VRF feature improves scalability by removing the requirement of one VRF per spoke. If the feature is not configured, when spokes are connected to the same PE router each spoke is configured in a separate VRF to ensure that the traffic between the spokes traverses the central link between the wholesale service provider and the ISP. However, this configuration is not scalable. When many spokes are connected to the same PE router, configuration of VRFs for each spoke becomes quite complex and greatly increases memory usage. This is especially true in large-scale wholesale service provider environments that support high-density remote access to Layer 3 VPNs.

The figure below shows a sample hub-and-spoke topology.

Figure 7 Hub-and-Spoke Topology



Upstream and Downstream VRFs

The MPLS VPN Half-Duplex VRF feature uses two unidirectional VRFs to forward IP traffic between the spokes and the hub PE router:

- The upstream VRF forwards IP traffic from the spokes toward the hub PE router. This VRF typically contains only a default route but might also contain summary routes and several default routes. The default route points to the interface on the hub PE router that connects to the upstream ISP. The router dynamically learns about the default route from the routing updates that the hub PE router or home gateway sends.

Note Although the upstream VRF is typically populated from the hub, it is possible also to have a separate local upstream interface on the spoke PE for a different local service that would not be required to go through the hub: for example, a local Domain Name System (DNS) or game server service.

- The downstream VRF forwards traffic from the hub PE router back to the spokes. This VRF can contain:
  - PPP peer routes for the spokes and per-user static routes received from the authentication, authorization, and accounting (AAA) server or from the Dynamic Host Control Protocol (DHCP) server
  - Routes imported from the hub PE router
  - Border Gateway Protocol (BGP), Open Shortest Path First (OSPF), Routing Information Protocol (RIP), or Enhanced Interior Gateway Routing Protocol (EIGRP) dynamic routes for the spokes

The spoke PE router redistributes routes from the downstream VRF into Multiprotocol Border Gateway Protocol (MP-BGP). That router typically advertises a summary route across the MPLS core for the connected spokes. The VRF configured on the hub PE router imports the advertised summary route.

Reverse Path Forwarding Check

The Reverse Path Forwarding (RPF) check ensures that an IP packet that enters a router uses the correct inbound interface. The MPLS VPN Half-Duplex VRF feature supports unicast RPF check on the spoke-side interfaces. Because different VRFs are used for downstream and upstream forwarding, the RPF mechanism ensures that source address checks occur in the downstream VRF.

Unicast RPF is not on by default. You need to enable it, as described in Configuring Unicast Reverse Path Forwarding .

How to Configure MPLS VPN Half-Duplex VRF

- Configuring the Upstream and Downstream VRFs on the Spoke PE Router, page 144
- Associating a VRF with an Interface, page 145
- Configuring the Downstream VRF for an AAA Server, page 146
- Verifying MPLS VPN Half-Duplex VRF Configuration, page 147

Configuring the Upstream and Downstream VRFs on the Spoke PE Router

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **address-family** {**ipv4** | **ipv6**}
6. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
7. **exit-address-family**
8. **end**

DETAILED STEPS

	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	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition vrf1`	Configures a VRF routing table and enters VRF configuration mode. • The *vrf-name* argument is the name of the VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 100:1`	Creates routing and forwarding tables for a VRF. • The *route-distinguisher* argument specifies to add an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter a route distinguisher in either of these formats: ◦ 16-bit autonomous system number (ASN): your 32-bit number For example, 101:3. ◦ 32-bit IP address: your 16-bit number For example, 192.168.122.15:1.

	Command or Action	Purpose
Step 5	**address-family** {**ipv4** | **ipv6**} **Example:** `Router(config-vrf) address-family ipv4`	Enters VRF address family configuration mode to specify an address family for a VRF. • The **ipv4** keyword specifies an IPv4 address family for a VRF. • The **ipv6** keyword specifies an IPv6 address family for a VRF. **Note** The MPLS VPN Half Duplex VRF feature supports only the IPv4 address family.
Step 6	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf-af)# route-target both 100:2`	Creates a route-target extended community for a VRF. • The **import** keyword specifies to import routing information from the target VPN extended community. • The **export** keyword specifies to export routing information to the target VPN extended community. • The **both** keyword specifies to import both import and export routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities.
Step 7	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode.
Step 8	**end** **Example:** `Router(config-vrf)# end`	Exits to privileged EXEC mode.

Associating a VRF with an Interface

Perform the following task to associate a VRF with an interface, which activates the VRF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name* [**downstream** *vrf-name2*
5. **ip address** *ip-address mask* [**secondary**]
6. **end**

DETAILED STEPS

	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 0/1`	Configures an interface type and enters interface configuration mode. • The *type* argument identifies the type of interface to be configured. • The *number* argument identifies the port, connector, or interface card number.
Step 4	**vrf forwarding** *vrf-name* [**downstream** *vrf-name2* **Example:** `Router(config-if)# vrf forwarding vrf1`	Associates a VRF with an interface or subinterface. • The *vrf-name* argument is the name of the VRF. • The **downstream** *vrf-name2* keyword and argument combination is the name of the downstream VRF into which peer and per-user routes are installed.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.24.24.24 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask of the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 6	**end** **Example:** `Router(config-if) end`	Exits to privileged EXEC mode.

Configuring the Downstream VRF for an AAA Server

To configure the downstream VRF for an AAA (RADIUS) server in broadband or remote access situations, enter the following Cisco attribute value:

lcp:interface-config=ip vrf forwarding U downstream D

In standard VPN situations, enter instead the following Cisco attribute value:

ip:vrf-id=U downstream D

Verifying MPLS VPN Half-Duplex VRF Configuration

To verify the Downstream VRF for an AAA Server configuration, perform the following steps.

SUMMARY STEPS

1. **show vrf** [**brief** | **detail** | **id** | **interfaces** | **lock** | **select**] [*vrf-name*]
2. **show ip route vrf** *vrf-name*
3. **show running-config** [**interface** *type number*]

DETAILED STEPS

Step 1 **show vrf** [**brief** | **detail** | **id** | **interfaces** | **lock** | **select**] [*vrf-name*]

Use this command to display information about all of the VRFs configured on the router, including the downstream VRF for each associated interface or VAI:

Example:

```
Router# show vrf
Name     Default RD     Interfaces
Down     100:1          POS3/0/3 [D]
                        POS3/0/1 [D]
         100:3          Loopback2
                        Virtual-Access3 [D]
                        Virtual-Access4 [D]
Up       100:2          POS3/0/3
                        POS3/0/1
         100:4          Virtual-Access3
```

show vrf detail *vrf-name*

Use this command to display detailed information about the VRF you specify, including all interfaces, subinterfaces, and VAIs associated with the VRF.

If you do not specify a value for the vrf-name argument, detailed information about all of the VRFs configured on the router appears.

The following example shows how to display detailed information for the VRF called vrf1, in a broadband or remote access case:

Example:

```
Router# show vrf detail vrf1
VRF D; default RD 2:0; default VPNID <not set>
  Interfaces:
         Loopback2           Virtual-Access3 [D]  Virtual-Access4 [D]
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:2:0
  Import VPN route-target communities
    RT:2:1
```

```
  No import route-map
  No export route-map
VRF U; default RD 2:1; default VPNID <not set>
  Interfaces:
    Virtual-Access3          Virtual-Access4
  Connected addresses are not in global routing table
  No Export VPN route-target communities
  Import VPN route-target communities
    RT:2:1
  No import route-map
  No export route-map
```

The following example shows the VRF detail in a standard VPN situation:

Example:

```
Router# show vrf detail
VRF Down; default RD 100:1; default VPNID <not set> VRF Table ID = 1
  Description: import only from hub-pe
  Interfaces:
    Pos3/0/3 [D]        Pos3/0/1:0.1 [D]
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:100:0
  Import VPN route-target communities
    RT:100:1
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
    VRF Up; default RD 100:2; default VPNID <not set> VRF Table ID = 2
  Interfaces:
    Pos3/0/1            Pos3/0/3
  Connected addresses are not in global routing table
  No Export VPN route-target communities
  Import VPN route-target communities
    RT:100:1
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
```

Step 2 **show ip route vrf** *vrf-name*

Use this command to display the IP routing table for the VRF you specify, and information about the per-user routes installed in the downstream VRF.

The following example shows how to display the routing table for the downstream VRF named D, in a broadband or remote access situation:

Example:

```
Router# show ip route vrf D

Routing Table: D
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS interarea
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is not set
    10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
U         10.0.0.2/32 [1/0] via 10.0.0.1
S         10.0.0.0/8 is directly connected, Null0
U         10.0.0.5/32 [1/0] via 10.0.0.2
C         10.8.1.2/32 is directly connected, Virtual-Access4
C         10.8.1.1/32 is directly connected, Virtual-Access3
```

The following example shows how to display the routing table for the downstream VRF named Down, in a standard VPN situation:

Example:

```
Router# show ip route vrf Down

Routing Table: Down
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is 10.13.13.13 to network 0.0.0.0
C     10.2.0.0/8 is directly connected, Pos3/0/3
     10.3.0.0/32 is subnetted, 1 subnets
B       10.4.16.16 [200/0] via 10.13.13.13, 1w3d
B     10.6.0.0/8 [200/0] via 10.13.13.13, 1w3d
C     10.0.0.0/8 is directly connected, Pos3/0/1
    10.7.0.0/16 is subnetted, 1 subnets
B         10.7.0.0 [20/0] via 10.0.0.2, 1w3d
     10.0.6.0/32 is subnetted, 1 subnets
B       10.0.6.14 [20/0] via 10.0.0.2, 1w3d
     10.8.0.0/32 is subnetted, 1 subnets
B       10.8.15.15 [20/0] via 10.0.0.2, 1w3d
B*   0.0.0.0/0 [200/0] via 10.0.0.13, 1w3d
```

The following example shows how to display the routing table for the upstream VRF named U in a broadband or remote access situation:

Example:

```
Router# show ip route vrf U
Routing Table: U
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS interarea
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is 192.168.0.20 to network 0.0.0.0
    10.0.0.0/32 is subnetted, 1 subnets
C         10.0.0.8 is directly connected, Loopback2
B*   0.0.0.0/0 [200/0] via 192.168.0.20, 1w5d
```

The following example shows how to display the routing table for the upstream VRF named Up in a standard VPN situation:

Example:

```
Router# show ip route vrf Up
Routing Table: Up
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is 10.13.13.13 to network 0.0.0.0
```

```
    10.2.0.0/32 is subnetted, 1 subnets
C          10.2.0.1 is directly connected, Pos3/0/3
     10.3.0.0/32 is subnetted, 1 subnets
B       10.3.16.16 [200/0] via 10.13.13.13, 1w3d
B     10.6.0.0/8 [200/0] via 10.13.13.13, 1w3d
    10.0.0.0/32 is subnetted, 1 subnets
C          10.0.0.1 is directly connected, Pos3/0/1
B*   0.0.0.0/0 [200/0] via 10.13.13.13, 1w3d
```

Step 3 **show running-config** [**interface** *type number*]

Use this command to display information about the interfaceyou specify, including information about the associated upstream and downstream VRFs.

The following example shows how to display information about the subinterface named POS3/0/1:

Example:

```
Router# show running-config interface POS3/0/1
Building configuration...
Current configuration : 4261 bytes
!
interface POS3/0/1
ip vrf forwarding Up downstream Down
ip address 10.0.0.1 255.0.0.0
end
```

Configuration Examples for MPLS VPN Half-Duplex VRF

- Example Configuring the Upstream and Downstream VRFs on the Spoke PE Router, page 150
- Example Associating a VRF with an Interface, page 151
- Example Configuring MPLS VPN Half-Duplex VRF Using Static CE-PE Routing, page 151
- Example Configuring MPLS VPN Half-Duplex VRF Using RADIUS Server and Static CE-PE Routing, page 152
- Example Configuring MPLS VPN Half-Duplex VRF Using Dynamic CE-PE Routing, page 153

Example Configuring the Upstream and Downstream VRFs on the Spoke PE Router

The following example configures an upstream VRF named Up:

```
Router> enable
Router# configure terminal
Router(config)# vrf definition Up
Router(config-vrf)# rd 1:0
Router(config-vrf)# address-family ipv4
Router(config-vrf-af)# route-target import 1:0
Router(config-vrf-af)# exit-address-family
```

The following example configures a downstream VRF named Down:

```
Router> enable
Router# configure terminal
Router(config)# vrf definition Down
```

```
Router(config-vrf)# rd 1:8
Router(config-vrf)# address-family ipv4
Router(config-vrf-af)# route-target import 1:8
Router(config-vrf-af)# exit-address-family
```

Example Associating a VRF with an Interface

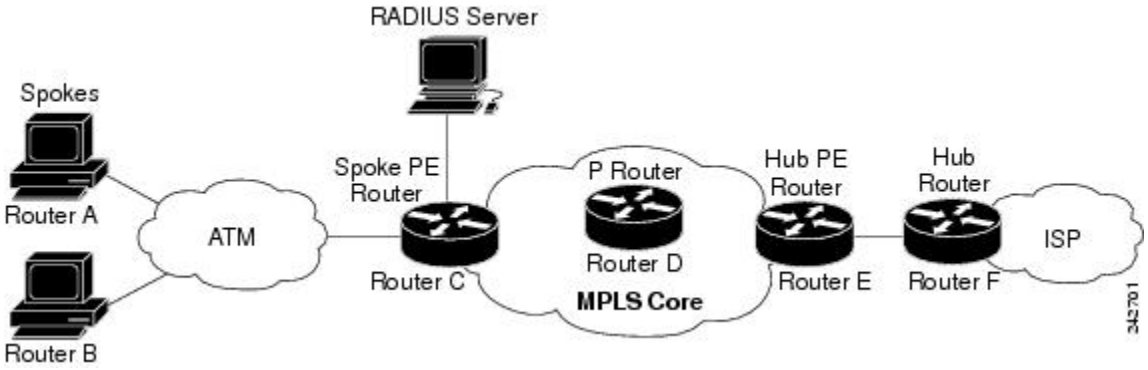
The following example associates the VRF named Up with POS 3/0/1 subinterface and specifies the downstream VRF named Down:

```
Router> enable
Router# configure terminal
Router(config)# interface POS 3/0/1
Router(config-if)# vrf forwarding Up downstream Down
Router(config-if)# ip address 10.0.0.1 255.0.0.0
```

Example Configuring MPLS VPN Half-Duplex VRF Using Static CE-PE Routing

This example uses the hub-and-spoke topology shown in the figure below with local authentication (that is, the RADIUS server is not used):

Figure 8 Sample Topology




```
vrf definition D
 rd 1:8
 address-family ipv4
 route-target export 1:100
 exit-address-family
!
vrf definition U
 rd 1:0
 address-family ipv4
 route-target import 1:0
 exit-address-family
!
ip cef
vpdn enable
!
vpdn-group U
 accept-dialin
  protocol pppoe
  virtual-template 1
!
interface Loopback 2
 vrf forwarding U
```

```
 ip address 10.0.0.8 255.255.255.255
!
interface ATM 2/0
 description Mze ATM3/1/2
 no ip address
 no atm ilmi-keepalive
 pvc 0/16 ilmi
!
 pvc 3/100
  protocol pppoe
!
pvc 3/101
  protocol pppoe
!
```

Example Configuring MPLS VPN Half-Duplex VRF Using RADIUS Server and Static CE-PE Routing

The following example shows how to connect two Point-to-Point Protocol over Ethernet (PPPoE) clients to a single VRF pair on the spoke PE router named Router C. Although both PPPoE clients are configured in the same VRF, all communication occurs using the hub PE router. Half-duplex VRFs are configured on the spoke PE. The client configuration is downloaded to the spoke PE from the RADIUS server.

This example uses the hub-and-spoke topology shown in the figure above.

Note The wholesale provider can forward the user authentication request to the corresponding ISP. If the ISP authenticates the user, the wholesale provider appends the VRF information to the request that goes back to the PE router.

```
aaa new-model
!
aaa group server radius R
 server 10.0.20.26 auth-port 1812 acct-port 1813
!
aaa authentication ppp default group radius
aaa authorization network default group radius
!
vrf defintion D
 description Downstream VRF - to spokes
 rd 1:8
 address-family ipv4
 route-target export 1:100
 exit-address-family
!
vrf definition U
 description Upstream VRF - to hub
 rd 1:0
 address-family ipv4
 route-target import 1:0
 exit-address-family
!
ip cef
vpdn enable
!
vpdn-group U
 accept-dialin
  protocol pppoe
  virtual-template 1
!
interface Loopback2
 vrf forwarding U
 ip address 10.0.0.8 255.255.255.255
!
interface ATM2/0
```

```
  pvc 3/100
  protocol pppoe
 !
pvc 3/101
  protocol pppoe
 !
interface virtual-template 1
 no ip address
 ppp authentication chap
!
router bgp 1
 no synchronization
 neighbor 172.16.0.34 remote-as 1
 neighbor 172.16.0.34 update-source Loopback0
 no auto-summary
 !
address-family vpnv4
  neighbor 172.16.0.34 activate
  neighbor 172.16.0.34 send-community extended
  auto-summary
  exit-address-family
 !
address-family ipv4 vrf U
  no auto-summary
  no synchronization
  exit-address-family
!
address-family ipv4 vrf D
  redistribute static
  no auto-summary
  no synchronization
  exit-address-family
!
ip local pool U-pool 10.8.1.1 2.8.1.100
ip route vrf D 10.0.0.0 255.0.0.0 Null0
!
radius-server host 10.0.20.26 auth-port 1812 acct-port 1813
radius-server key cisco
```

Example Configuring MPLS VPN Half-Duplex VRF Using Dynamic CE-PE Routing

The following example shows how to use OSPF to dynamically advertise the routes on the spoke sites.

This example uses the hub-and-spoke topology shown in the figure above.

Creating the VRFs

```
vrf definition Down
rd 100:1
address-family ipv4
route-target export 100:0
exit-address-family
!
vrf definition Up
rd 100:2
address-family ipv4
route-target import 100:1
exit-address-family
```

Enabling MPLS

```
mpls ldp graceful-restart
mpls ldp router-id Loopback0 force
mpls label protocol ldp
```

Configuring BGP Toward Core

```
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 10.13.13.13 remote-as 100
 neighbor 10.13.13.13 update-source Loopback0
 !
 address-family vpnv4
 neighbor 10.13.13.13 activate
 neighbor 10.13.13.13 send-community extended
 bgp scan-time import 5
 exit-address-family
```

Configuring BGP Toward Edge

```
address-family ipv4 vrf Up
no auto-summary
no synchronization
exit-address-family
!
address-family ipv4 vrf Down
redistribute ospf 1000 vrf Down
no auto-summary
no synchronization
exit-address-family
```

Spoke PE's Core-Facing Interfaces and Processes

```
interface Loopback 0
 ip address 10.11.11.11 255.255.255.255
!
interface POS 3/0/2
 ip address 10.0.1.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
!
router ospf 100
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 network 10.11.11.11 0.0.0.0 area 100
 network 10.0.1.0 0.255.255.255 area 100
```

Spoke PE's Edge-Facing Interfaces and Processes

```
interface Loopback 100
vrf forwarding Down
 ip address 10.22.22.22 255.255.255.255
!
interface POS 3/0/1
vrf forwarding Up downstream Down
 ip address 10.0.0.1 255.0.0.0
!
interface POS 3/0/3
vrf forwarding Up downstream Down
 ip address 10.2.0.1 255.0.0.0
!
router ospf 1000 vrf Down
 router-id 10.22.22.22
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
```

```
redistribute bgp 100 metric-type 1 subnets
network 10.22.22.22 0.0.0.0 area 300
network 10.0.0.0 0.255.255.255 area 300
network 10.2.0.0 0.255.255.255 area 300
default-information originate
```

Additional References

Related Documents

Related Topic	Document Title
MPLS VPNs	Configuring MPLS Layer 3 VPNs
MPLS commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
Configuring IPv4 and IPv6 VRFs	MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs
Unicast Reverse Path Forwarding	Configuring Unicast Reverse Path Forwarding

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 software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2547	BGP/MPLS VPNs

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 MPLS VPN Half-Duplex VRF

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 10 Feature Information for MPLS VPN Half-Duplex VRF

Feature Name	Releases	Feature Information
MPLS VPN - Half Duplex VRF (HDVRF) Support with Static Routing	12.3(6) 12.3(11)T 12.2(28)SB	This feature ensures that VPN clients that connect to the same PE router at the edge of the MPLS VPN use the hub site to communicate. In 12.3(6), this feature was introduced. In 12.4(20)T, this feature was integrated. In 12.2(28)SB, this feature was integrated

Feature Name	Releases	Feature Information
MPLS VPN Half-Duplex VRF	12.2(28)SB2 12.4(20)T 12.2(33)SRC	In 12.2(28)SB2, support for dynamic routing protocols was added. For the Cisco 10000 series routers, see the "Half-Duplex VRF" section of the "Configuring Multiprotocol Label Switching" chapter in the Cisco 10000 Series Router Broadband Aggregation, Leased-Line, and MPLS Configuration Guide at the following URL: http://www.cisco.com/univercd/cc/td/doc/product/aggr/10000/swconfig/cfggdes/bba/dffsrv.htm#wp1065648 In 12.4(20)T, this feature, with support for dynamic routing protocols, was integrated. In Cisco IOS Release 12.2(33)SRC this feature, with support for dynamic routing protocols, was integrated into the SR train. The following commands were introduced or modified: **show ip interface**, **show vrf**

MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs

This document describes how to configure a Virtual Private Network (VPN) routing and forwarding (VRF) instance for IPv4 and IPv6 VPNs and describes how to upgrade your existing single-protocol IPv4-only VRF to a multiprotocol VRF configuration.

The MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs feature introduces Cisco IOS command-line interface (CLI) commands that allow you to enable an IPv4 and IPv6 VPN in the same VRF instance and to simplify the migration from a single-protocol VRF configuration to a multiprotocol VRF configuration.

- Finding Feature Information, page 159
- Prerequisites for MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs, page 159
- Restrictions for MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs, page 160
- Information About MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs, page 160
- How to Configure MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs, page 162
- Configuration Examples for MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs, page 170
- Additional References, page 174
- Feature Information for MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs, page 175
- Glossary, page 176

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs

- For migration--An IPv4 Multiprotocol Label Switching (MPLS) VPN VRF must exist.
- For a new VRF configuration--Cisco Express Forwarding and an MPLS label distribution method, either Label Distribution Protocol (LDP) or MPLS traffic engineering (TE), must be enabled on all routers in the core, including the provider edge (PE) routers.

Restrictions for MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs

- Once you have converted to a multiprotocol VRF, you cannot convert the VRF back to an IPv4-only single-protocol VRF.
- You can associate an interface with only one VRF. You cannot configure a VRF for IPv4 and a different VRF for IPv6 on the same interface.
- You can configure only IPv4 and IPv6 address families in a multiprotocol VRF. Other protocols (IPX, AppleTalk, and the like) are not supported.

Information About MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs

- VRF Concepts Similar for IPv4 and IPv6 MPLS VPNs, page 160
- Single-Protocol VRF to Multiprotocol VRF Migration, page 160
- Multiprotocol VRF Configurations Characteristics, page 161

VRF Concepts Similar for IPv4 and IPv6 MPLS VPNs

VPNs for IPv6 use the same VRF concepts that IPv4 MPLS VPNs use, such as address families, route distinguishers, route targets, and VRF identifiers. Customers that use both IPv4 and IPv6 VPNs might want to share VRF policies between address families. They might want a way to define applicable VRF policies for all address families, instead of defining VRF policies for an address family individually as they do for or a single-protocol IPv4-only VRF.

Prior to Cisco IOS Release 12.2(33)SRB, a VRF applied only to an IPv4 address family. A one-to-one relationship existed between the VRF name and a routing and forwarding table identifier, between a VRF name and a route distinguisher (RD), and between a VRF name and a VPN ID. This configuration is called a single-protocol VRF.

Cisco IOS Release 12.2(33)SRB introduces support for a multiple address-family (multi-AF) VRF structure. The multi-AF VRF allows you to define multiple address families under the same VRF. A given VRF, identified by its name and a set of policies, can apply to both an IPv4 VPN and an IPv6 VPN at the same time. This VRF can be activated on a given interface, even though the routing and forwarding tables are different for the IPv4 and IPv6 protocols. This configuration is called a multiprotocol VRF.

Single-Protocol VRF to Multiprotocol VRF Migration

Prior to Cisco IOS Release 12.2(33)SRB, you could create a single-protocol IPv4-only VRF. You created a single-protocol VRF by entering the **ip vrf** command. To activate the single-protocol VRF on an interface, you entered the **ip vrf forwarding** (interface configuration) command.

After the introduction of the MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs feature in Cisco IOS Release 12.2(33)SRB, you create a multiprotocol VRF by entering the **vrf definition** command. To activate the multiprotocol VRF on an interface, you enter the **vrf forwarding** command.

The MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs feature introduces the **vrf upgrade-cli multi-af-mode** {**common-policies** | **non-common-policies**} [**vrf** *vrf-name*] command that forces VRF configuration migration from a single-protocol VRF model to a multiprotocol VRF model:

- If the route-target policies apply to all address families configured in the multi-AF VRF, use the **common-policies** keyword.

- If the route-target policies apply only to the IPv4 address family that you are migrating, use the **non-common-policies** keyword.

After you enter the **vrf upgrade-cli** command and save the configuration to NVRAM, the single-protocol VRF configuration is saved as a multiprotocol VRF configuration. In the upgrade process, the**ip vrf** command is converted to the **vrf definition** command (global configuration commands) and the **ip vrf forwarding** command is converted to the **vrf forwarding** command (interface configuration command). The **vrf upgrade-cli** command has a one-time immediate effect.

You might have both IPv4-only VRFs and multiprotocol VRFs on your router. Once you create a VRF, you can edit it using only the commands in the mode in which it was created. For example, you created a VRF named vrf2 with the following multiprotocol VRF commands:

```
Router# configure terminal
Enter configuration command, one per line. End with CNTL/Z
Router(config)# vrf definition vrf2
Router(config-vrf)# rd 2:2
Router(config-vrf)# route-target import 2:2
Router(config-vrf)# route-target export 2:2
Router(config-vrf)# end
```

If you try to edit VRF vrf2 with IPv4-only VRF commands, you receive the following message:

```
Router# configure terminal
Enter configuration command, one per line. End with CNTL/Z
Router(config)# ip vrf vrf2
% Use ‘vrf definition vrf2’ command
```

If you try to edit an IPv4-only VRF with the multiprotocol VRF commands, you would receive this message, where <vrf-name> is the name of the IPv4-only VRF:

```
% Use ‘ip vrf <vrf-name>’ command
```

The**ip vrf** *name* and **ip vrf forwarding** (interface configuration) *name* commands will be available for a period of time before they are removed. Use the **vrf upgrade-cli** command to migrate your older IPv4-only VRFs to the new multiprotocol VRF configuration. When you need to create a new VRF--whether the VRF is for an IPv4 VPN, or IPv6 VPN, or both--use the multiprotocol VRF **vrf definition** and **vrf forwarding** commands that support a multi-AF configuration.

Multiprotocol VRF Configurations Characteristics

In a multiprotocol VRF, you can configure both IPv4 VRFs and IPv6 VRFs under the same address family or configure separate VRFs for each IPv4 or IPv6 address family. The multiprotocol VRF configuration has the following characteristics:

- The VRF name identifies a VRF, which might have both IPv4 and IPv6 address families. On the same interface, you cannot have IPv4 and IPv6 address families using different VRF names.
- The RD, VPN ID, and Simple Network Management Protocol (SNMP) context are shared by both IPv4 and IPv6 address families for a given VRF.
- The policies (route target, for example) specified in multi-AF VRF mode, outside the address-family configuration, are defaults to be applied to each address family. Route targets are the only VRF characteristics that can be defined inside and outside an address family.

The following is also true when you associate a multiprotocol VRF with an interface:

- Binding an interface to a VRF (**vrf forwarding** *vrf-name* command) removes all IPv4 and IPv6 addresses configured on that interface.

- Once you associate a VRF with a given interface, all active address families belong to that VRF. The exception is when no address of the address-family type is configured, in which case the protocol is disabled.
- Configuring an address on an interface that is bound to a VRF requires that the address family corresponding to the address type is active for that VRF. Otherwise, an error message is issued stating that the address family must be activated first in the VRF.

Backward compatibility with the single-protocol VRF CLI is supported in Cisco IOS Release 12.2(33)SRB. This means that you might have single-protocol and multiprotocol CLI on the same router, but not in the same VRF configuration.

The single-protocol CLI continues to allow you to define an IPv4 address within a VRF and an IPv6 address in the global routing table on the same interface.

How to Configure MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs

This feature provides Cisco IOS CLI commands that allow you to configure a multiprotocol VRF (IPv4 and IPv6 VPNs in the same VRF) and to migrate a single-protocol VRF configuration (IPv4-only VRF) to a multiprotocol VRF configuration.

A multiprotocol VRF allows you to share route targets policies (import and export) between IPv4 and IPv6 or to configure separate route-target policies for IPv4 and IPv6 VPNs.

- Configuring a VRF for IPv4 and IPv6 MPLS VPNs, page 162
- Associating a Multiprotocol VRF with an Interface, page 164
- Verifying the MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs Configuration, page 166
- Migrating from a Single-Protocol IPv4-Only VRF to a Multiprotocol VRF Configuration, page 169

Configuring a VRF for IPv4 and IPv6 MPLS VPNs

Perform the following task to configure a VRF for IPv4 and IPv6 MPLS VPNs. When you configure a VRF for both IPv4 and IPv6 VPNs (a multiprotocol VRF), you can choose to configure route-target policies that apply to all address families in the VRF or you can configure route-target policies that apply to individual address families in the VRF.

The following task shows how to configure a VRF that has that has route-target policies defined for IPv4 and IPv6 VPNs in separate VRF address families.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **address-family** {**ipv4** | **ipv6**}
6. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
7. **exit-address-family**
8. **address-family** {**ipv4** | **ipv6**}
9. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
10. **end**

DETAILED STEPS

	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	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition vrf1`	Configures a VRF routing table and enters VRF configuration mode. • The *vrf-name* argument is the name of the VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 100:1`	Creates routing and forwarding tables for a VRF. • The *route-distinguisher* argument specifies to add an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter a route distinguisher in either of these formats: ◦ 16-bit autonomous system number (ASN): your 32-bit number For example, 101:3. ◦ 32-bit IP address: your 16-bit number For example, 192.168.122.15:1.
Step 5	**address-family** {**ipv4** | **ipv6**} **Example:** `Router(config-vrf) address-family ipv4`	Enters VRF address family configuration mode to specify an address family for a VRF. • The **ipv4** keyword specifies an IPv4 address family for a VRF. • The **ipv6** keyword specifies an IPv6 address family for a VRF.
Step 6	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf-af)# route-target both 100:2`	Creates a route-target extended community for a VRF. • The **import** keyword specifies to import routing information from the target VPN extended community. • The **export** keyword specifies to export routing information to the target VPN extended community. • The **both** keyword specifies to import both import and export routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities.

	Command or Action	Purpose
Step 7	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits from VRF address family configuration mode.
Step 8	**address-family** {**ipv4** | **ipv6**} **Example:** `Router(config-vrf) address-family ipv6`	Enters VRF address family configuration mode to specify an address family for a VRF. • The **ipv4** keyword specifies an IPv4 address family for a VRF. • The **ipv6** keyword specifies an IPv6 address family for a VRF.
Step 9	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf-af)# route-target both 100:3`	Creates a route-target extended community for a VRF. • The **import** keyword specifies to import routing information from the target VPN extended community. • The **export** keyword specifies to export routing information to the target VPN extended community. • The **both** keyword specifies to import both import and export routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities. Enter the **route-target** command one time for each target community.
Step 10	**end** **Example:** `Router(config-vrf-af)# end`	Exits to privileged EXEC mode.

Associating a Multiprotocol VRF with an Interface

Perform the following task to associate a multiprotocol VRF with an interface. Associating the VRF with an interface activates the VRF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name*
5. **ip address** *ip-addressmask* [**secondary**]
6. **ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*}
7. **end**

DETAILED STEPS

	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 0/1`	Configures an interface type and enters interface configuration mode. • The *type* argument identifies the type of interface to be configured. • The *number* argument identifies the port, connector, or interface card number.
Step 4	**vrf forwarding** *vrf-name* **Example:** `Router(config-if)# vrf forwarding vrf1`	Associates a VRF with an interface or subinterface. • The *vrf-name* argument is the name of the VRF.
Step 5	**ip address** *ip-addressmask* [**secondary**] **Example:** `Router(config-if)# ip address 10.24.24.24 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask of the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.

	Command or Action	Purpose
Step 6	**ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*} **Example:** `Router(config-if)# ipv6 address 2001:0DB8:0300:0201::/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface. • The *ipv6-address* argument is the IPv6 address to be used. • The *prefix-length* argument is the length of the IPv6 prefix, which is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value. • The *prefix-name* argument is a general prefix that specifies the leading bits of the network to be configured on the interface. • The *sub-bits* argument is the subprefix bits and host bits of the address to be concatenated with the prefixes provided by the general prefix specified with the *prefix-name* argument. The *sub-bits* argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
Step 7	**end** **Example:** `Router(config-if) end`	Exits to privileged EXEC mode.

Verifying the MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs Configuration

Perform the following task to verify the MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs feature configuration, that is, to show that the VRF configuration is upgraded to a multi-AF multiprotocol VRF.

SUMMARY STEPS

1. **enable**
2. **show running-config vrf** [*vrf-name*]
3. **show vrf**
4. **show vrf detail** [*vrf-name*]
5. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password, if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 **show running-config vrf** [*vrf-name*]

Use this command to verify that the upgrade to a multi-AF multiprotocol VRF configuration was successful. The following is sample command output before the upgrade to a multi-AF multiprotocol VRF:

Example:

```
Router# show running-config vrf vpn2
Building configuration...
Current configuration : 604 bytes
ip vrf vpn2
 rd 1:1
 route-target both 1:1
!
!
interface Loopback1
 ip vrf forwarding vpn2
 ip address 10.43.43.43 255.255.255.255
!
```

The following is sample command output after you upgrade to a multi-AF multiprotocol VRF with common policies for all address families:

Example:

```
Router# show running-config vrf vpn1
Building configuration...
Current configuration : 604 bytes
vrf definition vpn1
 rd 1:1
 route-target both 1:1
!
 address-family 1pv4
 exit-address-family
!
!
interface Loopback1
 ip vrf forwarding vpn1
 ip address 10.43.43.43 255.255.255.255
!
```

This configuration contains the **vrf definition** command. The **vrf definition** command replaces the **ip vrf** command in the multi-AF multiprotocol VRF configuration.

Step 3 **show vrf**

Use this command to verify that the upgrade to a multi-AF multiprotocol VRF configuration was successful. The **show vrf** command replaces the **show ip vrf** command when a VRF configuration is updated to a multi-AF multiprotocol VRF configuration. The**show vrf** command displays the protocols defined for a VRF. The following command shows sample output after you upgrade a single-protocol VRF configuration to a multi-AF multiprotocol VRF configuration:

Example:

```
Router# show vrf vpn1
  Name                             Default RD     Protocols         Interfaces
  vpn1                             1:1            ipv4              Lo1/0
```

The following is sample output from the **show ip vrf vp1** command. Compare this to the output of the **show vrf vpn1** command. The protocols under the VRF are not displayed.

Example:

```
Router# show ip vrf vrf1
  Name      Default RD   Interface
  vpn1      1:1          Loopback1
```

The following is sample output from the **show vrf** command for multiprotocol VRFs, one of which contains both IPv4 and IPv6 protocols:

Example:

```
Router# show vrf
  Name                             Default RD     Protocols         Interfaces
  vpn1                             1:1            ipv4              Lo1/0
  vpn2                             100:3          ipv4              Lo23  AT3/0/0.1
  vpn4                             100:2          ipv4,ipv6
```

Step 4 **show vrf detail** [*vrf-name*]

Use this command to display all characteristics of the defined VRF to verify that the configuration is as you expected. For example, if your VRF configuration for VRF vpn1 is as follows:

Example:

```
vrf definition vpn1
 route-target both 100:1
 route-target import 100:2
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 route-target both 100:1
 route-target import 100:3
 exit-address-family
```

This command would display the following:

Example:

```
Router# show vrf detail vpn1
VRF vpn1 (VRF Id = 3); default RD <not set>; default VPNID <not set>
  No interfaces
Address family ipv4 (Table ID = 3 (0x3)):
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:100:1
  Import VPN route-target communities
    RT:100:1                RT:100:2
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
Address family ipv6 (Table ID = 503316483 (0x1E000003)):
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:100:1
  Import VPN route-target communities
    RT:100:1                RT:100:3
```

```
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
```

Step 5 **exit**

Use this command to exit to user EXEC mode. For example:

Example:

```
Router# exit
Router>
```

Migrating from a Single-Protocol IPv4-Only VRF to a Multiprotocol VRF Configuration

Perform the following task to force migration from a single-protocol IPv4-only VRF to a multiprotocol VRF configuration.

The multiprotocol VRF configuration allows you to define multiple address families under the same VRF. A given VRF, identified by its name and a set of policies, can apply to both an IPv4 VPN and an IPv6 VPN at the same time. This VRF can be activated on a given interface, even though the routing and forwarding tables are different for the IPv4 and IPv6 protocols.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf upgrade-cli multi-af-mode** {**common-policies** | **non-common-policies**} [**vrf** *vrf-name*]
4. **exit**
5. **show running-config vrf** [*vrf-name*]

DETAILED STEPS

	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	**vrf upgrade-cli multi-af-mode** {**common-policies** | **non-common-policies**} [**vrf** *vrf-name*] **Example:** `Router(config)# vrf upgrade-cli multi-af-mode common-policies vrf vpn4`	Upgrades a VRF instance or all VRFs configured on the router to support multiple address families under the same VRF. • The **multi-af-mode** keyword specifies an upgrade of a single-protocol VRF or all VRFs to a multiprotocol VRF that supports multi-AFs configuration. • The **common-policies** keyword specifies to copy the route-target policies to the common part of the VRF configuration so that the policies apply to all address families configured in the multi-AF VRF. • The **non-common-policies** keyword specifies to copy the route-target policies to the IPv4 address family part of the VRF configuration so that the policies apply only to IPv4. • The **vrf** keyword specifies a VRF for the upgrade to a multi-AF VRF configuration. • The *vrf-name* argument is the name of the single-protocol VRF to upgrade to a multi-AF VRF configuration.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits to privileged EXEC mode.
Step 5	**show running-config vrf** [*vrf-name*] **Example:** `Router# show running-config vrf vpn4`	Displays the subset of the running configuration of a router that is linked to a specific VRF instance or to all VRFs configured on the router. • The *vrf-name* argument is the name of the VRF of which you want to display the configuration. **Note** The Cisco IOS image that supports the multiprotocol VRF commands might not support the **show running-config vrf** command. You can use the **show running-config** command instead.

Configuration Examples for MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs

- Example Multiprotocol VRF Configuration Single Protocol with Noncommon Policies, page 171
- Example Multiprotocol VRF Configuration Multiprotocol with Noncommon Policies, page 171
- Example Multiprotocol VRF Configuration Multiprotocol with Common Policies, page 171
- Example Multiprotocol VRF Configuration Multiprotocol with Common and Noncommon Policies, page 172
- Example Configuring a VRF for IPv4 and IPv6 VPNs, page 172
- Example Associating a Multiprotocol VRF with an Interface, page 173
- Example Migrating from a Single-Protocol IPv4-Only VRF Configuration to a Multiprotocol VRF Configuration, page 173

Example Multiprotocol VRF Configuration Single Protocol with Noncommon Policies

The following is an example of a multiprotocol VRF configuration for a single protocol (IPv4) with route-target policies in the address family configuration:

```
vrf definition vrf2
 rd 2:2
 !
 address-family ipv4
 route-target export 2:2
 route-target import 2:2
 exit-address-family
```

The RD (2:2) applies to all address families defined for VRF vrf2.

Example Multiprotocol VRF Configuration Multiprotocol with Noncommon Policies

The following is an example of a multiprotocol VRF configuration for IPv4 and IPv6 VPNs in which the route-target policies are defined in the separate address family configurations:

```
vrf definition vrf2
 rd 2:2
 !
 address-family ipv4
 route-target export 2:2
 route-target import 2:2
 exit-address-family
 !
 address-family ipv6
 route-target export 3:3
 route-target import 3:3
 exit-address-family
```

Example Multiprotocol VRF Configuration Multiprotocol with Common Policies

The following is an example of a multiprotocol VRF configuration for IPv4 and IPv6 VPNs with route-target policies defined in the global part of the VRF:

```
vrf definition vrf2
 rd 2:2
 route-target export 2:2
 route-target import 2:2
  !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
```

The route-target policies are defined outside the address family configurations. Therefore, the policies apply to all address families defined in VRF vrf2.

Example Multiprotocol VRF Configuration Multiprotocol with Common and Noncommon Policies

The following is an example of a multiprotocol VRF with route-target policies defined in both global and address family areas:

- For IPv6, the route-target definitions are defined under the address family. These definitions are used and the route-target definitions in the global area are ignored. Therefore, the IPv6 VPN ignores import 100:2.
- For IPv4, no route-target policies are defined under the address family, therefore, the global definitions are used.

```
vrf definition vfr1
 route-target export 100:1
 route-target import 100:1
 route-target import 100:2
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 route-target export 100:1
 route-target import 100:1
 route-target import 100:3
 exit-address-family
```

Example Configuring a VRF for IPv4 and IPv6 VPNs

The following example shows how to configure a VRF for IPv4 and IPv6 VPNs:

```
configure terminal
!
vrf definition vrf1
 rd 100:1
!
 address-family ipv4
 route-target both 100:2
 exit-address-family
!
 address-family ipv6
 route-target both 100:3
 exit-address-family
```

In this example, noncommon policies are defined in the address family configuration.

The following is an example of a VRF for IPv4 and IPv6 that has common policies defined in the global part of the VRF configuration:

```
configure terminal
!
vrf definition vrf2
 rd 200:1
 route-target both 200:2
!
 address-family ipv4
 exit-address-family
!
 address-family ipv6
 exit-address-family
 end
```

Example Associating a Multiprotocol VRF with an Interface

The following example shows how to associate a multiprotocol VRF with an interface:

```
configure terminal
!
interface Ethernet 0/1
 vrf forwarding vrf1
 ip address 10.24.24.24 255.255.255.255
 ipv6 address 2001:0DB8:0300:0201::/64
 end
```

Example Migrating from a Single-Protocol IPv4-Only VRF Configuration to a Multiprotocol VRF Configuration

This section contains examples that show how to migrate from a single-protocol IPv4-only VRF to a multiprotocol VRF configuration.

This example shows a single-protocol IPv4-only VRF before the Cisco IOS VRF CLI for IPv4 and IPv6 is entered on the router:

```
ip vrf vrf1
 rd 1:1
 route-target both 1:1
interface Loopback1
 ip vrf forwarding V1
 ip address 10.3.3.3 255.255.255.255
```

This example shows how to force the migration of the single-protocol VRF vrf1 to a multiprotocol VRF configuration:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
!
Router(config)# vrf upgrade-cli multi-af-mode common-policies vrf vrf1
You are about to upgrade to the multi-AF VRF syntax commands.
You will loose any IPv6 address configured on interfaces
belonging to upgraded VRFs.
Are you sure ? [yes]: yes
Number of VRFs upgraded: 1
Router(config)# exit
```

This example shows the multiprotocol VRF configuration after the forced migration:

```
vrf definition vrf1
 rd 1:1
 route-target both 1:1
 !
 address-family ipv4
 exit-address-family
!
interface Loopback1
 vrf forwarding V1
 ip address 10.3.3.3 255.255.255.255
```

The following is another example of a multi-AF multiprotocol VRF configuration:

```
vrf definition vrf2
 rd 100:1
 address family ipv6
 route-target both 200:1
 exit-address-family
!
```

```
ip vrf vrf1
 rd 200:1
 route-target both 200:1
!
interface Ethernet0/0
 vrf forwarding vrf2
 ip address 10.50.1.2 255.255.255.0
 ipv6 address 2001:0DB8:0:1::/64
!
interface Ethernet0/1
 ip vrf forwarding vrf1
 ip address 10.60.1.2 255.255.255.0
 ipv6 address 2001:0DB8:1 :1::/64
```

In this example, all addresses (IPv4 and IPv6) defined for interface Ethernet0/0 are in VRF vrf2. For the interface Ethernet0/1, the IPv4 address is defined in VRF vrf1 but the IPv6 address is in the global IPv6 routing table.

Additional References

Related Documents

Related Topic	Document Title
MPLS	MPLS Product Literature
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Commands for configuring MPLS and MPLS VPNs	*Cisco IOS Multiprotocol Label Switching 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 software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1771	*A Border Gateway Protocol 4 (BGP-4)*
RFC 4364	*BGP MPLS/IP Virtual Private Networks (VPNs)*

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--VRF CLI for IPv4 and IPv6 VPNs

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 11 Feature Information for MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs

Feature Name	Releases	Feature Information
MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs	12.2(33)SRB 12.2(33)SXI	This document describes how to configure a multiprotocol Virtual Private Network (VPN) routing and forwarding (VRF) instance for IPv4 and IPv6 VPNs and describes how to upgrade your existing single-protocol IPv4-only VRF to a multiprotocol VRF configuration. The MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs feature introduces Cisco IOS command-line interface (CLI) commands that allow you to enable an IPv4 and IPv6 VPN in the same Multiprotocol Label Switching (MPLS) VRF instance and to simplify the migration from a single-protocol VRF configuration to a multiprotocol VRF configuration. In 12.2(33)SRB, this feature was introduced on the Cisco 7600 router. In 12.2(33)SXI, this feature was integrated into a Cisco IOS 12.2SXI release.
		The following commands were introduced or modified: **show vrf**, **vrf definition**, **vrf forwarding**, **vrf upgrade-cli**.

Glossary

6PE --IPv6 provider edge router or a Multiprotocol Label Switching (MPLS) label switch router (LSR) edge router using IPv6.

6VPE --IPv6 Virtual Private Network (VPN) provider edge router.

AF --address family. Set of related communication protocols in which all members use a common addressing mechanism to identify endpoints. Also called protocol family.

AFI --Address Family Identifier. Carries the identity of the network-layer protocol that is associated with the network address.

BGP --Border Gateway Protocol. A routing protocol used between autonomous systems. It is the routing protocol that makes the internet work. BGP is a distance-vector routing protocol that carries connectivity

information and an additional set of BGP attributes. These attributes allow for a set of policies for deciding the best route to use to reach a given destination. BGP is defined by RFC 1771.

CE --customer edge router. A service provider router that connects to Virtual Private Network (VPN) customer sites.

FIB --Forwarding Information Base. Database that stores information about switching of data packets. A FIB is based on information in the Routing Information Base (RIB). It is the optimal set of selected routes that are installed in the line cards for forwarding.

HA --high availability. High availability is defined as the continuous operation of systems. For a system to be available, all components--including application and database servers, storage devices, and the end-to-end network--need to provide continuous service.

IP --Internet Protocol. Network-layer protocol in the TCP/IP stack offering a connectionless internetwork service. IP provides features for addressing, type-of-service specification, fragmentation and reassembly, and security.

IPv4 --IP Version 4. Network layer for the TCP/IP protocol suite. IPv4 is a connectionless, best-effort packet switching protocol.

IPv6 --IP Version 6. Replacement for IPv4. IPv6 is a next-generation IP protocol. IPv6 is backward compatible with and designed to fix the shortcomings of IPv4, such as data security and maximum number of user addresses. IPv6 increases the address space from 32 to 128 bits, providing for an unlimited number of networks and systems. It also supports quality of service (QoS) parameters for real-time audio and video.

MFI --MPLS Forwarding Infrastructure. In the Cisco MPLS subsystem, the data structure for storing information about incoming and outgoing labels and associated equivalent packets suitable for labeling.

MPLS --Multiprotocol Label Switching. MPLS is a method for forwarding packets (frames) through a network. It enables routers at the edge of a network to apply labels to packets (frames). ATM switches or existing routers in the network core can switch packets according to the labels with minimal lookup overhead.

PE --provider edge router. A router that is part of a service provider's network and that is connected to a customer edge (CE) router. The PE router function is a combination of an MLS edge label switch router (LSR) function with some additional functions to support Virtual Private Networks (VPNs).

RD (IPv4)--route distinguisher. An 8-byte value that is concatenated with an IPv4 prefix to create a unique VPN IPv4 (VPNv4) prefix.

RD (IPv6)--route distinguisher. A 64-bit value that is prepended to an IPv6 prefix to create a globally unique VPN-IPv6 address.

RIB --Routing Information Base. The set of all available routes from which to choose the Forwarding Information Base (FIB). The RIB essentially contains all routes available for selection. It is the sum of all routes learned by dynamic routing protocols, all directly attached networks (that is-networks to which a given router has interfaces connected), and any additional configured routes, such as static routes.

RT --route target. Extended community attribute used to identify the Virtual Private Network (VPN) routing and forwarding (VRF) routing table into which a prefix is to be imported.

VPN --Virtual Private Network. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses "tunneling" to encrypt all information at the IP level.

VRF --Virtual Private Network (VPN) routing and forwarding instance. A VRF 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 what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

VRF table --A routing and a forwarding table associated to a Virtual Private Network (VPN) routing and forwarding (VRF) instance. This is a customer-specific table, enabling the provider edge (PE) router to maintain independent routing states for each customer.

MPLS VPN--Route Target Rewrite

The MPLS VPN--Route Target Rewrite feature allows the replacement of route targets on incoming and outgoing Border Gateway Protocol (BGP) updates. Typically, Autonomous System Border Routers (ASBRs) perform the replacement of route targets at autonomous system boundaries. Route Reflectors (RRs) and provider edge (PE) routers can also perform route target replacement.

The main advantage of the MPLS VPN--Route Target Rewrite feature is that it keeps the administration of routing policy local to the autonomous system.

- Finding Feature Information, page 179
- Prerequisites for MPLS VPN--Route Target Rewrite, page 179
- Restrictions for MPLS VPN--Route Target Rewrite, page 180
- Information About MPLS VPN--Route Target Rewrite, page 180
- How to Configure MPLS VPN--Route Target Rewrite, page 181
- Configuration Examples for MPLS VPN--Route Target Rewrite, page 192
- Additional References, page 194
- Feature Information for MPLS VPN--Route Target Rewrite, page 195
- Glossary, page 196

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for MPLS VPN--Route Target Rewrite

- You should know how to configure Multiprotocol Virtual Private Networks (MPLS VPNs).
- You need to configure your network to support interautonomous systems with different route target (RT) values in each autonomous system.
- You need to identify the RT replacement policy and target router for each autonomous system.

Restrictions for MPLS VPN--Route Target Rewrite

You can apply multiple replacement rules using the route-map continue clause. The MPLS VPN--Route Target Rewrite feature does not support the continue clause on outbound route maps.

Information About MPLS VPN--Route Target Rewrite

- Route Target Replacement Policy, page 180
- Route Maps and Route Target Replacement, page 181

Route Target Replacement Policy

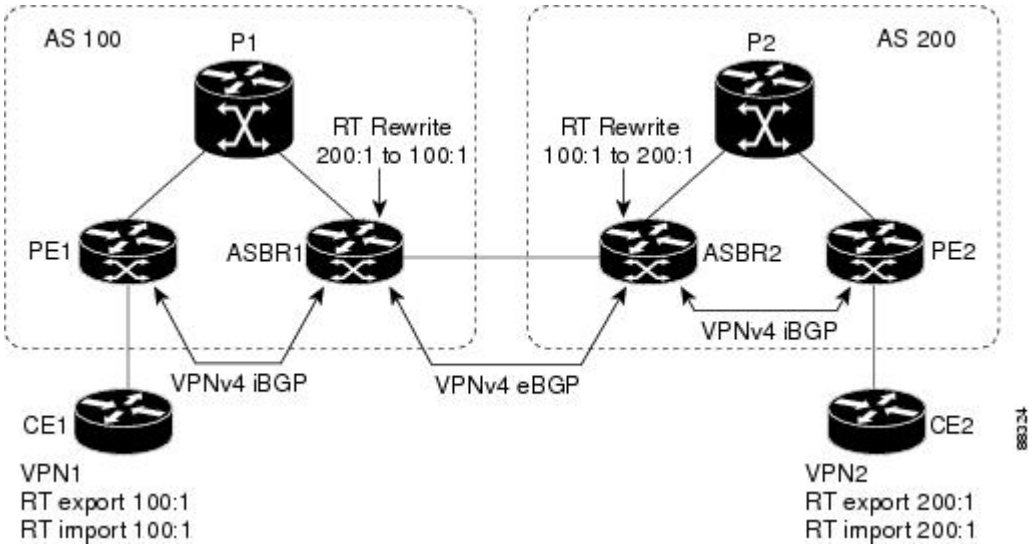
Routing policies for a peer include all configurations that may impact inbound or outbound routing table updates. The MPLS VPN Route Target Rewrite feature can influence routing table updates by allowing the replacement of route targets on inbound and outbound BGP updates. Route targets are carried as extended community attributes in BGP Virtual Private Network IP Version 4 (VPNv4) updates. Route target extended community attributes are used to identify a set of sites and VPN routing and forwarding (VRF) instances that can receive routes with a configured route target.

In general, ASBRs perform route target replacement at autonomous system borders when the ASBRs exchange VPNv4 prefixes. You can also configure the MPLS VPN Route Target Rewrite feature on PE routers and RR routers.

The figure below shows an example of route target replacement on ASBRs in an MPLS VPN interautonomous system topology. This example includes the following configurations:

- PE1 is configured to import and export RT 100:1 for VRF VPN1.
- PE2 is configured to import and export RT 200:1 for VRF VPN2.
- ASBR1 is configured to rewrite all inbound VPNv4 prefixes with RT 200:1 to RT 100:1.
- ASBR2 is configured to rewrite all inbound VPNv4 prefixes with RT 100:1 to RT 200:1.

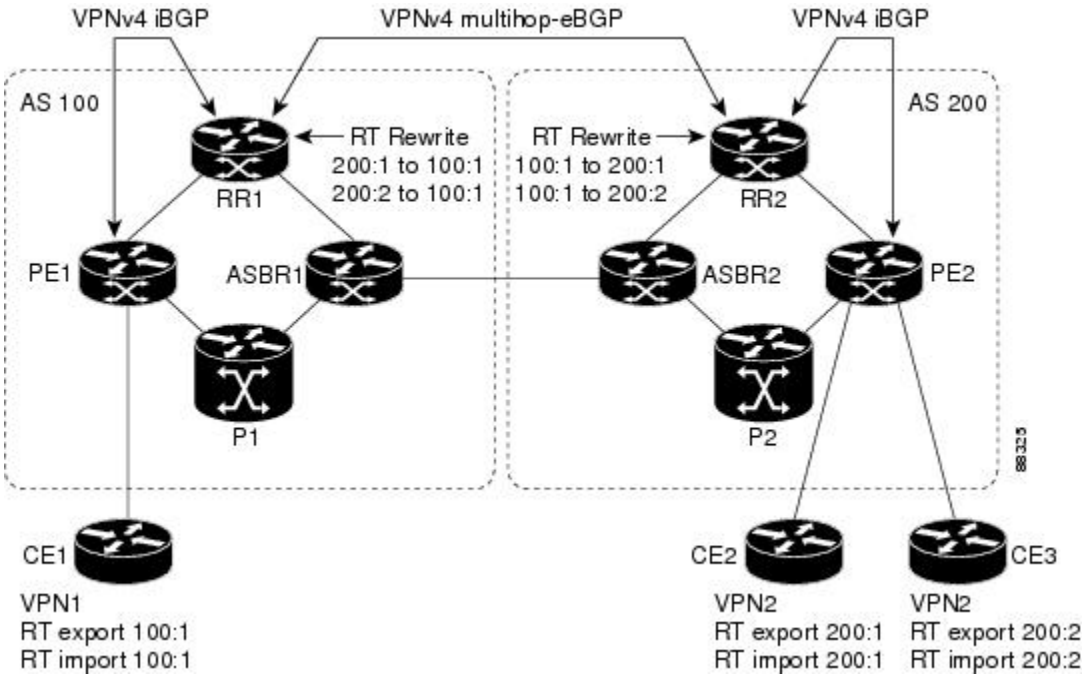
Figure 9 Route Target Replacement on ASBRs in an MPLS VPN Interautonomous System Topology



The figure below shows an example of route target replacement on route reflectors in an MPLS VPN interautonomous system topology. This example includes the following configurations:

- EBGP is configured on the route reflectors.
- EBGP and IBGP IPv4 label exchange is configured between all BGP routers.
- Peer groups are configured on the routers reflectors.
- PE2 is configured to import and export RT 200:1 for VRF VPN2.
- PE2 is configured to import and export RT 200:2 for VRF VPN3.
- PE1 is configured to import and export RT 100:1 for VRF VPN1.
- RR1 is configured to rewrite all inbound VPNv4 prefixes with RT 200:1 or RT 200:2 to RT 100:1.
- RR2 is configured to rewrite all inbound prefixes with RT 100:1 to RT 200:1 and RT 200:2.

Figure 10 Route Target Rewrite on Route Reflectors in an MPLS VPN Interautonomous System Topology



Route Maps and Route Target Replacement

The MPLS VPN--Route Target Rewrite feature extends the BGP inbound/outbound route map functionality to enable route target replacement. The **set extcomm-list delete** command entered in route-map configuration mode allows the deletion of a route target extended community attribute based on an extended community list.

How to Configure MPLS VPN--Route Target Rewrite

- Configuring a Route Target Replacement Policy, page 182
- Applying the Route Target Replacement Policy, page 185
- Verifying the Route Target Replacement Policy, page 189
- Troubleshooting Your Route Target Replacement Policy, page 190

Configuring a Route Target Replacement Policy

Perform this task to configure an RT replacement policy for your internetwork.

If you configure a PE to rewrite RT *x* to RT *y* and the PE has a VRF that imports RT *x* , you need to configure the VRF to import RT *y* in addition to RT *x* .

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip extcommunity-list** {*standard-list-number* | *expanded-list-number*} {**permit** | **deny**} [*regular-expression*] [**rt** | **soo** *extended-community-value*]
4. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
5. **match extcommunity** {*standard-list-number* | *expanded-list-number*}
6. **set extcomm-list** *extended-community-list-number* **delete**
7. **set extcommunity** {**rt** *extended-community-value* [**additive**] | **soo** *extended-community-value*}
8. **end**
9. **show route-map** *map-name*

DETAILED STEPS

	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 extcommunity-list** {*standard-list-number* | *expanded-list-number*} {**permit** | **deny**} [*regular-expression*] [**rt** | **soo** *extended-community-value*] **Example:** `Router(config)# ip extcommunity-list 1 permit rt 100:3`	Creates an extended community access list and controls access to it. • The *standard-list-number* argument is an integer from 1 to 99 that identifies one or more permit or deny groups of extended communities. • The *expanded-list-number* argument is an integer from 100 to 500 that identifies one or more permit or deny groups of extended communities. Regular expressions can be configured with expanded lists but not standard lists. • The **permit** keyword permits access for a matching condition. • The **deny** keyword denies access for a matching condition. • The *regular-expression* argument specifies an input string pattern to match against. When you use an expanded extended community list to match route targets, include the pattern RT: in the regular expression. • The **rt** keyword specifies the route target extended community attribute. The **rt** keyword can be configured only with standard extended community lists and not expanded community lists. • The **soo** keyword specifies the site of origin (SOO) extended community attribute. The **soo** keyword can be configured only with standard extended community lists and not expanded community lists. • The *extended-community-value* argument specifies the route target or site of origin. The value can be one of the following combinations: ◦ autonomous-system-number:network-number ◦ ip-address:network-number The colon is used to separate the autonomous system number and network number or IP address and network number.

Command or Action	Purpose
Step 4 **route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map extmap permit 10`	Defines the conditions for redistributing routes from one routing protocol into another or enables policy routing and enables route-map configuration mode. • The *map-name* argument defines a meaningful name for the route map. The **redistribute** router configuration command uses this name to reference this route map. Multiple route maps may share the same map name. • If the match criteria are met for this route map, and the **permit** keyword is specified, the route is redistributed as controlled by the set actions. In the case of policy routing, the packet is policy routed. If the match criteria are not met, and the **permit** keyword is specified, the next route map with the same map tag is tested. If a route passes none of the match criteria for the set of route maps sharing the same name, it is not redistributed by that set. The **permit** keyword is the default. • If the match criteria are met for the route map and the **deny** keyword is specified, the route is not redistributed. In the case of policy routing, the packet is not policy routed, and no further route maps sharing the same map tag name will be examined. If the packet is not policy routed, the normal forwarding algorithm is used. • The *sequence-number* argument is a number that indicates the position a new route map will have in the list of route maps already configured with the same name. If given with the **no** form of this command, the position of the route map should be deleted.

Step 5 **match extcommunity** {*standard-list-number* | *expanded-list-number*}
 	
Example:
`Router(config-route-map)# match extcommunity 1`
 	
Example:
`Router(config-route-map)# match extcommunity 101`
	Matches BGP extended community list attributes.
	• The *standard-list-number* argument is a number from 1 to 99 that identifies one or more permit or deny groups of extended community attributes.
	• The *expanded-list-number* argument is a number from 100 to 500 that identifies one or more permit or deny groups of extended community attributes.

Step 6 **set extcomm-list** *extended-community-list-number* **delete** **Example:** `Router(config-route-map)# set extcomm-list 1 delete`	Removes a route target from an extended community attribute of an inbound or outbound BGP VPNv4 update. • The *extended-community-list-number* argument specifies the extended community list number.

	Command or Action	Purpose
Step 7	**set extcommunity** {**rt** *extended-community-value* [**additive**] | **soo** *extended-community-value*} **Example:** `Router(config-route-map)# set extcommunity rt 100:4 additive`	Sets BGP extended community attributes. • The **rt** keyword specifies the route target extended community attribute. • The **soo** keyword specifies the site of origin extended community attribute. • The *extended-community-value* argument specifies the value to be set. The value can be one of the following combinations: ◦ autonomous-system-number : network-number ◦ ip-address : network-number The colon is used to separate the autonomous system number and network number or IP address and network number. • The **additive** keyword adds a route target to the existing route target list without replacing any existing route targets.
Step 8	**end** **Example:** `Router(config-route-map)# end`	(Optional) Exits to privileged EXEC mode.
Step 9	**show route-map** *map-name* **Example:** `Router# show route-map extmap`	(Optional) Use this command to verify that the match and set entries are correct. • The *map-name* argument is the name of a specific route map.

Applying the Route Target Replacement Policy

Perform the following tasks to apply the route target replacement policy to your internetwork:

- Associating Route Maps with Specific BGP Neighbors, page 185
- Refreshing BGP Session to Apply Route Target Replacement Policy, page 187
- Troubleshooting Tips, page 188

Associating Route Maps with Specific BGP Neighbors

Perform this task to associate route maps with specific BGP neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
5. **address-family vpnv4** [**unicast**]
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **extended** | **standard**]
8. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
9. **end**

DETAILED STEPS

	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	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and places the router in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 172.10.0.2 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.

	Command or Action	Purpose
Step 5	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes. • The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 172.16.0.2 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **extended** | **standard**] **Example:** `Router(config-router-af)# neighbor 172.16.0.2 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The **both** keyword sends standard and extended community attributes. • The **extended** keyword sends an extended community attribute. • The **standard** keyword sends a standard community attribute.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **Example:** `Router(config-router-af)# neighbor 172.16.0.2 route-map extmap in`	Apply a route map to incoming or outgoing routes • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP or multiprotocol peer group. • The *map-name* argument specifies the name of a route map. • The **in** keyword applies route map to incoming routes. • The **out** keyword applies route map to outgoing routes.
Step 9	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Refreshing BGP Session to Apply Route Target Replacement Policy

Perform this task to refresh the BGP session to apply the RT replacement policy.

After you have defined two routers to be BGP neighbors, the routers form a BGP connection and exchange routing information. If you subsequently change a routing policy, you must reset BGP connections for the configuration change to take effect. After configuring the RT replacement policy and applying it to the target routers in your system, you must refresh the BGP session to put the policy into operation.

SUMMARY STEPS

1. **enable**
2. **clear ip bgp** {***** | *neighbor-address* | *peer-group-name* [**soft** [**in** | **out**]} [**ipv4** {**multicast** | **unicast**} | **vpnv4 unicast** {**soft** | {**in** | **out**}]
3. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip bgp** {*****	*neighbor-address*	*peer-group-name* [**soft** [**in**	**out**]} [**ipv4** {**multicast**	**unicast**}	**vpnv4 unicast** {**soft**	{**in**	**out**}] **Example:** `Router# clear ip bgp vpnv4 unicast 172.16.0.2 in`	Resets a BGP connection using BGP soft reconfiguration. • The ***** keyword resets all current BGP sessions. • The *neighbor-address* argument resets only the identified BGP neighbor. • The *peer-group-name* argument resets the specified BGP peer group. • The **ipv4** keyword resets the specified IPv4 address family neighbor or peer group. The **multicast** or **unicast** keyword must be specified. • The **vpnv4** keyword resets the specified VPNv4 address family neighbor or peer group. The **unicast** keyword must be specified. • The **soft** keyword indicates a soft reset. Does not reset the session. The **in** or **out** keywords do not follow the **soft** keyword when a connection is cleared under the VPNv4 or IPv4 address family because the **soft** keyword specifies both. • The **in** and **out** keywords trigger inbound or outbound soft reconfiguration, respectively. If the **in** or **out** keyword is not specified, both inbound and outbound soft reset are triggered.
Step 3	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Troubleshooting Tips

To determine whether a BGP router supports the route refresh capability, use the **show ip bgp neighbors** command. If a router supports the route refresh capability, the following message is displayed:

```
Received route refresh capability from peer.
```

You can issue the **debug ip bgp updates** command on the router where you entered the **clear ip bgp** command to verify that the updates are occurring.

Note Issuing the **debug ip bgp updates** command could impair performance if the router sends or receives a large number of BGP updates.

Verifying the Route Target Replacement Policy

Perform this task to verify the operation of your RT replacement policy.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4 all** *network-address*
3. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 **show ip bgp vpnv4 all** *network-address*

Use this command to verify that all VPNv4 prefixes with a specified RT extended community attribute are replaced with the proper RT extended community attribute at the ASBRs or route reflectors and to verify that the PE routers receive the rewritten RT extended community attributes from the ASBRs or route reflectors. The following examples verify route target replacement on ABSR1 and ABSR2.

Verify route target replacement on ABSR1:

Example:

```
Router# show ip bgp vpnv4 all 172.16.17.17
BGP routing table entry for 100:1:172.16.17.17/32, version 6
Paths: (1 available, best #1, no table)
  Advertised to update-groups:
     1
  300
    172.16.11.11 (metric 589) from 172.16.11.11 (172.16.11.11)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:200:1
```

Verify route target replacement on ABSR2:

Example:

```
Router# show ip bgp vpnv4 all 172.16.17.17
BGP routing table entry for 100:1:172.16.17.17/32, version 6
Paths: (1 available, best #1, no table)
  Advertised to update-groups:
```

```
    1
  100 300
    192.168.1.1 from 192.168.1.1 (172.16.13.13)
      Origin incomplete, localpref 100, valid, external, best
      Extended Community: RT:100:1
```

The following examples verify route target replacement on PE1 and PE2.

Verify route target on PE1:

Example:

```
Router# show ip bgp vpnv4 all 172.16.17.17
BGP routing table entry for 100:1:172.16.17.17/32, version 13
Paths: (1 available, best #1, table vpn1)
  Advertised to update-groups:
     1
  300
    192.168.2.1 (via vpn1) from 192.168.2.1 (172.16.19.19)
      Origin incomplete, metric 0, localpref 100, valid, external, best
      Extended Community: RT:200:1
```

Verify route target on PE2:

Example:

```
Router# show ip bgp vpnv4 all 172.16.17.17
BGP routing table entry for 100:1:172.16.17.17/32, version 13
Paths: (1 available, best #1, table vpn1)
  Advertised to update-groups:
     3
  100 300
    192.168.1.1 (metric 20) from 172.16.16.16 (172.16.16.16)
      Origin incomplete, localpref 100, valid, internal, best
      Extended Community: RT:100:1
```

Step 3 **exit**

Use this command to exit to user EXEC mode:

Example:

```
Router# exit
Router>
```

Troubleshooting Your Route Target Replacement Policy

Perform this task to troubleshoot your RT replacement policy.

SUMMARY STEPS

1. **enable**
2. **debug ip bgp updates**
3. **show ip bgp vpnv4 all** *network-address*
4. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 **debug ip bgp updates**

Use the following command to verify that BGP updates are occurring on the ASBR. The ASBR in this example has the IP address 172.16.16.16.

Example:

```
Router# debug ip bgp updates
BGP(2): no valid path for 100:1:172.16.20.20/32
BGP(2): no valid path for 100:1:10.0.0.0/8
%BGP-5-ADJCHANGE: neighbor 172.16.16.16 Down User reset
BGP(2): nettable_walker 100:1:172.16.20.20/32 no RIB
BGP(2): nettable_walker 100:1:192.168.3.0/8 no RIB
BGP(2): 172.16.11.11 computing updates, afi 2, neighbor version 13,
table version 15, starting at 0.0.0.0
BGP(2): 172.16.11.11 send unreachable 100:1:172.16.20.20/32
BGP(2): 172.16.11.11 send UPDATE 100:1:172.16.20.20/32 -- unreachable
BGP(2): 172.16.11.11 send UPDATE 100:1:192.168.3.0/8 -- unreachable
BGP(2): 1 updates (average = 58, maximum = 58)
BGP(2): 172.16.11.11 updates replicated for neighbors: 172.16.11.11
BGP(2): 172.16.11.11 update run completed, afi 2, ran for 0ms,
neighbor version 15, start version 15, throttled to 15
BGP: Import walker start version 13, end version 15
BGP: ... start import cfg version = 30
%BGP-5-ADJCHANGE: neighbor 172.16.16.16 Up
BGP(2): 172.16.16.16 computing updates, afi 2, neighbor version 0,
table version 15, starting at 0.0.0.0
BGP(2): 172.16.16.16 send UPDATE (format) 100:1:172.16.0.0/16,
next 172.16.11.11, metric 0, path 300, extended community RT:2:2
RT:7777:222222222 RT:20000:111 RT:65535:999999999
BGP(2): 172.16.16.16 send UPDATE (prepend, chgflags: 0x0)
100:1:172.16.19.19/32, next 172.16.11.11, metric 0, path 300,
extended community RT:2:2 RT:7777:222222222 RT:20000:111
RT:65535:999999999
BGP(2): 172.16.16.16 send UPDATE (format) 100:1:192.168.2.0/8,
next 172.16.11.11, metric 0, path , extended community
RT:2:2 RT:7777:222222222 RT:20000:111 RT:65535:999999999
BGP(2): 2 updates (average = 111, maximum = 121)
BGP(2): 172.16.16.16 updates replicated for neighbors: 172.16.16.16
BGP(2): 172.16.16.16 update run completed, afi 2, ran for 0ms,
neighbor version 15, start version 15, throttled to 15
BGP(2): 172.16.16.16 rcvd UPDATE w/ attr: nexthop 172.16.15.15,
origin ?, path 200, extended community RT:100:1
BGP(2): 172.16.16.16 rcvd 100:1:192.168.3.0/8
BGP(2): 172.16.16.16 rcvd UPDATE w/ attr: nexthop 172.16.15.15,
origin ?, path 200 400, extended community RT:100:1
BGP(2): 172.16.16.16 rcvd 100:1:172.16.0.0/16
BGP(2): 172.16.16.16 rcvd 100:1:172.16.20.20/32
BGP(2): nettable_walker 100:1:172.16.20.20/32 no RIB
BGP(2): nettable_walker 100:1:192.168.3.0/8 no RIB
BGP: Import walker start version 15, end version 17
BGP: ... start import cfg version = 30
BGP(2): 172.16.11.11 computing updates, afi 2,
neighbor version 15, table version 17,
starting at 0.0.0.0
```

```
BGP(2): 172.16.11.11 NEXT_HOP part 1 net 100:1:172.16.20.20/32,
next 172.16.15.15
BGP(2): 172.16.11.11 send UPDATE (format) 100:1:172.16.20.20/32,
next 172.16.15.15,metric 0, path 200 400, extended community
RT:1:1 RT:10000:111 RT:33333:888888888
RT:65535:999999999
BGP(2): 172.16.11.11 NEXT_HOP part 1 net 100:1:10.0.0.0/8,
next 172.16.15.15
BGP(2): 172.16.11.11 send UPDATE (format) 100:1:192.168.3.0/8,
next 172.16.15.15, metric 0, path 200, extended community
RT:1:1 RT:10000:111 RT:33333:888888888 RT:65535:999999999
BGP(2): 2 updates (average = 118, maximum = 121)
BGP(2): 172.16.11.11 updates replicated for neighbors: 172.16.11.11
BGP(2): 172.16.11.11 update run completed, afi 2, ran for 0ms,
neighbor version 17, start version 17, throttled to 17
```

You can also reset the BGP connection using the **clear ip bgp *** command and enter the **debug ip bgp updates** command again to verify that BGP updates are occurring as shown in the output after the **clear ip bgp** command is entered.

Step 3 **show ip bgp vpnv4 all** *network-address*

Use this command to verify that RT extended community attributes are replaced correctly. For example:

Example:

```
Router# show ip bgp vpnv4 all 172.16.17.17
BGP routing table entry for 100:1:172.16.17.17/32, version 6
Paths: (1 available, best #1, no table)
  Advertised to update-groups:
     1
  100 300
    192.168.1.1 from 192.168.1.1 (172.16.13.13)
      Origin incomplete, localpref 100, valid, external, best
      Extended Community: RT:100:1
```

This example shows VPN address information from the BGP table and verifies that RT extended community attributes are replaced correctly.

Step 4 **exit**

Use this command to exit to user EXEC mode:

Example:

```
Router# exit
Router>
```

Configuration Examples for MPLS VPN--Route Target Rewrite

- Configuring Route Target Replacement Policies Examples, page 192
- Applying Route Target Replacement Policies Examples, page 193

Configuring Route Target Replacement Policies Examples

This example shows the RT replacement configuration of an ASBR (ASBR1) that exchanges VPNv4 prefixes with another ASBR (ASBR2). The route map extmap is configured to replace RTs on inbound

updates. Any incoming update with RT 100:3 is replaced with RT 200:3. Any other prefixes with an RT whose autonomous system number is 100 is rewritten to RT 200:4.

```
!
ip extcommunity-list 1 permit rt 100:3
ip extcommunity-list 101 permit RT:100:*
!
route-map extmap permit 10
match extcommunity 1
set extcomm-list 1 delete
set extcommunity rt 200:3 additive
!
route-map regexp permit 10
match extcommunity 101
set extcomm-list 101 delete
set extcommunity rt 200:4 additive
!
route-map regexp permit 20
```

This example shows the use of the route-map configuration **continue** command when you need to apply more than one replacement rule on an update. In this example, an incoming update with RT 100:3 is replaced with RT 200:3. Without the **continue 20** command, route-map evaluation would stop when a match on sequence 10 is made. With the **continue 20** command, route-map evaluation continues into sequence 20 even if a match occurs in sequence 10. If the incoming update has an RT 100:4, the router replaces it with RT 200:4.

```
!
ip extcommunity-list 1 permit rt 100:3
ip extcommunity-list 2 permit rt 100:4
!
route-map extmap permit 10
match extcommunity 1
set extcomm-list 1 delete
set extcommunity rt 200:3 additive
continue 20
!
route-map extmap permit 20
match extcommunity 2
set extcomm-list 2 delete
set extcommunity rt 200:4 additive
!
route-map extmap permit 30
```

Note The route-map configuration **continue** command is not supported on outbound route maps.

Applying Route Target Replacement Policies Examples

This section contains the following examples:

- Associating Route Maps with Specific BGP Neighbor Example, page 193
- Refreshing the BGP Session to Apply the Route Target Replacement Policy Example, page 194

Associating Route Maps with Specific BGP Neighbor Example

This example shows the association of route map extmap with a BGP neighbor. The BGP inbound route map is configured to replace RTs on incoming updates.

```
router bgp 100
.
```

```
.
.
 neighbor 172.16.0.2 remote-as 100
.
.
.
!
address family vpnv4
 neighbor 172.16.0.2 activate
 neighbor 172.16.0.2 send-community extended
 neighbor 172.16.0.2 route-map extmap in
```

This example shows the association of the same route map with the outbound BGP neighbor. The route map is configured to replace RTs on outgoing updates.

```
router bgp 100
.
.
.
 neighbor 172.16.0.2 remote-as 100
.
.
.
!
address family vpnv4
 neighbor 172.16.0.2 activate
 neighbor 172.16.0.2 send-community extended
 neighbor 172.16.0.2 route-map extmap out
```

Refreshing the BGP Session to Apply the Route Target Replacement Policy Example

The following example shows the **clear ip bgp** command used to initiate a dynamic reconfiguration in the BGP peer 172.16.0.2. This command requires that the peer supports the route refresh capability.

```
Router# clear ip bgp 172.16.0.2 vpnv4 unicast in
```

Additional References

Related Documents

Related Topic	Document Title
MPLS, MPLS VPN, and MPLS VPN interautonomous systems configuration tasks	*MPLS Layer 3 Inter-AS and CSC Configuration Guide*
Commands to configure MPLS and MPLS VPNs	*Cisco IOS Multiprotocol Label Switching Command Reference*
BGP configuration tasks	*IP Routing Protocols Configuration Guide*
Commands to configure and monitor BGP	*Cisco IOS IP Routing Protocols Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	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 software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	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 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 MPLS VPN--Route Target Rewrite

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 12 Feature Information for MPLS VPN--Route Target Rewrite

Feature Name	Releases	Feature Information
MPLS VPN--Route Target Rewrite	12.0(26)S 12.2(25)S 12.2(33)SRA 12.2(33)SXH 12.4(20)T	The MPLS VPN--Route Target Rewrite feature allows the replacement of route targets on incoming and outgoing Border Gateway Protocol (BGP) updates. Typically, Autonomous System Border Routers (ASBRs) perform the replacement of route targets at autonomous system boundaries. Route Reflectors (RRs) and provider edge (PE) routers can also perform route target replacement. The main advantage of the MPLS VPN--Route Target Rewrite feature is that it keeps the administration of routing policy local to the autonomous system. In 12.0(26)S, this feature was introduced for the Cisco 7200, 7500, and 12000 series routers. In 12.2(25)S, this feature was integrated into a Cisco IOS 12.2S release to support the Cisco 7500 series router. In 12.2(33)SRA, this feature was integrated into a Cisco IOS 12.2SRA release. In 12.2(33)SXH, this feature was integrated into a Cisco IOS 12.2SXH release. In 12.4(20)T, this feature was integrated into a Cisco IOS 12.4T release. The following command was modified: **set extcomm-list delete**.

Glossary

autonomous system --A collection of networks that share the same routing protocol and that are under the same system administration.

ASBR --autonomous system border router. A router that connects and exchanges information between two or more autonomous systems.

BGP --Border Gateway Protocol. The exterior border gateway protocol used to exchange routing information between routers in separate autonomous systems. BGP uses Transmission Control Protocol (TCP). Because TCP is a reliable protocol, BGP does not experience problems with dropped or fragmented data packets.

CE router --customer edge router. The customer router that connects to the provider edge (PE) router.

EBGP --External Border Gateway Protocol.A BGP session between routers in different autonomous systems. When a pair of routers in different autonomous systems are more than one IP hop away from each other, an EBGP session between those two routers is called multihop EBGP.

IBGP --Internal Border Gateway Protocol.A BGP session between routers within the same autonomous system.

IGP --Interior Gateway Protocol. Internet protocol used to exchange routing information within an autonomous system. Examples of common Internet IGPs include Internal Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).

LDP --Label Distribution Protocol. A standard protocol between MPLS-enabled routers to negotiate the labels (addresses) used to forward packets. The Cisco proprietary version of this protocol is the Tag Distribution Protocol (TDP).

LER --label edge router. The edge router that performs label imposition and disposition.

LSR --label switch router. The role of an LSR is to forward packets in an MPLS network by looking only at the fixed-length label.

MPLS --Multiprotocol Label Switching. A switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

NLRI --Network Layer Reachability Information.BGP sends routing update messages containing NLRI, which describes the route. In this context, an NLRI is a prefix. A BGP update message carries one or more NLRI prefixes and the attributes of a route for the NLRI prefixes. The route attributes include a BGP next-hop gateway address, community values, and other information.

P router --provider router. The core router in the service provider network that connects to provider edge (PE) routers. In a packet-switched star topology, a router that is part of the backbone and that serves as the single pipe through which all traffic from peripheral networks must pass on its way to other peripheral networks.

PE router --provider edge router.The label edge router (LER) in the service provider network that connects to the customer edge (CE) router.

RD --route distinguisher. An 8-byte value that is concatenated with an IPv4 prefix to create a unique VPN IPv4 (VPNv4) prefix.

RR --route reflector. A router that advertises, or reflects, IBGP learned routes to other IBGP peers without requiring a full network mesh.

RT --route target. Extended community attribute used to identify the VRF routing table into which a prefix is to be imported.

VPN --Virtual Private Network.A group of sites that, as a result of a set of administrative policies, can communicate with each other over a shared backbone.

VPNv4 prefix --IPv4 prefix preceded by an 8-byte route distinguisher. The VPN addresses are made unique by adding a route distinguisher to the front of the address.

VRF --VPN routing and forwarding instance. A VRF 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 what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a provider edge (PE) router.

MPLS VPN VRF Selection Using Policy-Based Routing

The MPLS VPN: VRF Selection Using Policy-Based Routing feature is an extension of the MPLS VPN: VRF Selection Based on Source IP Address feature. This feature introduces a policy-based routing (PBR) mechanism to classify and forward Virtual Private Network (VPN) traffic based on multiple VPN routing and forwarding (VRF) selection match criteria.

- Finding Feature Information, page 199
- Prerequisites for VRF Selection Using Policy-Based Routing, page 199
- Restrictions for VRF Selection Using Policy-Based Routing, page 200
- Information About VRF Selection Using Policy-Based Routing, page 200
- How to Configure VRF Selection Using Policy-Based Routing, page 201
- Configuration Examples for VRF Selection Using Policy-Based Routing, page 209
- Additional References, page 210
- Feature Information for VRF Selection Using Policy-Based Routing, page 212
- Glossary, page 212

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for VRF Selection Using Policy-Based Routing

The router must support PBR to configure this feature. For platforms that do not support PBR, use the "VRF Selection Based on Source IP Address" feature introduced in Cisco IOS Release 12.0(22)S.

A VRF must be defined prior to the configuration of this feature. An error message is displayed on the console if no VRF exists.

This document assumes that multiprotocol BGP (mBGP), Multiprotocol Label Switching (MPLS), and Cisco Express Forwarding are enabled in your network.

Restrictions for VRF Selection Using Policy-Based Routing

The VRF Selection Using Policy-Based Routing feature is supported only in service provider (-p-) images.

The VRF Selection Using Policy-Based Routing feature can coexist with the VRF Selection Based on Source IP address feature on the same router, but these features cannot be configured together on the same interface. This is designed behavior to prevent VRF table selection conflicts that could occur if these features were misconfigured together. An error message is displayed on the console if you attempt to configure the ip vrf select source and the ip policy route-map commands on the same interface.

Protocol Independent Multicast (PIM) and multicast packets do not support PBR and cannot be configured for a source IP address that is a match criterion for this feature.

The VRF Selection Using Policy-Based Routing feature cannot be configured with IP prefix lists.

Information About VRF Selection Using Policy-Based Routing

- Introduction to VRF Selection Using Policy-Based Routing, page 200
- Policy-Based Routing Set Clauses Overview, page 200

Introduction to VRF Selection Using Policy-Based Routing

The VRF Selection Using Policy-Based Routing feature is an extension of the VRF Selection Based on Source IP Address feature. The PBR implementation of the VRF selection feature allows you to policy route VPN traffic based on match criteria. Match criteria are defined in an IP access list or based on packet length. The following match criteria are supported in Cisco software:

- IP access lists--Define match criteria based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. All IP access-list configuration options in Cisco software can be used to define match criteria.
- Packet lengths--Define match criteria based on the length of a packet in bytes. The packet length filter is defined in a route map with the **match length** route-map configuration command.

Policy routing is defined in the route map. The route map is applied to the incoming interface with the **ip policy route-map** interface configuration command. An IP access list is applied to the route map with the **match ip address** route-map configuration command. Packet length match criteria are applied to the route map with the **match length** route-map configuration command. The set action is defined with the **set vrf** route-map configuration command. The match criteria are evaluated, and the appropriate VRF is selected by the set clause. This combination allows you to define match criteria for incoming VPN traffic and policy route VPN packets out to the appropriate VRF.

Policy-Based Routing Set Clauses Overview

When you are configuring PBR, the following four set clauses can be used to change normal routing and forwarding behavior:

- set default interface
- set interface
- set ip default next-hop
- set ip next-hop

Configuring any of the set clauses will overwrite normal routing forwarding behavior of a packet.

The VRF Selection Using Policy-Based Routing feature introduces the fifth set clause that can be used to change normal routing and forwarding behavior. The set vrf command is used to select the appropriate VRF after the successful match occurs in the route map.

How to Configure VRF Selection Using Policy-Based Routing

- Defining the Match Criteria for PBR VRF Selection Based on Packet Length, page 201
- Configuring PBR VRF Selection in a Route Map, page 203
- Configuring PBR on the Interface, page 205
- Configuring IP VRF Receive on the Interface, page 206
- Verifying the Configuration of the VRF Selection Using Policy-Based Routing, page 208

Defining the Match Criteria for PBR VRF Selection Based on Packet Length

The match criteria for PBR VRF route selection are defined in an access list. Standard and named access lists are supported. Match criteria can also be defined based on the packet length using the **match length** route-map configuration command. This configuration option is defined entirely within a route map.

- Prerequisites, page 201
- Configuring PBR VRF Selection with a Standard Access List, page 201
- Configuring PBR VRF Selection with a Named Access List, page 202

Prerequisites

Before you perform this task, make sure that the VRF and associated IP address are already defined.

Configuring PBR VRF Selection with a Standard Access List

Use the following commands to create a standard access list and define the PBR VRF route selection match criteria in it in order to permit or deny the transmission of VPN traffic data packets.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} *source-addr* [*source-wildcard*] [**log**]

DETAILED STEPS

	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	**access-list** *access-list-number* {**deny** | **permit**} *source-addr* [*source-wildcard*] [**log**] **Example:** `Router(config)# access-list 40 permit 10.1.0.0/24 0.0.0.255`	Creates an access list and defines the match criteria for the route map. • Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access-list filtering options. Named, numbered, standard, and extended access lists are supported. All IP access list configuration options in Cisco software can be used to define match criteria. • The example creates a standard access list numbered 40. This filter will permit traffic from any host with an IP address in the 10.1.0.0/24 subnet.

Configuring PBR VRF Selection with a Named Access List

Use the following commands to define the PBR VRF route selection match criteria in a named access list in order to permit or deny the transmission of VPN traffic data packets.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*]
4. [*sequence-number*] {**permit** | **deny**} *protocol source-addr source-wildcard destination-addr destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

DETAILED STEPS

	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** {**standard** | **extended**} [*access-list-name* | *access-list-number*] **Example:** `Router(config)# ip access-list extended NAMEDACL`	Specifies the IP access list type and enters the corresponding access-list configuration mode. • A standard, extended, or named access list can be used.
Step 4	[*sequence-number*] {**permit** | **deny**} *protocol source-addr source-wildcard destination-addr destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-ext-nacl)# permit ip any any option any-options`	Defines the criteria for which the access list will permit or deny packets. • Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access-list filtering options. Named, numbered, standard, and extended access lists are supported. All IP access-list configuration options in Cisco software can be used to define match criteria. • The example creates a named access list that permits any configured IP option.

Configuring PBR VRF Selection in a Route Map

Use the following commands to configure the VRF through which the outbound VPN packets will be policy routed in order to permit or deny the transmission of VPN traffic data packets.

Incoming packets are filtered through the match criteria that are defined in the route map. After a successful match occurs, the **set vrf** command configuration determines the VRF through which the outbound VPN packets will be policy routed.

- The VRF must be defined prior to the configuration of the route map; otherwise an error message is displayed on the console.
- A receive entry must be added to the VRF selection table with the **ip vrf receive** command. If a match and set operation occurs in the route map but there is no receive entry in the local VRF table, the packet will be dropped if the packet destination is local.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. Do one of the following:
   - **match ip address** {*acl-number* [*acl-number ...* | *acl-name ...*] | *acl-name* [*acl-name ...* | *acl-number ...*]}
   -
   - **match length** *minimum-length maximum-length*
5. **set vrf** *vrf-name*
6. **exit**

DETAILED STEPS

	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	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map map1 permit 10`	Enters route map configuration mode. Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing.

Step 4	Do one of the following:
	• **match ip address** {*acl-number* [*acl-number ...* | *acl-name ...*] | *acl-name* [*acl-name ...* | *acl-number ...*]}
	•
	• **match length** *minimum-length maximum-length*
	
	Example:
	`Router(config-route-map)# match ip address 1`
	
	Example:
	`Router(config-route-map)# match length 3 200`
		Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets.
		• IP access lists are supported.
		• The example configures the route map to use standard access list 1 to define match criteria.
		or
		Specifies the Layer 3 packet length in the IP header as a match criterion in a class map.
		• The example configures the route map to match packets that are 3 to 200 bytes in size.

Step 5	**set vrf** *vrf-name* **Example:** `Router(config-route-map)# set vrf map1`	Defines which VRF to route VPN packets that are successfully matched in the same route map sequence for PBR VRF selection. • The example policy routes matched packets out to the VRF named map1.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.

Configuring PBR on the Interface

Use the following commands to filter incoming VPN traffic data packets. Incoming packets are filtered through the match criteria that are defined in the route map.

The route map is applied to the incoming interface. The route map is attached to the incoming interface with the ip policy route-map global configuration command.

Note

- The VRF Selection Using Policy-Based Routing feature can coexist with the VRF Selection Based on Source IP address feature on the same router, but the two features cannot be configured together on the same interface. This is designed behavior to prevent VRF table selection conflicts that could occur if these features were misconfigured together. An error message is displayed on the console if you attempt to configure the ip vrf select source and the ip policy route-map commands on the same interface.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip policy route-map** *map-tag*
5. **ip vrf receive** *vrf-name*
6. **exit**

DETAILED STEPS

	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** *type number* [*name-tag*] **Example:** `Router(config)# interface FastEthernet 0/1`	Configures an interface and enters interface configuration mode.
Step 4	**ip policy route-map** *map-tag* **Example:** `Router(config-if)# ip policy route-map map1`	Identifies a route map to use for policy routing on an interface. • The configuration example attaches the route map named map1 to the interface.
Step 5	**ip vrf receive** *vrf-name* **Example:** `Router(config-if)# ip vrf receive VRF1`	Adds the IP addresses that are associated with an interface into the VRF table. • This command must be configured for each VRF that will be used for VRF selection.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.

Configuring IP VRF Receive on the Interface

Use the following commands to insert the IP address of an interface as a connected route entry in a VRF routing table. This will prevent dropped packets.

The source IP address must be added to the VRF selection table. VRF selection is a one-way (unidirectional) feature. It is applied to the incoming interface. If a match and set operation occurs in the route map but there is no VRF receive entry in the local VRF table, the packet will be dropped if the packet destination is local.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip policy route-map** *map-tag*
5. **ip vrf receive** *vrf-name*
6. **end**

DETAILED STEPS

	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* [*name-tag*] **Example:** `Router(config)# interface FastEthernet 0/1`	Configures an interface and enters interface configuration mode.
Step 4	**ip policy route-map** *map-tag* **Example:** `Router(config-if)# ip policy route-map map1`	Identifies a route map to use for policy routing on an interface. • The configuration example attaches the route map named map1 to the interface.
Step 5	**ip vrf receive** *vrf-name* **Example:** `Router(config-if)# ip vrf receive VRF1`	Adds the IP addresses that are associated with an interface into the VRF table. • This command must be configured for each VRF that will be used for VRF selection.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode, and enters privileged EXEC mode.

Verifying the Configuration of the VRF Selection Using Policy-Based Routing

To verify the configuration of the VRF Selection Using Policy-Based Routing feature, perform each of the following steps in this section in the order specified.

SUMMARY STEPS

1. **enable**
2. **show ip access-list** [*access-list-number* | *access-list-name*]
3. **show route-map** [*map-name*]
4. **show ip policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip access-list** [*access-list-number*	*access-list-name*] **Example:** `Router# show ip access-list`	Displays the contents of all current IP access lists. • This command is used to verify the match criteria that are defined in the access list. Both named and numbered access lists are supported.
Step 3	**show route-map** [*map-name*] **Example:** `Router# show route-map`	Displays all route maps configured or only the one specified. • This command is used to verify match and set clauses within the route map.

	Command or Action	Purpose
Step 4	**show ip policy** **Example:** `Router# show ip policy`	Displays the route map used for policy routing. • This command can be used to display the route map and the associated interface.

Configuration Examples for VRF Selection Using Policy-Based Routing

- Example Defining PBR VRF Selection in Access List, page 209
- Example Verifying VRF Selection Using Policy-Based Routing, page 209

Example Defining PBR VRF Selection in Access List

In the following example, three standard access lists are created to define match criteria for three different subnets. Any packets received on the FastEthernet 0/1/0 interface will be policy routed through the PBR-VRF-Selection route map to the VRF that is matched in the same route map sequence. If the source IP address of the packet is part of the 10.1.0.0/24 subnet, VRF1 will be used for routing and forwarding.

```
access-list 40 permit 10.1.0.0 0.0.255.255
access-list 50 permit 10.2.0.0 0.0.255.255
access-list 60 permit 10.3.0.0 0.0.255.255
route-map PBR-VRF-Selection permit 10
 match ip address 40
 set vrf VRF1
 !
route-map PBR-VRF-Selection permit 20
 match ip address 50
 set vrf VRF2
 !
route-map PBR-VRF-Selection permit 30
 match ip address 60
 set vrf VRF3
 !
interface FastEthernet0/1/0
 ip address 10.1.0.0/24 255.255.255.252
 ip policy route-map PBR-VRF-Selection
 ip vrf receive VRF1
 ip vrf receive VRF2
 ip vrf receive VRF3
```

Example Verifying VRF Selection Using Policy-Based Routing

The following verification examples show defined match criteria and route-map policy configuration.

- Verifying Match Criteria, page 210
- Verifying Route-Map Configuration, page 210
- Verifying PBR VRF Selection Policy, page 210

Verifying Match Criteria

To verify the configuration of match criteria for PBR VRF selection, use the **show ip access-list** command.

The following **show ip access-list** command output displays three subnet ranges defined as match criteria in three standard access lists:

```
Router# show ip access-list
Standard IP access list 40
    10 permit 10.1.0.0, wildcard bits 0.0.255.255
Standard IP access list 50
    10 permit 10.2.0.0, wildcard bits 0.0.255.255
Standard IP access list 60
    10 permit 10.3.0.0, wildcard bits 0.0.255.255
```

Verifying Route-Map Configuration

To verify route-map configuration, use the **show route-map** command. The output displays the match criteria and set action for each route-map sequence. The output also displays the number of packets and bytes that have been policy routed per each route-map sequence.

```
Router# show route-map
route-map PBR-VRF-Selection, permit, sequence 10
  Match clauses:
    ip address (access-lists): 40
  Set clauses:
    vrf VRF1
  Policy routing matches: 0 packets, 0 bytes
route-map PBR-VRF-Selection, permit, sequence 20
  Match clauses:
    ip address (access-lists): 50
  Set clauses:
    vrf VRF2
  Policy routing matches: 0 packets, 0 bytes
route-map PBR-VRF-Selection, permit, sequence 30
  Match clauses:
    ip address (access-lists): 60
  Set clauses:
    vrf VRF3
  Policy routing matches: 0 packets, 0 bytes
```

Verifying PBR VRF Selection Policy

The following **show ip policy** command output displays the interface and associated route map that is configured for policy routing:

```
Router# show ip policy
Interface               Route map
FastEthernet0/1/0       PBR-VRF-Selection
```

Additional References

Related Documents

Related Topic	Document Title
VRF selection based on the source IP address	Directing MPLS VPN Traffic Using a Source IP Address

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 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 standards has not been modified by this 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 VRF Selection Using Policy-Based Routing

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 13 Feature Information for VRF Selection Using Policy-Based Routing

Feature Name	Releases	Feature Information
MPLS VPN: VRF Selection Using Policy-Based Routing	12.3(7)T 12.2(25)S 12.2(33)SRB 12.2(33)SXI	The MPLS VPN: VRF Selection Using Policy-Based Routing feature is an extension of the MPLS VPN: VRF Selection Based on Source IP Address feature. This feature introduces a policy-based routing (PBR) mechanism to classify and forward Virtual Private Network (VPN) traffic based on multiple VPN routing and forwarding (VRF) selection match criteria. In 12.3(7)T, this feature was introduced. This feature was integrated into Cisco IOS Release 12.2(25)S. This feature was integrated into Cisco IOS Release 12.2(33)SRB. This feature was integrated into Cisco IOS Release 12.2(33)SXI. The following commands were introduced or modified: **ip vrf receive**, **set vrf**.

Glossary

PBR --policy-based routing.

VPN --Virtual Private Network.

VRF --virtual routing and forwarding.

MPLS VPN - Interautonomous System Support

An autonomous system is a single network or group of networks that is controlled by a common system administration group and that uses a single, clearly defined routing protocol. The MPLS VPN - Interautonomous System Support feature allows an Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) to span service providers and autonomous systems.

This document explains how to enable Autonomous System Boundary Routers (ASBRs) to use exterior Border Gateway Protocol (eBGP) to exchange IPv4 Network Layer Reachability Information (NLRI) in the form of VPN-IPv4 addresses.

As VPNs grow, their requirements expand. In some cases, VPNs need to reside on different autonomous systems in different geographic areas. Also, some VPNs need to extend across multiple service providers (overlapping VPNs). Regardless of the complexity and location of the VPNs, the connection between autonomous systems must be seamless to the customer. The MPLS VPN - Interautonomous System Support feature provides this functionality.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the Feature Information for MPLS VPN - Interautonomous System Support, page 271.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

- Finding Feature Information, page 216
- Prerequisites for MPLS VPN - Interautonomous System Support, page 216
- Restrictions for MPLS VPN - Interautonomous System Support, page 217
- Information About MPLS VPN - Interautonomous System Support, page 217
- How to Configure MPLS VPN - Interautonomous System Support, page 227
- Configuration Examples for MPLS VPN - Interautonomous System Support, page 250
- Additional References, page 270
- Feature Information for MPLS VPN - Interautonomous System Support, page 271
- Glossary, page 274

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for MPLS VPN - Interautonomous System Support

Before you configure eBGP routing between autonomous systems or subautonomous systems in an MPLS VPN, ensure that you have properly configured all MPLS VPN routing instances and sessions. The configuration tasks outlined in the How to Configure MPLS VPN - Interautonomous System Support, page 227 build from those configuration tasks.

Perform (as appropriate to the existing network configuration) the following tasks as described in the the Configuring MPLS VPNs feature module.

- Define VPN routing instances
- Configure BGP routing sessions in the service provider (P) network
- Configure provider edge (PE) to PE routing sessions in the service provider (P) network
- Configure BGP PE to customer edge (CE) routing sessions

A VPN-IPv4 eBGP session must be configured between directly connected ASBRs.

This feature is supported on the Cisco IOS 12000 series line cards listed in the table below.

Table 14 Cisco I2000 Series Line Card Support Added for Cisco IOS Releases

Type	Line Cards	Cisco IOS Release Added
Packet over SONET (POS)	4-Port OC-3 POS	12.0(16)ST
	1-Port OC-12 POS	12.0(17)ST
	8-Port OC-3 POS	12.0(22)S
	16-Port OC-3 POS	
	4-Port OC-12 POS	
	1-Port OC-48 POS	
	4-Port OC-3 POS ISE	
	8-Port OC-3 POS ISE	
	16-Port OC-3 POS ISE	
	4-Port OC-12 POS ISE	
	1-Port OC-48 POS ISE	

Type	Line Cards	Cisco IOS Release Added
Electrical Interface	6-Port DS3	12.0(21ST
	12-Port DS3	12.0(22)S
	6-Port E3	
	12-Port E3	
Ethernet	3-Port GbE	12.0(23)S
	1-Port 10-GbE Modular GbE/FE	12.0(24)S
ATM	4-Port OC-3 ATM	12.0(16)ST
	1-Port OC12 ATM	12.0(17)ST
	4-Port OC-12 ATM	12.0(23)S
	8-Port OC-3 ATM	
Channelized Interface	2-Port CHOC-3	12.0(22)S
	6-Port Ch T3 (DS1)	
	1-Port CHOC-12 (DS3)	
	1-Port CHOC-12 (OC-3)	
	4-Port CHOC-12 ISE	
	1-Port CHOC-48 ISE	

Restrictions for MPLS VPN - Interautonomous System Support

Note the following restrictions to the MPLS VPN - Interautonomous System Support feature:

- A VPN-IPv4 eBGP session must be configured between directly connected ASBRs.
- For networks configured with eBGP multihop, a label switched path (LSP) must be established between nonadjacent routers (RFC 3107).
- PPP encapsulation on the ASBRs is not supported with this feature.

Information About MPLS VPN - Interautonomous System Support

- MPLS VPN Interautonomous System Benefits, page 218
- Interautonomous System Communication with ASBRs, page 218
- Interautonomous System Configurations Supported in an MPLS VPN, page 218
- How Information Is Exchanged in an MPLS VPN Inter-AS with ASBRs, page 219
- Load Sharing with MPLS VPN Inter-AS ASBRs, page 225

MPLS VPN Interautonomous System Benefits

An MPLS VPN Inter-AS provides the following benefits:

- Allows a VPN to cross more than one service provider backbone—Service providers running separate autonomous systems can jointly offer MPLS VPN services to the same end customer. A VPN can begin at one customer site and traverse different VPN service provider backbones before arriving at another site of the same customer. Before the release of this feature, MPLS VPN could only traverse a single BGP autonomous system service provider backbone. The MPLS VPN - Interautonomous System Support feature allows multiple autonomous systems to form a continuous (and seamless) network between customer sites of a service provider.
- Allows a VPN to exist in different areas—A service provider can create a VPN in different geographic areas. Having all VPN traffic flow through one point (between the areas) allows for better rate control of network traffic between the areas.
- Allows confederations to optimize internal Border Gateway Protocol (iBGP) meshing—iBGP meshing in an autonomous system is more organized and manageable. You can divide an autonomous system into multiple, separate subautonomous systems and then classify them into a single confederation (even though the entire VPN backbone appears as a single autonomous system). This capability allows a service provider to offer MPLS VPNs across the confederation because it supports the exchange of labeled VPN-IPv4 NLRI between the subautonomous systems that form the confederation.

Interautonomous System Communication with ASBRs

Separate autonomous systems from different service providers can communicate by exchanging IPv4 NLRI in the form of VPN-IPv4 addresses. The ASBRs use eBGP to exchange that information. Then an Interior Gateway Protocol (IGP) distributes the network layer information for VPN-IPv4 prefixes throughout each VPN and each autonomous system. Routing information uses the following protocols:

- Within an autonomous system, routing information is shared using an IGP.
- Between autonomous systems, routing information is shared using an eBGP. An eBGP allows a service provider to set up an interdomain routing system that guarantees the loop-free exchange of routing information between separate autonomous systems.

The primary function of an eBGP is to exchange network reachability information between autonomous systems, including information about the list of autonomous system routes. The autonomous systems use EGBP border edge routers to distribute the routes, which include label switching information. Each border edge router rewrites the next hop and MPLS labels. See the How Information Is Exchanged in an MPLS VPN Inter-AS with ASBRs, page 219 section for more information.

Interautonomous System Configurations Supported in an MPLS VPN

Interautonomous system configurations supported in an MPLS VPN can include:

- Interprovider VPN—MPLS VPNs that include two or more autonomous systems, connected by separate border edge routers. The autonomous systems exchange routes using eBGP. No IGP or routing information is exchanged between the autonomous systems.
- BGP confederations—MPLS VPNs that divide a single autonomous system into multiple subautonomous systems, and classify them as a single, designated confederation. The network recognizes the confederation as a single autonomous system. The peers in the different autonomous systems communicate over eBGP sessions; however, they can exchange route information as if they were iBGP peers.

How Information Is Exchanged in an MPLS VPN Inter-AS with ASBRs

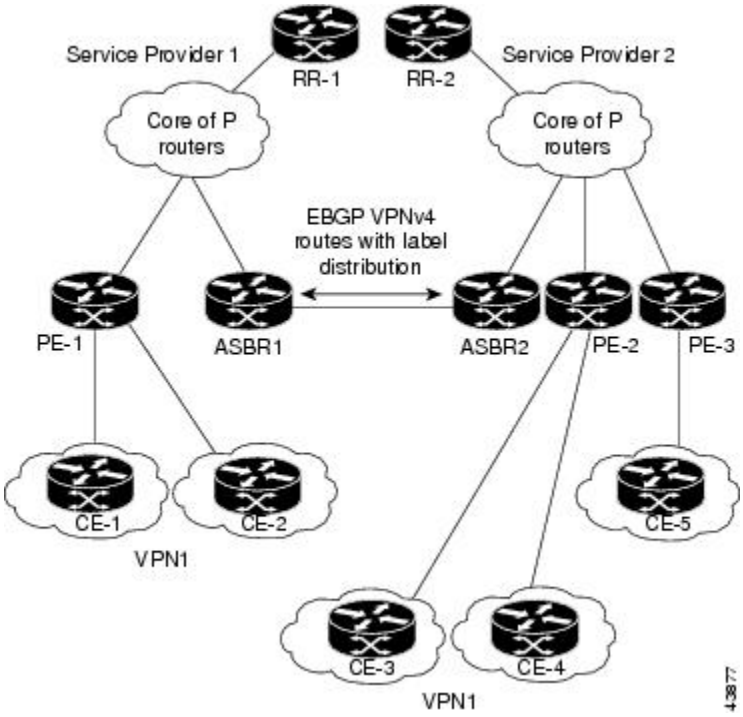
This section contains the following topics about how information is exchanged in an MPLS VPN Inter-AS with ASBRs exchanging VPN-IPv4 addresses:

- Information Sent in an MPLS VPN Inter-AS with ASBRs, page 219
- VPN Routing Information Exchange in an MPLS VPN Inter-AS with ASBRs, page 220
- Packet Forwarding Between MPLS VPN Interautonomous Systems with ASBRs, page 222
- Confederation Configuration for MPLS VPN Inter-AS with ASBRs, page 224

Information Sent in an MPLS VPN Inter-AS with ASBRs

The figure below illustrates one MPLS VPN consisting of two separate autonomous systems. Each autonomous system operates under different administrative control and runs a different IGP. Service providers exchange routing information through eBGP border edge routers (ASBR1, ASBR2).

Figure 11 eBGP Connection Between Two MPLS VPN Interautonomous Systems with ASBRs Exchanging VPN-IPv4 Addresses



The table below describes the process to transmit information in an Inter-As configuration with ASBRs exchanging VPN-IPv4 addresses.

Table 15 Information Transmission Process in an Inter-AS with ASBRs Exchanging VPN-IPv4 AddressesMPLS VPN - Interautonomous System Support

Inter-AS Component	Process Competed During Information Transmission
Provider edge router: PE-1	Assigns a label for a route before distributing that route. The PE router uses the multiprotocol extensions of BGP to transmit label mapping information. The PE router distributes the route as a VPN-IPv4 address. The address label and the VPN identifier are encoded as part of the NLRI.
Route reflectors: RR-1 and RR-2	Reflects VPN-IPv4 internal routes within the autonomous system. The autonomous systems' border edge routers (ASBR1 and ASBR2) advertise the VPN-IPv4 external routes.
eBGP border edge router: ASBR1	Redistributes the route to the next autonomous system (ASBR2). ASBR1 specifies its own address as the value of the eBGP next-hop attribute and assigns a new label. The address ensures the following: • That the next-hop router is always reachable in the service provider (P) backbone network. • That the label assigned by the distributing router is properly interpreted. (The label associated with a route must be assigned by the corresponding next-hop router.)
eBGP border edge router: ASBR2	Redistributes the route in one of the following ways, depending on its configuration: • If the iBGP neighbors are configured with the **neighbor next-hop-self** command, ASBR2 changes the next-hop address of updates received from the eBGP peer, then forwards it. • If the iBGP neighbors are not configured with the **neighbor next-hop-self** command, the next-hop address does not get changed. ASBR2 must propagate a host route for the eBGP peer through the IGP. To propagate the eBGP VPN-IPv4 neighbor host route, use the **redistribute connected subnets** command. The eBGP VPN-IPv4 neighbor host route is automatically installed in the routing table when the neighbor comes up. This is essential to establish the label-switched path between PE routers in different autonomous systems

VPN Routing Information Exchange in an MPLS VPN Inter-AS with ASBRs

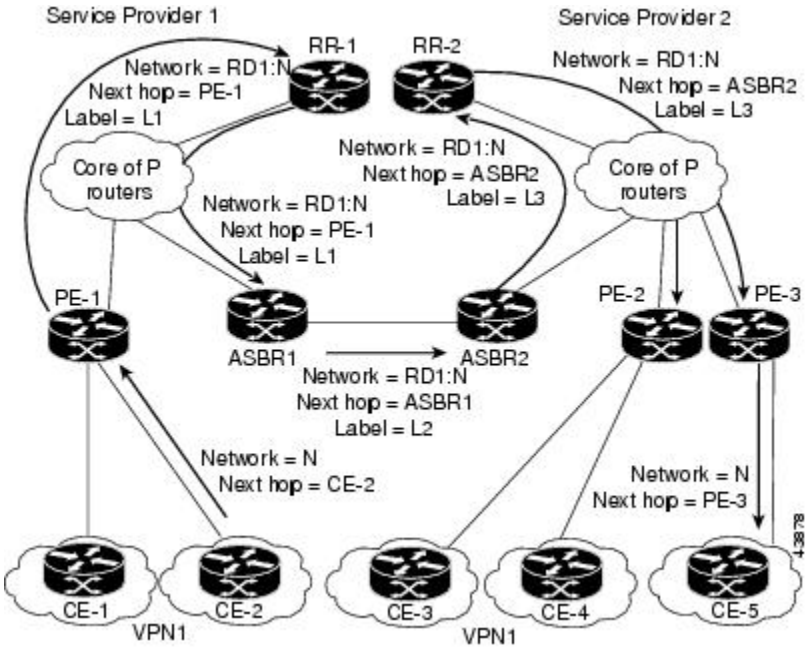
Autonomous systems exchange VPN routing information (routes and labels) to establish connections. To control connections between autonomous systems, the PE routers and eBGP border edge routers maintain a Label Forwarding Information Base (LFIB).

The LFIB manages the labels and routes that the PE routers and eBGP border edge routers receive during the exchange of VPN information.

The figure below illustrates the exchange of VPN route and label information between autonomous systems. The autonomous systems use the following guidelines to exchange VPN routing information:

- Routing information:
  - The destination network (N)
  - The next-hop field associated with the distributing router
  - A local MPLS label (L)
- An RD1: route distinguisher is part of a destination network address. It makes the VPN-IPv4 route globally unique in the VPN service provider environment.
- The ASBRs are configured to change the next hop (next-hop-self) when sending VPN-IPv4 NLRIs to the iBGP neighbors. Therefore, the ASBRs must allocate a new label when they forward the NLRI to the iBGP neighbors.

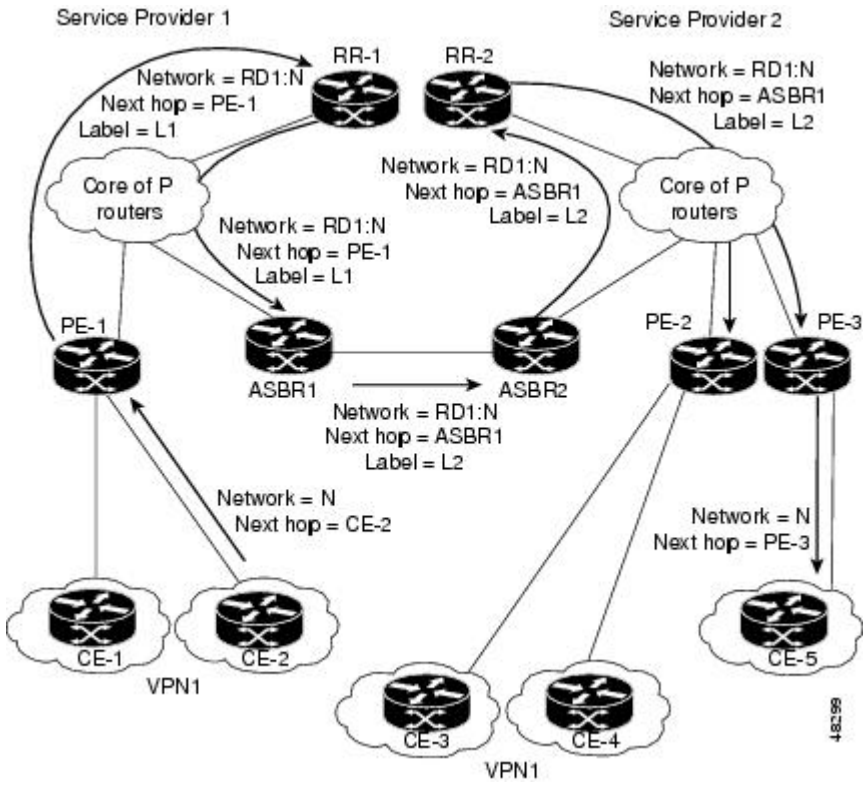
Figure 12 Exchanging Routes and Labels Between MPLS VPN Inter-AS Systems with ASBRs Exchanging VPN-IPv4 Addresses



The figure below illustrates the exchange of VPN route and label information between autonomous systems. The only difference is that ASBR2 is configured with the **redistribute connected** command,

which propagates the host routes to all PEs. The redistribute connected command is necessary because ASBR2 is not configured to change the next-hop address.

Figure 13 Exchanging Routes and Labels with the redistributed connected Command in an MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses



Packet Forwarding Between MPLS VPN Interautonomous Systems with ASBRs

The figure below illustrates how packets are forwarded between autonomous systems in an interprovider network using the following packet forwarding method.

Packets are forwarded to their destination by means of MPLS. Packets use the routing information stored in the LFIB of each PE router and eBGP border edge router.

The service provider VPN backbone uses dynamic label switching to forward labels.

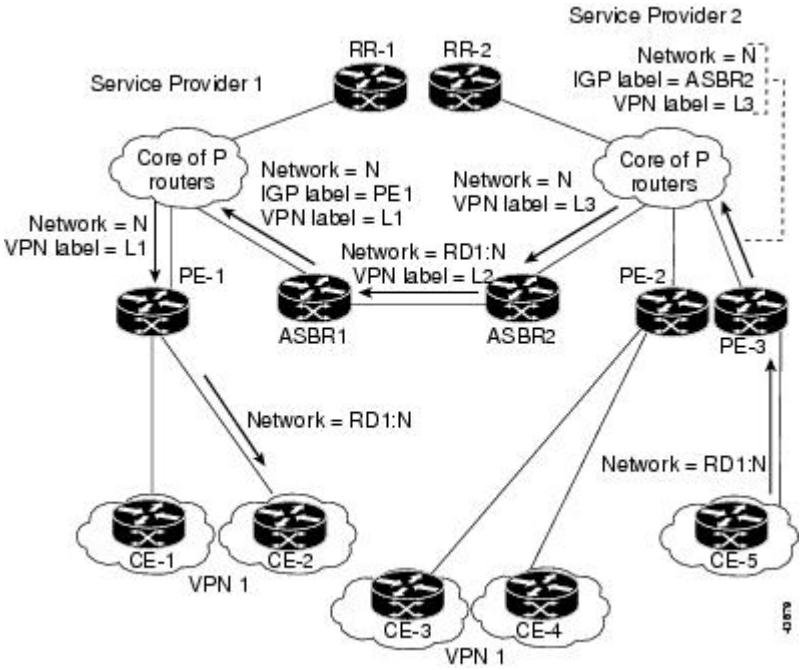
Each autonomous system uses standard multilevel labeling to forward packets between the edges of the autonomous system routers (for example, from CE-5 to PE-3). Between autonomous systems, only a single level of labeling is used, corresponding to the advertised route.

A data packet carries two levels of labels when traversing the VPN backbone:

- The first label (IGP route label) directs the packet to the correct PE router or eBGP border edge router. (For example, the IGP label of ASBR2 points to the ASBR2 border edge router.)

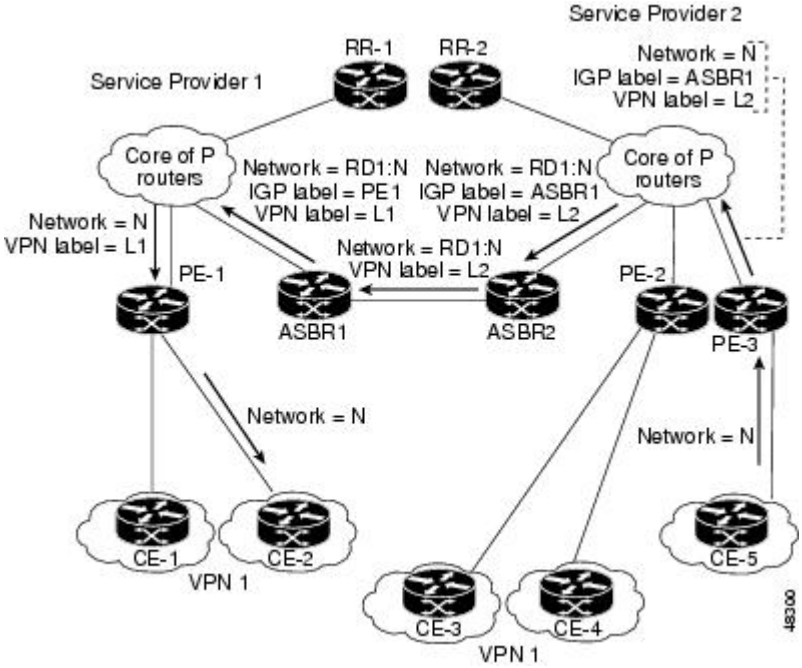
- The second label (VPN route label) directs the packet to the appropriate PE router or eBGP border edge router.

Figure 14 Packet Forwarding Between MPLS VPN Interautonomous Systems with ASBRs Exchanging VPN-IPv4 Addresses



The figure below shows the same packet forwarding method, except the eBGP router (ASBR1) forwards the packet without reassigning it a new label.

Figure 15 Forwarding Packets Without a New Label Assignment Between MPLS VPN Interautonomous Systems with ASBRs Exchanging VPN-IPv4 Addresses



Confederation Configuration for MPLS VPN Inter-AS with ASBRs

A confederation is multiple subautonomous systems grouped together. A confederation reduces the total number of peer devices in an autonomous system. A confederation divides an autonomous system into subautonomous systems and assigns a confederation identifier to the autonomous systems. A VPN can span service providers running in separate autonomous systems or in multiple subautonomous systems that form a confederation.

In a confederation, each subautonomous system is fully meshed with other subautonomous systems. The subautonomous systems communicate using an IGP, such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS). Each subautonomous system also has an eBGP connection to the other subautonomous systems. The confederation eBGP (CeBGP) border edge routers forward next-hop-self addresses between the specified subautonomous systems. The next-hop-self address forces the BGP to use a specified address as the next hop rather than letting the protocol choose the next hop.

You can configure a confederation with separate subautonomous systems in either of two ways:

- You can configure a router to forward next-hop-self addresses between only the CeBGP border edge routers (both directions). The subautonomous systems (iBGP peers) at the subautonomous system border do not forward the next-hop-self address. Each subautonomous system runs as a single IGP domain. However, the CeBGP border edge router addresses are known in the IGP domains.
- You can configure a router to forward next-hop-self addresses between the CeBGP border edge routers (both directions) and within the iBGP peers at the subautonomous system border. Each subautonomous system runs as a single IGP domain but also forwards next-hop-self addresses between the PE routers in the domain. The CeBGP border edge router addresses are known in the IGP domains.

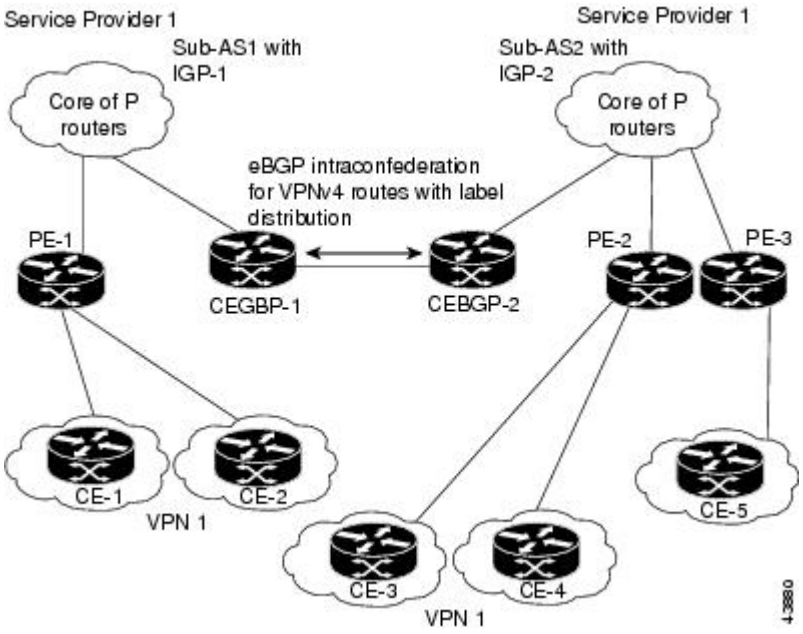
Note The second and third figures above illustrate how two autonomous systems exchange routes and forward packets. Subautonomous systems in a confederation use a similar method of exchanging routes and forwarding packets.

The figure below illustrates a typical MPLS VPN confederation configuration. In this confederation configuration:

- The two CeBGP border edge routers exchange VPN-IPv4 addresses with labels between the two subautonomous systems.
- The distributing router changes the next-hop addresses and labels and uses a next-hop-self address.

IGP-1 and IGP-2 know the addresses of CeBGP-1 and CeBGP-2.

Figure 16 eBGP Connection Between Two Subautonomous Systems in a Confederation



In this confederation configuration:

- CeBGP border edge routers function as neighboring peers between the subautonomous systems. The subautonomous systems use eBGP to exchange route information.
- Each CeBGP border edge router (CeBGP-1, CeBGP-2) assigns a label for the route before distributing the route to the next subautonomous system. The CeBGP border edge router distributes the route as a VPN-IPv4 address by using the multiprotocol extensions of BGP. The label and the VPN identifier are encoded as part of the NLRI.
- Each PE and CeBGP border edge router assigns its own label to each VPN-IPv4 address prefix before redistributing the routes. The CeBGP border edge routers exchange VPN-IPv4 addresses with the labels. The next-hop-self address is included in the label (as the value of the eBGP next-hop attribute). Within the subautonomous systems, the CeBGP border edge router address is distributed throughout the iBGP neighbors, and the two CeBGP border edge routers are known to both confederations.

Load Sharing with MPLS VPN Inter-AS ASBRs

Before the MPLS VPN - Interautonomous System Support feature, if multiple paths existed across ASBRs, BGP executed the best path algorithm and marked only one of the paths as the best path. This path was added to the routing table and became the only path that was used for forwarding traffic between ASBRs.

The MPLS VPN—Multipath Support for Inter-AS VPNs feature extends the functionality of BGP so that it can pick one path as the best path and mark the other legitimate paths between ASBRs as multipath. This allows the load sharing of traffic among the different multipaths and the best path to reach the destination. No Routing Information Base (RIB) or Cisco Express Forwarding entries are associated with the VPN-IPv4 prefixes.

The MPLS VPN—Multipath Support for Inter-AS VPNs feature applies to ASBRs that do not have a VPN routing and forwarding (VRF) instance configuration. BGP installs a number of learned VPN-IPv4 prefixes

into the MPLS forwarding table (LFIB). VPN-IPv4 entries in the LFIB consist of the Route Distinguisher (RD) and the IPv4 prefix and are called VPNv4 entries.

The **maximum-paths** command is used to set the number of parallel (equal-cost) routes that BGP installs in the routing table to configure multipath load sharing. The number of paths that can be configured is determined by the version of Cisco IOS software. The following list shows the limits:

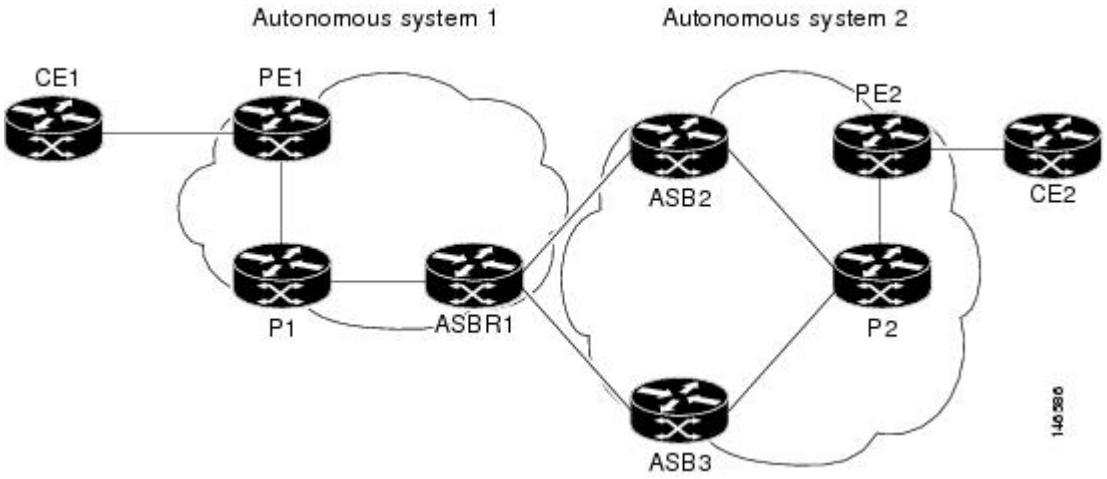
- Cisco IOS Release 12.0S-based software: 8 paths
- Cisco IOS Release 12.3T-based software: 16 paths
- Cisco IOS Release 12.2S-based software: 32 paths

The MPLS VPN—Multipath Support for Inter-AS VPNs feature requires that you configure the **maximum-paths** *number-of-paths* command in address family configuration mode.

Note The **maximum-paths** command cannot be configured with the **maximum-paths eibgp** command for the same BGP routing process.

The figure below shows an example of VPNv4 load balancing for ASBRs in an Inter-AS network. In this example, ASBR1 load balances the traffic from the CE router CE1 to CE2 using the two available links—ASBR2 and ASBR3.

Figure 17 Example of VPNv4 Load Balancing for ASBRs in an Inter-AS Network



When you configure an ASBR for VPNv4 load balancing, you must configure the **next-hop-self** command for the iBGP peers. Without this command, the next hop that is propagated to the iBGP peer is the ASBR2 address or the ASBR3 address, depending on which one BGP selects as the best path. Configuring the **next-hop-self** command provides direct VPNv4 forwarding entries in the MPLS forwarding table for the VPNv4 prefixes learned from the remote ASBRs. VPNv4 forwarding entries are not created if you do not configure the **next-hop-self** command.

Note If the number of forwarding entries in the MPLS forwarding table on the system or on a line card is a concern for your network, we recommend that you do not enable VPNv4 multipath on ASBRs.

How to Configure MPLS VPN - Interautonomous System Support

Perform the following tasks to configure MPLS VPN Inter-AS with ASBRs exchanging VPN-IPv4 addresses:

- Configuring an eBGP ASBR to Exchange MPLS VPN-IPv4 Addresses, page 227
- Configuring eBGP Routing to Exchange MPLS VPN Routes Between Subautonomous Systems in a Confederation, page 238
- Verifying Inter-AS for ASBRs Exchanging MPLS VPN-IPv4 Addresses, page 241
- Configuring eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs, page 243
- Verifying eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs, page 248

Configuring an eBGP ASBR to Exchange MPLS VPN-IPv4 Addresses

Perform one of the following tasks to configure an eBGP ASBR to exchange MPLS VPN-IPv4 routes with another autonomous system:

- Configuring Peering with Directly Connected Interfaces Between ASBRs, page 227
- Configuring Peering of the Loopback Interface of Directly Connected ASBRs, page 229

Configuring Peering with Directly Connected Interfaces Between ASBRs

Perform this task to configure peering with directly connected interfaces between ASBRs so that the ASBRs can distribute BGP routes with MPLS labels.

The figure below shows the configuration for the peering with directly connected interfaces between ASBRs. This configuration is used as the example in the tasks that follow.

Figure 18 Configuration for Peering with Directly Connected Interfaces Between ASBRs

Note When eBGP sessions come up, BGP automatically generates the **mpls bgp forwarding** command on the connecting interface.

Note Issue the **redistribute connected subnets** command in the IGP configuration portion of the router to propagate host routes for VPN-IPv4 eBGP neighbors to other routers and provider edge routers. Alternatively, you can specify the next-hop-self address when you configure iBGP neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default route-target filter**
5. **address-family vpnv4** [**unicast**]
6. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
7. **neighbor** {*ip-address* | *peer-group-name*} **activate**
8. **exit-address-family**
9. **end**

DETAILED STEPS

	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	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and places the router in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535. In this instance an eBGP routing process is configured.
Step 4	**no bgp default route-target filter** **Example:** `Router(config-router)# no bgp default route-target filter`	Disables BGP route-target community filtering. All received BGP VPN-IPv4 routes are accepted by the router. Accepting VPN-IPv4 routes is the desired behavior for a router configured as an ASBR.

	Command or Action	Purpose
Step 5	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode. • The **unicast** keyword specifies a unicast prefix. This command configures a routing session to carry VPN-IPv4 addresses across the VPN backbone. Each address is globally unique by the addition of an 8-byte RD.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor 10.0.0.1 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs. The address of the eBGP neighbor or the eBGP peer group is identified to the specified autonomous system.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. These commands activate the advertisement of the VPNv4 address family to a neighboring eBGP router or an eBGP peer group.
Step 8	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits from the address family configuration mode.
Step 9	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

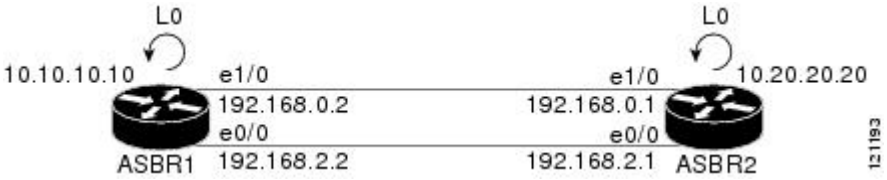
Configuring Peering of the Loopback Interface of Directly Connected ASBRs

This functionality is provided with the release of the MPLS VPN - Interautonomous System Support feature on Cisco IOS Release 12.0(29)S and later releases. An eBGP session configured between loopbacks of directly connected ASBRs allows load sharing between loopback addresses.

Perform the following tasks in this section to configure peering of loopback interfaces of directly connected ASBRs:

The figure below shows the loopback configuration for directly connected ASBR1 and ASBR2 routers. This configuration is used as the example in the tasks that follow.

Figure 19 Loopback Interface Configuration for Directly Connected ASBR1 and ASBR2 Routers



- Configuring Loopback Interface Addresses for Directly Connected ASBRs, page 230
- Examples, page 231
- Configuring Static Routes to the eBGP Neighbor Loopback, page 231
- Examples, page 233
- Configuring Forwarding on the Directly Connected Interfaces, page 233
- Examples, page 234
- Configuring an eBGP Session Between the Loopbacks, page 235
- Examples, page 238

Configuring Loopback Interface Addresses for Directly Connected ASBRs

Perform the following task to configure loopback interface addresses for directly connected ASBRs.

Note Loopback addresses need to be configured for each directly connected ASBR. That is, configure a loopback address for ASBR1 and for ASBR2 (see the figure above).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *interface number*
4. **ip address** *ip-address mask* [**secondary**]
5. **end**

DETAILED STEPS

	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 loopback** *interface number* **Example:** `Router(config)# interface loopback 0`	Configures a software-only virtual interface that emulates an interface that is always up. • The *interface-number* argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces that you can create.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.10.10.10 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Examples

The following example shows the configuration of a loopback address for ASBR1:

```
configure terminal
interface loopback 0
 ip address 10.10.10.10 255.255.255.255
```

The following example shows the configuration of a loopback address for ASBR2:

```
configure terminal
interface loopback 0
 ip address 10.20.20.20 255.255.255.255
```

Configuring Static Routes to the eBGP Neighbor Loopback

Perform the following task to configure /32 static routes to the eBGP neighbor loopback.

A /32 static route is established with the following commands:

```
Router(config)# ip route X.X.X.X 255.255.255.255 Ethernet 1/0 Y.Y.Y.Y
Router(config)# ip route X.X.X.X 255.255.255.255 Ethernet 1/0 Z.Z.Z.Z
```

Where *X.X.X.X* is the neighboring loopback address and Ethernet 1/0 and Ethernet 0/0 are the links connecting the peering routers. *Y.Y.Y.Y* and *Z.Z.Z.Z* are the respective next-hop addresses on the interfaces.

Note You need to configure /32 static routes on each of the directly connected ASBRs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type ip-address interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*]
4. **end**

DETAILED STEPS

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 route** *prefix mask* {*ip-address* | *interface-type ip-address interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Router(config)# ip route 10.20.20.20 255.255.255.255 Ethernet 1/0 192.168.0.1`	Establishes static routes. • The *prefix* argument is the IP route prefix for the destination. • The *mask* argument is the prefix mask for the destination. • The *ip-address* argument is the IP address of the next hop that you can use to reach the specified network. • The *interface-type* and *interface-number* arguments are the network interface type and interface number. • The *distance* argument is an administrative distance. • The *name* argument applies a name to the specified route. • The **permanent** keyword specifies that the route is not to be removed, even if the interface shuts down. • The **tag** *tag* keyword-argument pair names a tag value that can be used as a "match" value for controlling redistribution through the use of route maps.
Step 4 **end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Examples

The following example shows the configuration of a /32 static route from the ASBR1 router to the loopback address of the ASBR2 router:

```
configure terminal
ip route 10.20.20.20 255.255.255.255 e1/0 192.168.0.1
ip route 10.20.20.20 255.255.255.255 e0/0 192.168.2.1
```

The following example shows the configuration of a /32 static route from the ASBR2 router to the loopback address of the ASBR1 router:

```
configure terminal
ip route vrf vpn1 10.10.10.10 255.255.255.255 Ethernet 1/0 192.168.0.2
ip route vrf vpn1 10.10.10.10 255.255.255.255 Ethernet 0/0 192.168.2.2
```

Configuring Forwarding on the Directly Connected Interfaces

Perform this task to configure forwarding on the directly connected interfaces.

This task is required for sessions between loopbacks. In the Configuring Static Routes to the eBGP Neighbor Loopback, page 231 task, Ethernet 1/0 and Ethernet 0/0 are the connecting interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type slot/port*
4. **ip address** *ip-address mask* [**secondary**]
5. **mpls bgp forwarding**
6. **exit**
7. Repeat Steps 3, 4, and 5 for another connecting interface (Ethernet 0/0).
8. **end**

DETAILED STEPS

	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** *interface-type slot/port* **Example:** `Router(config)# interface ethernet 1/0`	Configures an interface type and enters interface configuration mode. • The *interface-type* argument is the type of interface to be configured. • The *slot* argument is the slot number. Refer to the appropriate hardware manual for slot and port information. • The */port* keyword and argument are the port number. Refer to the appropriate hardware manual for slot and port information.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 192.168.0.2 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 5	**mpls bgp forwarding** **Example:** `Router(config-if)# mpls bgp forwarding`	Configures BGP to enable MPLS forwarding on connecting interfaces.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits to global configuration mode.
Step 7	Repeat Steps 3, 4, and 5 for another connecting interface (Ethernet 0/0).	
Step 8	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Examples

The following example shows the configuration of BGP MPLS forwarding on the interfaces connecting the ASBR1 router with the ASBR2 router:

```
configure terminal
interface ethernet 1/0
 ip address 192.168.0.2 255.255.255.0
 mpls bgp forwarding
 exit
!
interface ethernet 0/0
 ip address 192.168.2.2 255.255.255.0
```

```
 mpls bgp forwarding
 exit
```

The following example shows the configuration of BGP MPLS forwarding on the interfaces connecting the ASBR2 router with the ASBR1 router:

```
configure terminal
interface ethernet 1/0
 ip address 192.168.0.1 255.255.255.0
 mpls bgp forwarding
 exit
!
interface ethernet 0/0
 ip address 192.168.2.1 255.255.255.0
 mpls bgp forwarding
 exit
```

Configuring an eBGP Session Between the Loopbacks

Perform the following tasks to configure an eBGP session between the loopbacks.

Note You need to configure an EGBP session between loopbacks on each directly connected ASBR.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default route-target filter**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check**
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
8. **address-family vpnv4** [**unicast**]
9. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
10. **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
11. **end**
12. **show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]

DETAILED STEPS

	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	**router bgp** *as-number* **Example:** `Router(config)# router bgp 200`	Configures the BGP routing process. • The *as-number* indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along.
Step 4	**no bgp default route-target filter** **Example:** `Router(config-router)# no bgp default route-target filter`	Disables BGP route-target filtering. All received BGP VPN-IPv4 routes are accepted by the router.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.20.20.20 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group. • The *as-number* argument is the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check** **Example:** `Router(config-router)# neighbor 10.20.20.20 disable-connected-check`	Allows peering between loopbacks. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group.

	Command or Action	Purpose
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Router(config-router)# neighbor 10.20.20.20 update-source loopback 0`	Allows BGP sessions in Cisco IOS releases to use any operational interface for TCP connections. • The *ip-address* argument is the IPv4 address of the BGP-speaking neighbor. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons. • The *peer-group-name* argument is the name of a BGP peer group. • The *interface-type* argument is the interface type. • The *interface-number* argument is the interface number.
Step 8	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing protocols such as BGP, Routing Information Protocol (RIP), and static routing. • The **vpnv4** keyword configures sessions that carry customer VPN-IPv4 prefixes, each of which has been made globally unique by the addition of an 8-byte route distinguisher. • The **unicast** keyword specifies unicast prefixes.
Step 9	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Router(config-router-af)# neighbor 10.20.20.20 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring router. • The *peer-group-name* argument is the name of a BGP peer group. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.
Step 10	**neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **Example:** `Router(config-router-af)# neighbor 10.20.20.20 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring router. • The *peer-group-name*argument is the name of a BGP peer group. • The **both** keyword specifies that both standard and extended communities will be sent. • The **standard** keyword specifies that only standard communities will be sent. • The **extended** keyword specifies that only extended communities will be sent.
Step 11	**end** **Example:** `Router(config-router-af)# end`	Exits to privileged EXEC mode.

Command or Action	Purpose
Step 12 **show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**] **Example:** `Router# show mpls forwarding-table`	Displays the contents of the MPLS LFIB. Use this command to verify that load balancing occurs between loopbacks. You need to ensure that the MPLS LFIB entry for the neighbor route lists the available paths and interfaces.

Examples

The following example shows the configuration for VPNv4 sessions on the ASBR1 router:

```
configure terminal
router bgp 200
 bgp log-neighbor-changes
 neighbor 10.20.20.20 remote-as 100
 neighbor 10.20.20.20 disable-connected-check
 neighbor 10.20.20.20 update-source loopback 0
!
 address-family vpnv4
 neighbor 10.20.20.20 activate
 neighbor 10.20.20.20 send-community extended
 end
```

The following example shows the configuration for VPNv4 sessions on the ASBR2:

```
configure terminal
router bgp 100
 bgp log-neighbor-changes
 neighbor 10.10.10.10 remote-as 200
 neighbor 10.10.10.10 disable-connected-check
 neighbor 10.10.10.10 update-source Loopback 0
!
 address-family vpnv4
 neighbor 10.10.10.10 activate
 neighbor 10.10.10.10 send-community extended
 end
```

Configuring eBGP Routing to Exchange MPLS VPN Routes Between Subautonomous Systems in a Confederation

Perform this task to configure eBGP routing to exchange MPLS VPN routes between subautonomous systems in a confederation.

Note To ensure that the host routes for VPN-IPv4 eBGP neighbors are propagated (by means of the IGP) to the other routers and provider edge routers, specify the **redistribute connected** command in the IGP configuration portion of the CeBGP router. If you are using OSPF, make sure that the OSPF process is not enabled on the CeBGP interface where the “redistribute connected” subnet exists.

Note In this confederation, subautonomous system IGP domains must know the addresses of CeBGP-1 and CeBGP-2. If you do not specify a next-hop-self address as part of the router configuration, ensure that the addresses of all PE routers in the subautonomous system are distributed throughout the network, not just the addresses of CeBGP-1 and CeBGP-2.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *sub-autonomous-system*
4. **bgp confederation identifier** *as-number*
5. **bgp confederation peers** *sub-autonomous-system*
6. **no bgp default route-target filter**
7. **address-family vpnv4** [**unicast**]
8. **neighbor** *peer-group-name* **remote-as** *as-number*
9. **neighbor** *peer-group-name* **next-hop-self**
10. **neighbor** *peer-group-name* **activate**
11. **exit-address-family**
12. **end**

DETAILED STEPS

	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	**router bgp** *sub-autonomous-system* **Example:** `Router(config)# router bgp 2`	Enters router configuration mode, creates an eBGP routing process, and assigns it an autonomous system number. The subautonomous system number is passed along to identify the router to eBGP routers in other subautonomous systems.

	Command or Action	Purpose
Step 4	**bgp confederation identifier** *as-number* **Example:** `Router(config-router)# bgp confederation identifier 100`	Defines an eBGP confederation by specifying a confederation identifier associated with each subautonomous system. The subautonomous systems appear as a single autonomous system.
Step 5	**bgp confederation peers** *sub-autonomous-system* **Example:** `Router(config-router)# bgp confederation peers 1`	Specifies the subautonomous systems that belong to the confederation (identifies neighbors of other subautonomous systems within the confederation as special eBGP peers).
Step 6	**no bgp default route-target filter** **Example:** `Router(config-router)# no bgp default route-target filter`	Disables BGP route-target community filtering. All received BGP VPN-IPv4 routes are accepted by the router.
Step 7	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode and configures a routing session to carry VPNv4 addresses across the VPN backbone. Each address has been made globally unique by the addition of an 8-byte route distinguisher (RD). • The **unicast** keyword specifies a unicast prefix.
Step 8	**neighbor** *peer-group-name* **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor R remote-as 1`	Specifies a neighboring eBGP peer group. This eBGP peer group is identified to the specified subautonomous system.
Step 9	**neighbor** *peer-group-name* **next-hop-self** **Example:** `Router(config-router-af)# neighbor R next-hop-self`	Advertises the router as the next hop for the specified neighbor. If you specify a next-hop-self address as part of the router configuration, you do not need to use the **redistribute connected** command.
Step 10	**neighbor** *peer-group-name* **activate** **Example:** `Router(config-router-af)# neighbor R activate`	Activates the advertisement of the VPNv4 address family to a neighboring PE router in the specified subautonomous system.

Command or Action	Purpose
Step 11 **exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 12 **end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Verifying Inter-AS for ASBRs Exchanging MPLS VPN-IPv4 Addresses

Perform this task to verify that Inter-AS for ASBRs Exchanging MPLS VPN-IPv4 addresses operates as you expected.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4 all**
3. **show ip bgp vpnv4 all labels**
4. **show mpls forwarding-table**
5. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if required. For example:

Example:

```
Router> enable
Router#
```

Step 2 **show ip bgp vpnv4 all**

Use this command to verify that all VPNv4 information in the BGP table on the ASBR is as you expected. For example:

Example:

```
Router# show ip bgp vpnv4 all

BGP table version is 99, local router ID is 172.16.10.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin coeds: i - IGP, e - EGP, ? incomplete
```

```
  Network          Next Hop       Metric  LocPrf  Weight Path
Route Distinguisher 100:1
*> 10.1.1.0/24     10.1.1.1           50     100       0 200 ?
* i                 10.1.1.5          100     100       0 200 ?
Route Distinguisher 100:2
* 192.168.1.0/24   10.1.1.1          100     100       0 200 ?
*>i                10.1.1.5           50     100       0 200 ?
* 172.16.1.0/24     10.1.1.1          100     100       0 200 ?
+>i                10.1.1.5           50     100       0 200 ?
Route Distinguisher 200:1
*>i172.16.1.0/24    10.1.1.2           50     100       0 200 ?
*> 10.2.1.0/24      0.0.0.0.           0           32768 ?
Route Distinguisher 200:2
*>i172.16.1.0/24   10.1.1.5           50     100       0 200 ?
*>i172.16.1.0/24   10.1.1.5           50     100       0 200 ?
*> 10.2.1.0/24      0.0.0.0            0           32768 ?
```

Step 3 **show ip bgp vpnv4 all labels**

Use this command to display information about all VPNv4 labels. For example:

Example:

```
Router# show ip bgp vpnv4 all labels
      Network          Next Hop        In label/Out label
Route Distinguisher 100:1
      10.1.1.0/24      172.16.10.3       20/29
Route Distinguisher 100:2
      10.1.1.0/24      172.16.10.3       21/35
      10.2.1.0/24      172.16.10.3       24/36
Route Distinguisher 200:1
      10.30.1.0/24       10.1.1.2        23/164
Route Distinguisher 200:2
      10.31.1.0/24       10.1.1.2        27/165
```

Step 4 **show mpls forwarding-table**

Use this command to display the contents of the MPLS LFIB (such as VPNv4 prefix/length and BGP next-hop destination for the route) and see how the VPN-IPv4 LFIB entries appear. For example:

Example:

```
Router# show mpls forwarding-table
Local Outgoing      Prefix            Bytes tag Outgoing       Next Hop
tag   tag or VC     or Tunnel Id      switched  interface
33    33            10.120.4.0/24     0         Hs0/0          point2point
35    27            100:12:10.200.0.1/32 \
                                      0         Hs0/0          point2point
```

In this example, the Prefix field appears as a VPN-IPv4 RD, plus the prefix. If the value is longer than the width of the Prefix column (as illustrated in the last line of the example), the output automatically wraps onto the next line in the forwarding table, preserving column alignment.

Step 5 **exit**

Use this command to exit to user EXEC mode. For example:

Example:

```
Router# exit
Router>
```

Configuring eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs

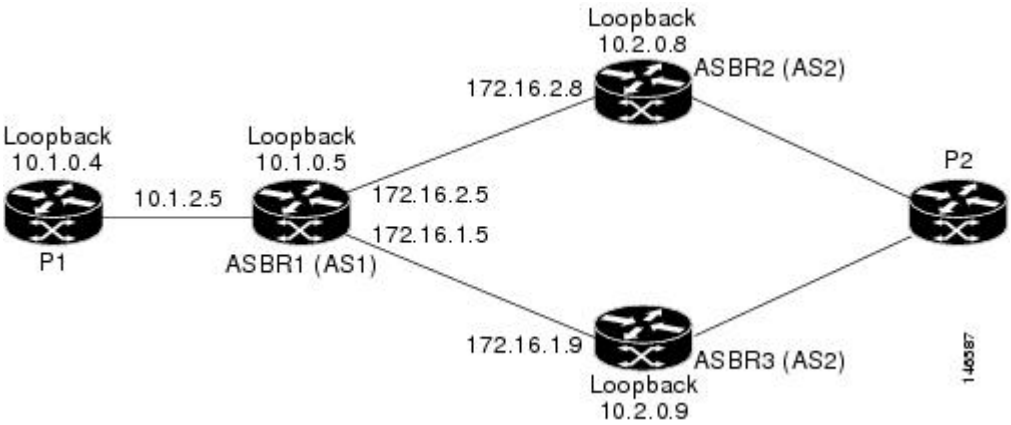
Perform this task to configure eBGP multipath load sharing for MPLS VPN Inter-AS ASBRs exchanging VPN-IPv4 routes. This allows for more efficient use of the LSPs in an interautonomous system network because you can set up the load sharing of traffic among the different multipaths and the best path to reach the destination.

Note The following restrictions apply to configuring multipath load sharing for MPLS VPN Inter-AS ASBRS exchanging VPN-IPv4 routes:

- Per packet load balancing is not supported for this feature. Load balancing for this features works on the IP source and destination hash or on the bottom label in the label stack, depending on the platform and depth of the MPLS label stack.
- If MPLS scalability is an issue for you, we recommend that you do not enable VPNv4 multipath on ASBRs.

The figure below shows an eBGP multipath configuration for three VPN-IPv4 ASBRs. The links from ASBR1 to ASBR2 and ASBR3 have an eBGP VPN-IPv4 session configured. In the figure below, eBGP multipath load sharing is configured on ASBR1. You configure the number of sessions from ASBR1 to ASBR2 and ASBR3 with the **maximum-paths** command in address family configuration mode.

Figure 20 eBGP Multipath Configuration for Three VPN-IPv4 ASBRs



The configurations in the figure above is used as an example for this task and for the task in the Verifying eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs, page 248.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default route-target filter**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type interface-number*
7. **neighbor** {*ip-address* | *peer-group-name*} **next-hop-self**
8. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
9. Repeat Step 8 for each BGP neighbor.
10. **address-family vpnv4** [**unicast**]
11. **neighbor** {*ip-address* | *peer-group-name*} **activate**
12. **neighbor** {*ip-address* | *peer-group-name*} **next-hop-self**
13. **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
14. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
15. **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
16. Repeat Steps 14 and 15 for each BGP neighbor.
17. **maximum-paths** *number-paths*
18. **exit-address-family**
19. **end**

DETAILED STEPS

	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	**router bgp** *as-number* **Example:** `Router(config)# router bgp 1`	Configures an eBGP routing process and places the router in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.

	Command or Action	Purpose
Step 4	**no bgp default route-target filter** **Example:** `Router(config-router)# no bgp default route-target filter`	Disables BGP route-target community filtering. All received VPN-IPv4 routes are accepted by the configured router. Accepting VPN-IPv4 routes is the desired behavior for a router configured as an ASBR.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.1.0.4 remote-as 1`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Router(config-router)# neighbor 10.1.0.4 update-source loopback 0`	Allows BGP sessions to use any operational interface for TCP connections. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *interface-typeinterface-number* arguments specify the type and number for the operational interface. This example shows how to set up BGP TCP connections for the specified neighbor with the IP address of the loopback interface rather than the best local address.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self** **Example:** `Router(config-router)# neighbor 10.1.0.4 next-hop-self`	Configures the router as the next hop for a BGP neighbor or peer group. • The *ip-address* argument specifies the IP address of the BGP neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 172.16.1.9 remote-as 2`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 9	Repeat Step 8 for each BGP neighbor.	—

Command or Action	Purpose
Step 10 **address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode. • The **unicast** keyword specifies a unicast prefix. This command configures a routing session to carry VPN-IPv4 addresses across the VPN backbone. Each address is globally unique by the addition of an 8-byte RD.
Step 11 **neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.1.0.4 activate`	Enables the exchange of information with a neighboring router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 12 **neighbor** {*ip-address* | *peer-group-name*} **next-hop-self** **Example:** `Router(config-router-af)# neighbor 10.1.0.4 next-hop-self`	Configures the router as the next hop for a BGP neighbor or peer group. • The *ip-address* argument specifies the IP address of the BGP neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 13 **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **Example:** `Router(config-router-af)# neighbor 10.1.0.4 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring router. • The *peer-group-name* argument is the name of a BGP peer group. • The **both** keyword specifies that both standard and extended communities will be sent. • The **standard** keyword specifies that only standard communities will be sent. • The **extended** keyword specifies that only extended communities will be sent.
Step 14 **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Router(config-router-af)# neighbor 172.16.1.9 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring router. • The *peer-group-name* argument is the name of a BGP peer group. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.

	Command or Action	Purpose
Step 15	**neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **Example:** `Router(config-router-af)# neighbor 172.16.1.9 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring router. • The *peer-group-name* argument is the name of a BGP peer group. • The **both** keyword specifies that both standard and extended communities will be sent. • The **standard** keyword specifies that only standard communities will be sent. • The **extended** keyword specifies that only extended communities will be sent.
Step 16	Repeat Steps 14 and 15 for each BGP neighbor.	
Step 17	**maximum-paths** *number-paths* **Example:** `Router(config-router-af)# maximum-paths 2`	Configures the maximum number of parallel routes that an IP routing protocol will install into the routing table. • The *number-paths* argument specifies the number of routes to install to the routing table. • See the Load Sharing with MPLS VPN Inter-AS ASBRs, page 225 for information on the number of parallel routes allowed by a specific Cisco IOS release.
Step 18	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits from address family configuration mode.
Step 19	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

- Examples, page 247

Examples

The following example shows the configuration for eBGP multipath for VPNv4 sessions on the ASBR1 router:

```
configure terminal
router bgp 1
 no bgp default route-target filter
 neighbor 10.1.0.4 remote-as 1
 neighbor 10.1.0.4 update-source Loopback 0
 neighbor 10.1.0.4 next-hop-self
 neighbor 172.16.1.9 remote-as 2
 neighbor 172.16.2.8 remote-as 2
```

```
!
 address-family vpnv4
 neighbor 10.1.0.4 activate
 neighbor 10.1.0.4 next-hop-self
 neighbor 10.1.0.4 send-community extended
 neighbor 172.16.1.9 activate
 neighbor 172.16.1.9 send-community extended
 neighbor 172.16.2.8 activate
 neighbor 172.16.2.8 send-community extended
 maximum-paths 2
 exit-address-family
 end
```

Verifying eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs

Perform the following task to verify that eBGP multipath load sharing for MPLS VPN Inter-AS ASBRs is operating as you expect.

The configurations in the figure above are used as an example for the task that follows.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4 all** [**summary**]
3. **show ip bgp vpnv4 all**
4. **show ip bgp vpnv4** [*network*]
5. **show mpls forwarding-table**
6. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if required. For example:

Example:

```
Router> enable
Router#
```

Step 2 **show ip bgp vpnv4 all** [**summary**]

Use this command to verify that all peers are up. for example:

Example:

```
Router# show ip bgp vpnv4 all summary
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.1.0.4        4     1      87      86        5    0    0 01:24:56        2
172.16.1.9      4     2      88      88        5    0    0 01:25:49        2
172.16.2.8      4     2      88      88        5    0    0 01:25:49        2
```

The output shows that all peers expected to be up are up and sending and receiving messages.

Step 3 **show ip bgp vpnv4 all**

Use this command to verify that BGP has paths from both remote ASBRs. For example:

Example:

```
Router# show ip bgp vpnv4 all
   Network          Next Hop            Metric LocPrf Weight Path
.
.
Route Distinguisher: 1:105
*>i192.168.0.1/32   10.1.0.3               11    100      0 ?
*> 192.168.0.2/32   172.16.2.8                            0 2 ?
*                   172.16.1.9                            0 2 ?
*>i192.168.1.0      10.1.0.3                0    100      0 ?
*> 192.168.2.0      172.16.2.8                            0 2 ?
*                   172.16.1.9                            0 2 ?
```

The bold entries in the output confirm that BGP has a path to ASBR2 (172.16.2.8) and to ASBR3 (172.16.1.9).

Step 4 **show ip bgp vpnv4** [*network*]

Use this command to verify that paths are marked as multipath. For example:

Example:

```
Router# show ip bgp vpnv4 192.168.2.0
BGP routing table entry for 1:105:192.168.2.0/24, version 3
Paths: (2 available, best #1, no table)
  Advertised to update-groups:
     2          3
  2
    172.16.2.8 from 172.16.2.8 (10.2.0.8)
      Origin incomplete, localpref 100, valid, external, multipath
, best
      Extended Community: RT:1:100 OSPF DOMAIN ID:0x0005:0x0000000A0200
        OSPF RT:0.0.0.0:2:0 OSPF ROUTER ID:192.168.2.2:512,
      mpls labels in/out 21/25
  2
    172.16.1.9 from 172.16.1.9 (10.2.0.9)
      Origin incomplete, localpref 100, valid, external, multipath
      Extended Community: RT:1:100 OSPF DOMAIN ID:0x0005:0x0000000A0200
        OSPF RT:0.0.0.0:2:0 OSPF ROUTER ID:192.168.2.2:512,
      mpls labels in/out 21/25
```

In the output, the “multipath” and “mpls labels in/out 21/25” are in bold text for example purposes only.

Step 5 **show mpls forwarding-table**

Use this command to verify that MPLS forwarding is properly set up and counters are increasing when traffic is present. For example:

Example:

```
Router# show mpls forwarding-table
Local  Outgoing      Prefix            Bytes Label   Outgoing   Next Hop
Label  Label or VC   or Tunnel Id      Switched      interface
.
.
16     Pop Label     172.16.1.9/32     0             Et1/0      172.16.1.9
17     Pop Label     172.16.2.8/32     0             Et2/0      172.16.2.8
18     Pop Label     10.1.1.0/24       0             Et0/0      10.1.2.4
19     16            10.1.0.3/32       0             Et0/0      10.1.2.4
20     Pop Label     10.1.0.4/32       0             Et0/0      10.1.2.4
21     25            1:105:192.168.2.0/24   \
                                       26658         Et1/0      172.16.1.9
       25            1:105:192.168.2.0/24   \
                                       1180          Et2/0      172.16.2.8
22     24            1:105:192.168.0.2/32   \
                                       15740         Et1/0      172.16.1.9
       24            1:105:192.168.0.2/32   \
                                       0             Et2/0      172.16.2.8
```

```
23      19              1:105:192.168.0.1/32    \
                                        15638           Et0/0       10.1.2.4
24      20              1:105:192.168.1.0/24    \
                                        32740           Et0/0       10.1.2.4
```

Step 6 **exit**

Use this command to exit to user EXEC mode. For example:

Example:

```
Router# exit
Router>
```

Configuration Examples for MPLS VPN - Interautonomous System Support

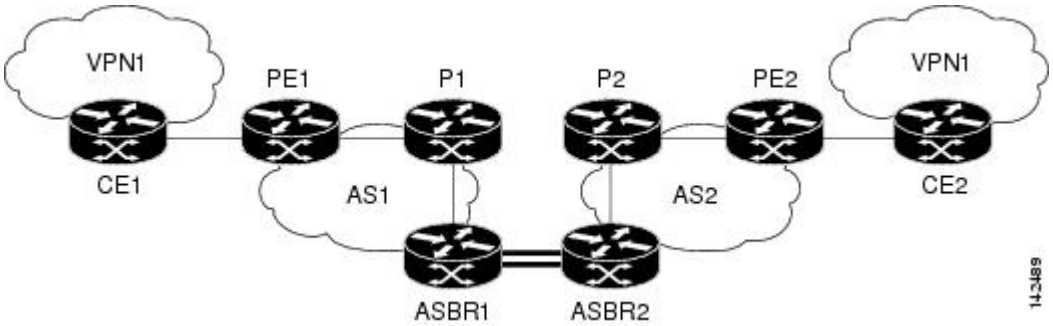
- Configuring Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses Example, page 250
- Configuring Inter-AS with ASBRs in a Confederation Example, page 256
- Configuring eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs Example, page 262

Configuring Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses Example

The network topology in the figure below shows two autonomous systems, which are configured as follows:

- Autonomous system 1 (AS1) contains PE1, P1, ASBR1. The IGP is OSPF.
- Autonomous system 2 (AS2) contains PE2, P2, ASBR2. The IGP is IS-IS.
- CE1 and CE2 belong to the same VPN, which is called VPN1.
- The P routers are route reflectors.
- ASBR1 is configured with the **redistribute connected subnets** command.
- ASBR2 is configured with the **neighbor next-hop-self** command.

Figure 21 Configuring Two Autonomous Systems



- Configuration for Autonomous System 1 CE1 Example for Two Autonomous Systems, page 251

- Configuration for Autonomous System 1 PE1 Example for Two Autonomous Systems, page 251
- Configuration for Autonomous System 1 P1 Example for Two Autonomous Systems, page 252
- Configuration for Autonomous System 1 ASBR1 Example for Two Autonomous Systems, page 253
- Configuration for Autonomous System 2 ASBR2 Example for Two Autonomous Systems, page 253
- Configuration for Autonomous System 2 P2 Example for Two Autonomous Systems, page 254
- Configuration for Autonomous System 2 PE2 Example for Two Autonomous Systems, page 255
- Configuration for Autonomous System 2 CE2 Example for Two Autonomous Systems, page 256

Configuration for Autonomous System 1 CE1 Example for Two Autonomous Systems

The following example shows how to configure the CE1 router in VPN1 in a topology with two autonomous systems (see the figure above):

```
!
hostname CE1
!
interface Loopback 1
 ip address 192.168.0.1 255.255.255.255
!
interface Ethernet 1/0
 description Link to PE1
 ip address 192.168.1.1 255.255.255.0
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
 end
```

Configuration for Autonomous System 1 PE1 Example for Two Autonomous Systems

The following example shows how to configure the PE1 router in autonomous system 1 in a topology with two autonomous systems (see the figure above):

```
!
hostname PE1
!
ip cef
!
ip vrf VPN1
 rd 1:105
 route-target export 1:100
 route-target import 1:100
!
interface Loopback 0
 ip address 10.1.0.3 255.255.255.255
!
interface Ethernet 0/0
 description Link to CE1
 ip vrf forwarding VPN1
 ip address 192.168.1.2 255.255.255.0
!
interface Ethernet 1/0
 description Link to P1
 ip address 10.1.1.3 255.255.255.0
 mpls ip
!
router ospf 10 vrf VPN1
 log-adjacency-changes
 redistribute bgp 1 metric 100 subnets
 network 192.168.0.0 0.0.255.255 area 0
!
router ospf 1
```

```
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 1
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor 10.1.0.4 peer-group R
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor 10.1.0.4 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
 redistribute ospf 10 vrf VPN1
 no auto-summary
 no synchronization
 exit-address-family
!
end
```

Configuration for Autonomous System 1 P1 Example for Two Autonomous Systems

The following example shows how to configure the P1 router in autonomous system 1 in a topology with two autonomous systems (see the figure above):

```
!
hostname P1
!
ip cef
!
interface Loopback 0
 ip address 10.1.0.4 255.255.255.255
!
interface Ethernet 0/0
 description Link to PE1
 ip address 10.1.1.4 255.255.255.0
 mpls ip
!
interface Ethernet 1/0
 description Link to ASBR1
 ip address 10.1.2.4 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
no synchronization
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 1
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor R route-reflector-client
 neighbor 10.1.0.3 peer-group R
 neighbor 10.1.0.5 peer-group R
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor R route-reflector-client
 neighbor 10.1.0.3 activate
 neighbor 10.1.0.5 activate
 exit-address-family
```

```
!
end
```

Configuration for Autonomous System 1 ASBR1 Example for Two Autonomous Systems

The following example shows how to configure ASBR1 in autonomous system 1 in a topology with two autonomous systems (see the figure above):

```
hostname ASBR1
!
ip cef
!
interface Loopback 0
 ip address 10.1.0.5 255.255.255.255
!
interface Ethernet 0/0
 description Link to P1
 ip address 10.1.2.5 255.255.255.0
 mpls ip
!
interface Ethernet 1/0
 description Link to ASBR2
 ip address 172.16.0.1 255.255.255.255
 mpls bgp forwarding
!
router ospf 1
 log-adjacency-changes
 redistribute connected subnets
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 1
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor 10.1.0.4 peer-group R
 neighbor 172.16.0.2 remote-as 2
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor R next-hop-self
 neighbor 10.1.0.4 activate
 neighbor 172.16.0.2 activate
 neighbor 172.16.0.2 send-community extended
 exit-address-family
!
end
```

Configuration for Autonomous System 2 ASBR2 Example for Two Autonomous Systems

The following example shows how to configure ASBR2 in autonomous system 2 in a topology with two autonomous systems (see the figure above):

```
!
hostname ASBR2
!
ip cef
!
interface Loopback 0
 ip address 10.2.0.8 255.255.255.255
 ip router isis
!
interface Ethernet 0/0
 description Link to ASBR1
```

```
 ip address 172.16.0.2 255.255.255.255
 mpls bgp forwarding
!
interface Serial 2/0
 description Link to P2
 ip address 10.2.2.8 255.255.255.0
 ip router isis
 mpls ip
 no fair-queue
 serial restart-delay 0
!
router isis
 net 49.0002.0000.0000.0003.00
!
router bgp 2
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor 10.2.0.7 remote-as 2
 neighbor 10.2.0.7 update-source Loopback 0
 neighbor 10.2.0.7 next-hop-self
 neighbor 172.16.0.1 remote-as 1
 no auto-summary
!
 address-family vpnv4
 neighbor 10.2.0.7 activate
 neighbor 10.2.0.7 send-community extended
 neighbor 10.2.0.7 next-hop-self
 neighbor 172.16.0.1 activate
 neighbor 172.16.0.1 send-community extended
 exit-address-family
!
end
```

Configuration for Autonomous System 2 P2 Example for Two Autonomous Systems

The following example shows how to configure the P2 router in autonomous system 2 in a topology with two autonomous systems (see the figure above):

```
!
hostname P2
!
ip cef
!
interface Loopback 0
 ip address 10.2.0.7 255.255.255.255
 ip router isis
!
interface Ethernet 1/0
 description Link to PE2
 ip address 10.2.1.7 255.255.255.0
 ip router isis
 mpls ip
!
interface Serial 2/0
 description Link to ASBR2
 ip address 10.2.2.7 255.255.255.0
 ip router isis
 mpls ip
 no fair-queue
 serial restart-delay 0
!
router isis
 net 49.0002.0000.0000.0008.00
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 2
 no neighbor R transport path-mtu-discovery
```

```
 neighbor R update-source Loopback 0
 neighbor R route-reflector-client
 neighbor 10.2.0.6 peer-group R
 neighbor 10.2.0.8 peer-group R
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor R route-reflector-client
 neighbor 10.2.0.6 activate
 neighbor 10.2.0.8 activate
 exit-address-family
!
end
```

Configuration for Autonomous System 2 PE2 Example for Two Autonomous Systems

The following example shows how to configure the PE2 router in autonomous system 2 in a topology with two autonomous systems (see the figure above):

```
!
hostname PE2
!
ip cef
!
ip vrf VPN1
 rd 1:105
 route-target export 1:100
 route-target import 1:100
!
interface Loopback 0
 ip address 10.2.0.6 255.255.255.255
 ip router isis
!
interface Ethernet 0/0
 description Link to P2
 ip address 10.2.1.6 255.255.255.0
 ip router isis
 mpls ip
!
interface Serial 2/0
 description Link to CE2
 ip vrf forwarding VPN1
 ip address 192.168.2.2 255.255.255.0
 no fair-queue
 serial restart-delay 0
!
router ospf 10 vrf VPN1
 log-adjacency-changes
 redistribute bgp 2 subnets
 network 192.168.0.0 0.0.255.255 area 0
!
router isis
 net 49.0002.0000.0000.0009.00
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.2.0.7 remote-as 2
 neighbor 10.2.0.7 update-source Loopback 0
 no auto-summary
 !
address-family vpnv4
 neighbor 10.2.0.7 activate
 neighbor 10.2.0.7 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf VPN1
 redistribute connected
 redistribute ospf 10 vrf VPN1
 no auto-summary
```

```
 no synchronization
 exit-address-family
!
end
```

Configuration for Autonomous System 2 CE2 Example for Two Autonomous Systems

The following example shows how to configure the CE2 router in autonomous system 2 in a topology with two autonomous systems (see the figure above):

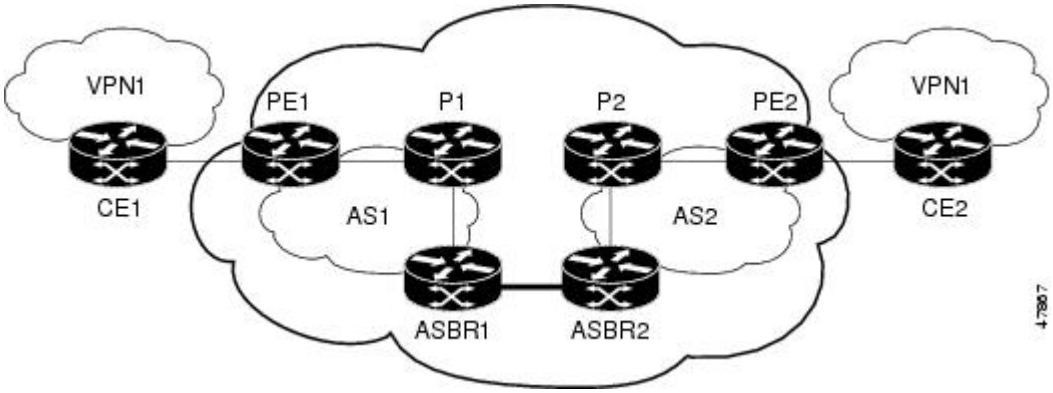
```
!
hostname CE2
!
interface Loopback 0
 ip address 192.168.0.2 255.255.255.255
!
interface Serial 2/0
 description Link to PE2
 ip address 192.168.2.1 255.255.255.0
 no fair-queue
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
end
```

Configuring Inter-AS with ASBRs in a Confederation Example

The network topology in the figure below shows a single Internet service provider (ISP), which is partitioning the backbone with confederations. The autonomous system number of the provider is 100. The two autonomous systems run their own IGPs and are configured as follows:

- Autonomous system 1 (AS1) contains PE1, P1, ASBR1. The IGP is OSPF.
- Autonomous system 2 (AS2) contains PE2, P2, ASBR2. The IGP is IS-IS.
- CE1 and CE2 belong to the same VPN, which is called VPN1.
- The P routers are route reflectors.
- ASBR1 is configured with the **redistribute connected subnets** command.
- ASBR2 is configured with the **neighbor next-hop-self** command.

Figure 22 Configuring Two Autonomous Systems in a Confederation



- Inter-AS Confederation Configuration for Autonomous System 1 CE1 Example, page 257

- Inter-AS Confederation Configuration for Autonomous System 1 PE1 Example, page 257
- Inter-AS Confederation Configuration for Autonomous System 1 P1 Example, page 258
- Inter-AS Confederation Configuration for Autonomous System 1 ASBR1 Example, page 259
- Inter-AS Confederation Configuration for Autonomous System 2 ASBR2 Example, page 259
- Inter-AS Confederation Configuration for Autonomous System 2 P2 Example, page 260
- Inter-AS Confederation Configuration for Autonomous System 2 PE2 Example, page 261
- Inter-AS Confederation Configuration for Autonomous System 2 CE2 Example, page 262

Inter-AS Confederation Configuration for Autonomous System 1 CE1 Example

The following example shows how to configure CE1 in VPN1 in an Inter-AS confederation (see the figure above):

```
!
hostname CE1
!
interface Loopback 1
 ip address 192.168.0.1 255.255.255.255
!
interface Ethernet 1/0
 description Link to PE1
 ip address 192.168.1.1 255.255.255.0
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
end
```

Inter-AS Confederation Configuration for Autonomous System 1 PE1 Example

The following example shows how to configure PE1 in autonomous system 1 in an Inter-AS confederation (see the figure above):

```
hostname PE1
!
ip cef
!
ip vrf VPN1
 rd 1:105
 route-target export 1:100
 route-target import 1:100
!
interface Loopback 0
 ip address 10.1.0.3 255.255.255.255
!
interface Ethernet 0/0
 description Link to CE1
 ip vrf forwarding VPN1
 ip address 192.168.1.2 255.255.255.0
!
interface Ethernet 1/0
 description Link to P1
 ip address 10.1.1.3 255.255.255.0
 mpls ip
!
router ospf 10 vrf VPN1
 log-adjacency-changes
 redistribute bgp 1 metric 100 subnets
 network 192.168.0.0 0.0.255.255 area 0
!
router ospf 1
 log-adjacency-changes
```

```
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 100
 neighbor R peer-group
 neighbor R remote-as 1
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor 10.1.0.4 peer-group R
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor 10.1.0.4 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
 redistribute ospf 10 vrf VPN1
 no auto-summary
 no synchronization
 exit-address-family
!
end
```

Inter-AS Confederation Configuration for Autonomous System 1 P1 Example

The following example shows how to configure P1 in autonomous system 1 in a confederation topology (see the figure above):

```
!
hostname P1
!
ip cef
!
interface Loopback 0
 ip address 10.1.0.4 255.255.255.255
!
interface Ethernet 0/0
 description Link to PE1
 ip address 10.1.1.4 255.255.255.0
 mpls ip
!
interface Ethernet 1/0
 description Link to ASBR1
 ip address 10.1.2.4 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 100
 neighbor R peer-group
 neighbor R remote-as 1
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor R route-reflector-client
 neighbor 10.1.0.3 peer-group R
 neighbor 10.1.0.5 peer-group R
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor R route-reflector-client
 neighbor 10.1.0.3 activate
 neighbor 10.1.0.5 activate
```

```
 exit-address-family
!
end
```

Inter-AS Confederation Configuration for Autonomous System 1 ASBR1 Example

The following example shows how to configure ASBR1 in autonomous system 1 in a confederation topology (see the figure above):

```
!
hostname ASBR1
!
ip cef
!
interface Loopback 0
 ip address 10.1.0.5 255.255.255.255
!
interface Ethernet 0/0
 description Link to P1
 ip address 10.1.2.5 255.255.255.0
 mpls ip
!
interface Ethernet 1/0
 description Link to ASBR2
 ip address 172.16.0.1 255.255.255.255
 mpls bgp forwarding
!
router ospf 1
 log-adjacency-changes
 redistribute connected subnets
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 bgp confederation identifier 100
 bgp confederation peers 2
 neighbor R peer-group
 neighbor R remote-as 1
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor 10.1.0.4 peer-group R
 neighbor 172.16.0.2 remote-as 2
 neighbor 172.16.0.2 next-hop-self
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor R next-hop-self
 neighbor 10.1.0.4 activate
 neighbor 172.16.0.2 activate
 neighbor 172.16.0.2 send-community extended
 neighbor 172.16.0.2 next-hop-self
 exit-address-family
!
end
```

Inter-AS Confederation Configuration for Autonomous System 2 ASBR2 Example

The following example shows how to configure ASBR2 in autonomous system 2 in a confederation topology (see the figure above):

```
!
hostname ASBR2
!
ip cef
!
```

```
interface Loopback 0
 ip address 10.2.0.8 255.255.255.255
 ip router isis
!
interface Ethernet 0/0
 description Link to ASBR1
 ip address 172.16.0.2 255.255.255.255
 mpls bgp forwarding
!
interface Serial 2/0
 description Link to P2
 ip address 10.2.2.8 255.255.255.0
 ip router isis
 mpls ip
no fair-queue
 serial restart-delay 0
!
router isis
 net 49.0002.0000.0000.0003.00
!
router bgp 2
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 bgp confederation identifier 100
 bgp confederation peers 1
 neighbor 10.2.0.7 remote-as 2
 neighbor 10.2.0.7 update-source Loopback 0
 neighbor 10.2.0.7 next-hop-self
 neighbor 172.16.0.1 remote-as 1
 neighbor 172.16.0.1 next-hop-self
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.2.0.7 activate
 neighbor 10.2.0.7 send-community extended
 neighbor 10.2.0.7 next-hop-self
 neighbor 172.16.0.1 activate
 neighbor 172.16.0.1 send-community extended
 neighbor 172.16.0.1 next-hop-self
 exit-address-family
!
end
```

Inter-AS Confederation Configuration for Autonomous System 2 P2 Example

The following example shows how to configure P2 in autonomous system 2 in a confederation topology (see the figure above):

```
!
hostname P2
!
ip cef
!
interface Loopback 0
 ip address 10.2.0.7 255.255.255.255
 ip router isis
!
interface Ethernet 1/0
 description Link to PE2
 ip address 10.2.1.7 255.255.255.0
 ip router isis
 mpls ip
!
interface Serial 2/0
 description Link to ASBR2
 ip address 10.2.2.7 255.255.255.0
 ip router isis
 mpls ip
 no fair-queue
 serial restart-delay 0
```

```
!
router isis
 net 49.0002.0000.0000.0008.00
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 100
 neighbor R peer-group
 neighbor R remote-as 2
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor R route-reflector-client
 neighbor 10.2.0.6 peer-group R
 neighbor 10.2.0.8 peer-group R
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor R route-reflector-client
 neighbor 10.2.0.6 activate
 neighbor 10.2.0.8 activate
 exit-address-family
!
end
```

Inter-AS Confederation Configuration for Autonomous System 2 PE2 Example

The following example shows how to configure PE2 in autonomous system 2 in a confederation topology (see the figure above):

```
!
hostname PE2
!
ip cef
!
ip vrf VPN1
 rd 1:105
 route-target export 1:100
 route-target import 1:100
!
interface Loopback 0
 ip address 10.2.0.6 255.255.255.255
 ip router isis
!
interface Ethernet 0/0
 description Link to P2
 ip address 10.2.1.6 255.255.255.0
 ip router isis
 mpls ip
!
interface Serial 2/0
 description Link to CE2
 ip vrf forwarding VPN1
 ip address 192.168.2.2 255.255.255.0
 no fair-queue
 serial restart-delay 0
!
router ospf 10 vrf VPN1
 log-adjacency-changes
 redistribute bgp 2 subnets
 network 192.168.0.0 0.0.255.255 area 0
!
router isis
 net 49.0002.0000.0000.0009.00
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 100
 neighbor 10.2.0.7 remote-as 2
```

```
 neighbor 10.2.0.7 update-source Loopback 0
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.2.0.7 activate
 neighbor 10.2.0.7 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf VPN1
 redistribute connected
 redistribute ospf 10 vrf VPN1
 no auto-summary
 no synchronization
 exit-address-family
!
end
```

Inter-AS Confederation Configuration for Autonomous System 2 CE2 Example

The following example shows how to configure CE2 in VPN1 in a confederation topology (see the figure above):

```
!
hostname CE2
!
interface Loopback 0
 ip address 192.168.0.2 255.255.255.255
!
interface Serial 2/0
 description Link to PE2
 ip address 192.168.2.1 255.255.255.0
 no fair-queue
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
end
```

Configuring eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs Example

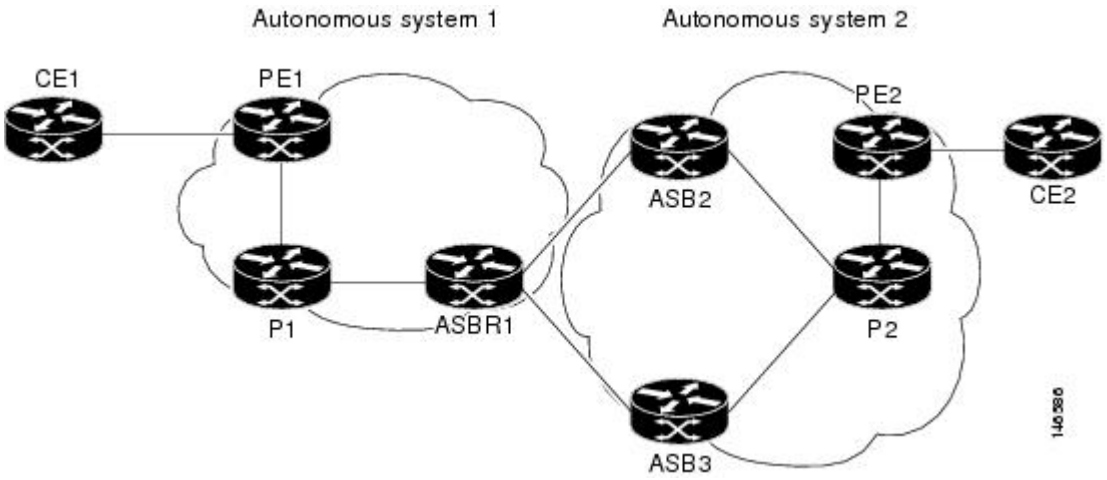
This section includes examples that show how to configure eBGP multipath load sharing for MPLS VPN Inter-AS ASBRS that exchange VPN-IPv4 routes. These configurations support the MPLS VPN - Interautonomous System Support feature.

The network topology in the figure below shows two autonomous systems, which are configured as follows:

- Autonomous system 1 contains PE1, P1, and ASBR1.
- Autonomous system 2 contains PE2, P2, ASBR2, and ASBR3.
- CE1 and CE2 belong to the same VPN, which is called VPN1.
- The P routers are route reflectors.
- ASBR1 and ASBR2 are configured with the **neighbor next-hop-self** command for the iBGP neighbors.

- ASBR1 and ASBR2 are configured with the **maximum paths** commands to set up eBGP multipath load sharing.

Figure 23 Configuring eBGP Multipath Load Sharing Between MPLS Inter-AS ASBRs Exchanging VPN-IPv4 Routes



The following examples shows how to configure eBGP multipath load sharing for MPLS VPN Inter-AS ASBRs that exchange VPN-IPv4 routes. This section includes sample configurations for P1, ASBR1, ASBR2, and P2 routers.

- Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 CE1 Example, page 263
- Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 PE1 Example, page 264
- Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 P1 Example, page 265
- Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 ASBR1 Example, page 265
- Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 ASBR2 Example, page 266
- Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 ASBR3 Example, page 267
- Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 P2 Example, page 268
- Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 PE2 Example, page 268
- Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 CE2 Example, page 269

Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 CE1 Example

The following example shows how to configure CE1 in VPN1 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
!
```

```
hostname CE1
!
interface Loopback 1
 ip address 192.168.0.1 255.255.255.255
!
interface Ethernet 1/0
 description Link to PE1
 ip address 192.168.1.1 255.255.255.0
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
end
```

Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 PE1 Example

The following example shows how to configure PE1 in autonomous system 1 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
!
hostname PE1
!
ip cef
!
ip vrf V1
 rd 1:105
 route-target export 1:100
 route-target import 1:100
!
interface Loopback 0
 ip address 10.1.0.3 255.255.255.255
!
interface Ethernet 0/0
 description Link to CE1
 ip vrf forwarding V1
 ip address 192.168.1.2 255.255.255.0
!
interface Ethernet 1/0
 description Link to P1
 ip address 10.1.1.3 255.255.255.0
 mpls ip
!
router ospf 10 vrf V1
 log-adjacency-changes
 redistribute bgp 1 metric 100 subnets
 network 192.168.0.0 0.0.255.255 area 0
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.1.0.4 remote-as 1
 no neighbor 10.1.0.4 transport path-mtu-discovery
 neighbor 10.1.0.4 update-source Loopback 0
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.1.0.4 activate
 neighbor 10.1.0.4 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf V1
 redistribute ospf 10 vrf V1
 no auto-summary
 no synchronization
 exit-address-family
```

```
!
end
```

Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 P1 Example

The following example shows how to configure P1 in autonomous system 1 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
!
hostname P1
!
ip cef
!
interface Loopback 0
 ip address 10.1.0.4 255.255.255.255
!
interface Ethernet 0/0
 description Link to PE1
 ip address 10.1.1.4 255.255.255.0
 mpls ip
!
interface Ethernet 1/0
 description Link to ASBR1
 ip address 10.1.2.4 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 1
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor R route-reflector-client
 neighbor 10.1.0.3 peer-group R
 neighbor 10.1.0.5 peer-group R
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor R route-reflector-client
 neighbor 10.1.0.3 activate
 neighbor 10.1.0.5 activate
 exit-address-family
!
end
```

Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 ASBR1 Example

The following example shows how to configure ASBR1 in autonomous system 1 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
hostname ASBR1
!
ip cef
!
interface Loopback 0
 ip address 10.1.0.5 255.255.255.255
!
interface Ethernet 0/0
 description Core link to P1
 ip address 10.1.2.5 255.255.255.0
```

```
 mpls ip
!
interface Ethernet 1/0
 description Link to ASBR2
 ip address 172.16.2.5 255.255.255.0
 mpls bgp forwarding
!
interface Serial 3/0
 description Link to ASBR3
 ip address 172.16.1.5 255.255.255.0
 mpls bgp forwarding
 serial restart-delay 0
!
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor 10.1.0.4 remote-as 1
 neighbor 172.16.1.9 remote-as 2
 neighbor 172.16.2.8 remote-as 2
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.1.0.4 activate
 neighbor 10.1.0.4 send-community extended
 neighbor 10.1.0.4 next-hop-self
 neighbor 172.16.1.9 activate
 neighbor 172.16.1.9 send-community extended
 neighbor 172.16.2.8 activate
 neighbor 172.16.2.8 send-community extended
 maximum-paths 2
 exit-address-family
!
end
```

Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 ASBR2 Example

The following example shows how to configure ASBR2 in autonomous system 2 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
!
hostname ASBR2
!
ip cef
!
interface Loopback 0
 ip address 10.2.0.8 255.255.255.255
!
interface Loopback 1
 no ip address
 shutdown
!
interface Ethernet 0/0
 description Link to ASBR1
 ip address 172.16.2.8 255.255.255.0
 mpls bgp forwarding
!
interface Serial 2/0
 description Link to P2
 ip address 10.2.2.8 255.255.255.0
 mpls ip
 no fair-queue
 serial restart-delay 0
!
```

```
router ospf 1
 log-adjacency-changes
 redistribute connected subnets
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 2
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor 10.2.0.7 remote-as 2
 neighbor 10.2.0.7 update-source Loopback 0
 neighbor 10.2.0.7 next-hop-self
 neighbor 172.16.2.5 remote-as 1
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.2.0.7 activate
 neighbor 10.2.0.7 send-community extended
 neighbor 10.2.0.7 next-hop-self
 neighbor 172.16.2.5 activate
 neighbor 172.16.2.5 send-community extended
 exit-address-family
!
end
```

Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 ASBR3 Example

The following example shows how to configure ASBR3 in autonomous system 2 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
!
hostname ASBR3
!
ip cef
!
interface Loopback 0
 ip address 10.2.0.9 255.255.255.255
!
interface Ethernet 0/0
 description Link to ASBR1
 ip address 172.16.1.9 255.255.255.0
 mpls bgp forwarding
!
interface Serial 3/0
 description Link to P2
 ip address 10.2.3.9 255.255.255.0
 mpls ip
 no fair-queue
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 redistribute connected subnets
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 2
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor 10.2.0.7 remote-as 2
 neighbor 10.2.0.7 update-source Loopback 0
 neighbor 10.2.0.7 next-hop-self
 neighbor 172.16.1.5 remote-as 1
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.2.0.7 activate
 neighbor 10.2.0.7 send-community extended
 neighbor 10.2.0.7 next-hop-self
```

```
 neighbor 172.16.1.5 activate
 neighbor 172.16.1.5 send-community extended
 exit-address-family
!
end
```

Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 P2 Example

The following example shows how to configure P2 in autonomous system 2 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
!
hostname P2
!
ip cef
!
interface Loopback 0
 ip address 10.2.0.7 255.255.255.255
!
interface Ethernet 1/0
 description Link to PE2
 ip address 10.2.1.7 255.255.255.0
 mpls ip
!
interface Serial 2/0
 description Link to ASBR2
 ip address 10.2.2.7 255.255.255.0
 mpls ip
 no fair-queue
 serial restart-delay 0
!
interface Serial 3/0
 description Link to ASBR3
 ip address 10.2.3.7 255.255.255.0
 mpls ip
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 2
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor R route-reflector-client
 neighbor 10.2.0.6 peer-group R
 neighbor 10.2.0.8 peer-group R
 neighbor 10.2.0.9 peer-group R
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor R route-reflector-client
 neighbor 10.2.0.6 activate
 neighbor 10.2.0.8 activate
 neighbor 10.2.0.9 activate
 exit-address-family
!
end
!
```

Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 PE2 Example

The following example shows how to configure PE2 in autonomous system 2 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
hostname PE2
!
ip cef
!
ip vrf V1
 rd 1:105
 route-target export 1:100
 route-target import 1:100
!
interface Loopback 0
 ip address 10.2.0.6 255.255.255.255
!
interface Ethernet 0/0
 description Link to P2
 ip address 10.2.1.6 255.255.255.0
 mpls ip
!
interface Serial 2/0
 description Link to CE2
 ip vrf forwarding V1
 ip address 192.168.2.2 255.255.255.0
 no fair-queue
 serial restart-delay 0
!
router ospf 10 vrf V1
 log-adjacency-changes
 redistribute bgp 2 subnets
 network 192.168.0.0 0.0.255.255 area 0
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.2.0.7 remote-as 2
 neighbor 10.2.0.7 update-source Loopback 0
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.2.0.7 activate
 neighbor 10.2.0.7 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf V1
 redistribute connected
 redistribute ospf 10 vrf V1
 no auto-summary
 no synchronization
 exit-address-family
!
end
```

Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 CE2 Example

The following example shows how to configure CE2 in VPN1 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
hostname CE2
!
interface Loopback 0
 ip address 192.168.0.2 255.255.255.255
!
interface Serial 2/0
 description Link to PE2
```

```
 ip address 192.168.2.1 255.255.255.0
 no fair-queue
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
end
```

Additional References

Related Documents

Related Topic	Document Title
Configuration tasks for basic MPLS VPNs	Configuring MPLS VPNs
Configuration tasks for MPLS VPN Inter-AS system exchanging IPv4 routes and MPLS labels	MPLS VPN - Inter-AS—IPv4 BGP Label Distribution
Information about monitoring MPLS VPNs with MIBs	MPLS VPN—SNMP MIB Support

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 software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1164	*Application of the Border Gateway Protocol in the Internet*
RFC 1700	*Assigned Numbers*
RFC 1771	*A Border Gateway Protocol 4*

RFC	Title
RFC 1965	*Autonomous System Confederation for BGP*
RFC 1966	*BGP Route Reflection: An Alternative to Full Mesh iBGP*
RFC 2547	*BGP/MPLS VPNs*
RFC 2842	*Capabilities Advertisement with BGP-4*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 3107	*Carrying Label Information in BGP-4*

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 MPLS VPN - Interautonomous System Support

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 16 Feature Information for MPLS VPN - Interautonomous System SupportMPLS VPN - Interautonomous System Support

Feature Name	Releases	Feature Information
MPLS VPN - Interautonomous System Support	12.1(5)T 12.0(16)ST 12.0(17)ST 12.0(22)S 12.0(23)S 12.2(13)T 12.0(24)S 12.2(14)S 12.0(29)S 12.2(33)SRA 12.2(33)SXH	The MPLS VPN - Interautonomous System Support feature allows an MPLS VPN to span service providers and autonomous systems. This feature module explains how to configure the Inter-AS using the ASBRs to exchange VPNv4 Addresses. In 12.1(5)T, this feature was introduced. In 12.0(16)ST, support for the Cisco 12000 series 4-Port OC-3c/STM-1c ATM line card (4-Port OC-3 ATM) and the Cisco 12000 series 4-Port OC-3c/STM-1c POS/SDH line card (4-port OC-3 POS) was added. In 12.0(17)ST, support for the Cisco 12000 series was added (See Feature Information for MPLS VPN - Interautonomous System Support, page 271 for the Cisco 12000 series line cards supported.) In 12.0(22)S, support for the Cisco 12000 series, the Cisco 10000 series edge services routers (ESRs), and the Cisco 10720 Internet routers was added. (See Feature Information for MPLS VPN - Interautonomous System Support, page 271 for the Cisco 12000 series line cards supported.) In 12.0(23)S, support was added for the Cisco 12000 series 8-port OC-3c/STM-1c ATM line card (8-Port OC-3 ATM) and the Cisco 12000 series 3-port Gigabit Ethernet line card (3-Port GbE). This feature was integrated into Cisco IOS Release 12.2(13)T. In 12.0(24)S, support was added for the Cisco 12000 series 1-port

Feature Name	Releases	Feature Information
		10-Gigabit Ethernet line card (1-Port 10-GbE) and the Cisco 12000 series modular Gigabit Ethernet/Fast Ethernet line card (modular GbE/FE) and this feature was implemented on Cisco IOS 12.0(24)S. This feature was integrated into Cisco IOS Release 12.2(14)S and implemented on Cisco 7200 and Cisco 7500 series routers. In 12.0(29)S, support was added for eBGP sessions between loopbacks of directly connected MPLS-enabled routers to provide for load sharing between neighbors. This feature was integrated into Cisco IOS Release 12.2(33)SRA. Support was added for load balancing of Virtual Private Network (VPN) traffic for VPNv4 peering. This feature was integrated into Cisco IOS Release 12.2(33)SXH.
MPLS VPN - Loadbalancing support for Inter-AS and CSC VPNs	12.0(29)S 12.2(33)SRA	This feature allows MPLS VPN Inter-AS and MPLS VPN Carrier Supporting Carrier (CSC) networks to load share traffic between adjacent LSRs that are connected by multiple links. The LSRs can be a pair of ASBRs or a CSC-PE and a CSC-CE. Using directly connected loopback peering allows load sharing at the IGP level, so more than one BGP session is not needed between the LSRs. No other label distribution mechanism is needed between the adjacent LSRs than BGP.

Feature Name	Releases	Feature Information
MPLS VPN—Multipath Support for Inter-AS VPNs	12.2(33)SRA 12.2(33)SXH	This feature supports Virtual Private Network (VPN)v4 multipath for Autonomous System Border Routers (ASBRs) in the interautonomous system (Inter-AS) Multiprotocol Label Switching (MPLS) VPN environment. It allows load balancing of VPN traffic when you use the VPNv4 peering model for Inter-AS VPNs.

Glossary

autonomous system—A collection of networks under a common administration sharing a common routing strategy.

BGP —Border Gateway Protocol. An interdomain routing protocol that exchanges network reachability information with other BGP systems (which may be within the same autonomous system or between multiple autonomous systems).

CeBGP —confederation exterior Border Gateway Protocol. A BGP between routers located within different subautonomous systems of a confederation. See *eBGP* and *iBGP* .

CE router—customer edge router. A router that is part of a customer network and that interfaces to a provider edge (PE) router. CE routers do not recognize associated MPLS VPNs.

confederation —An autonomous system divided into multiple, separate subautonomous systems and classified as a single unit.

eBGP —exterior Border Gateway Protocol. A BGP between routers located within different autonomous systems. When two routers, located in different autonomous systems, are more than one hop away from one another, the eBGP session between the two routers is considered a multihop BGP.

iBGP —interior Border Gateway Protocol. A BGP between routers within the same autonomous system.

IGP —Interior Gateway Protocol. Internet protocol used to exchange routing information within a single autonomous system. Examples of common Internet IGP protocols include IGRP, OSPF, IS-IS, and RIP.

LFIB —Label Forwarding Information Base. Data structure used in MPLS to hold information about incoming and outgoing labels and associated Forwarding Equivalence Class (FEC) packets.

MPLS —Multiprotocol Label Switching. The name of the IETF working group responsible for label switching, and the name of the label switching approach it has standardized.

NLRI —Network Layer Reachability Information. The BGP sends routing update messages containing NLRI to describe a route and how to get there. In this context, an NLRI is a prefix. A BGP update message carries one or more NLRI prefixes and the attributes of a route for the NLRI prefixes; the route attributes include a BGP next hop gateway address and extended community values.

PE router—provider edge router. A router that is part of a service provider's network. It is connected to a customer edge (CE) router and all MPLS VPN processing occurs in the PE router.

RD —route distinguisher. An 8-byte value that is concatenated with an IPv4 prefix to create a unique VPN-IPv4 prefix.

VPN —Virtual Private Network. A secure MPLS-based network that shares resources on one or more physical networks (typically implemented by one or more service providers). A VPN contains geographically dispersed sites that can communicate securely over a shared backbone network.

VRF —VPN routing and forwarding instance. Routing information that defines a Virtual Private Network (VPN) site that is attached to a provider edge (PE) router. A VRF 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 what goes into the forwarding table.

MPLS VPN--SNMP Notifications

This document describes the Simple Network Management Protocol (SNMP) agent support in Cisco IOS for Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) event notifications as implemented in the notifications section of the draft *MPLS/BGP Virtual Private Network Management Information Base Using SMIv2* (*draft-ietf-ppvpn-mpls-vpn-mib-03.txt*).

The MPLS VPN technology allows service providers to offer intranet and extranet VPN services that directly connect their customers' remote offices to a public network with the same security and service levels that a private network offers. The Provider-Provisioned VPN (PPVPN)-MPLS-VPN MIB notifications provide SNMP notification for critical MPLS VPN events.

The MPLS VPN--SNMP Notifications feature provides the following benefits:

- A standards-based SNMP interface for retrieving information about critical MPLS VPN events.
- The generation and queuing of notifications that call attention to major changes in the operational status of MPLS VPN enabled interfaces; the forwarding of notification messages to a designated NMS for evaluation and action by network administrators.
- Advanced warning when VPN routing tables are approaching or exceed their capacity.
- Warnings about the reception of illegal labels on a VRF enabled interface. Such receptions may indicate misconfiguration or an attempt to violate security.

Feature Specifications for the MPLS VPN--SNMP Notifications

Feature History	
Release	Modification
12.0(21)ST	This feature was introduced.
12.0(22)S	This feature was integrated into Cisco IOS Release 12.0(22)S.
12.2(13)T	This feature was integrated into Cisco IOS Release 12.2(13)T.
Supported Platforms	
Cisco IOS 12.0 S and ST Releases: Cisco 7500 series, Cisco 12000 series. Cisco IOS 12.2 T Releases: Cisco 3620, Cisco 3640, Cisco 7200 series, Cisco 7500 series, Cisco MGX 8850-RPM.	

Note In Cisco IOS Releases 12.0(21)ST and 12.0(22)S, the PPVPN MPLS-VPN-MIB notifications are described in the *MPLS VPN--SNMP MIB Support* feature module.

- Finding Feature Information, page 278
- Prerequisites for MPLS VPN--SNMP Notifications , page 278
- Restrictions for MPLS VPN--SNMP Notifications , page 278
- Information About MPLS VPN--SNMP Notifications , page 278
- How to Configure the MPLS VPN--SNMP Notifications, page 282
- Configuration Examples for MPLS VPN--SNMP Notifications, page 287
- Additional References, page 288
- Command Reference, page 289
- Glossary, page 290

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for MPLS VPN--SNMP Notifications

The MPLS VPN--SNMP Notifications feature requires the following:

- SNMP is installed and enabled on the label switching routers.
- MPLS is enabled on the label switching routers.
- Multiprotocol BGP is enabled on the label switching routers.
- Cisco Express Forwarding is enabled on the label switching routers.

Restrictions for MPLS VPN--SNMP Notifications

- The MPLS-VPN-MIB agent is not implemented in this release.
- Configuration of the MIB using the SNMP SET command is not supported in this release.
- The retrieval of MPLS-VPN-MIB objects using SNMP GET is not supported in this release.

Information About MPLS VPN--SNMP Notifications

- Cisco Implementation of MPLS-VPN-MIB, page 279
- Notification Specification for MPLS-VPN-MIB, page 281

- Monitoring the MPLS VPN--SNMP Notifications, page 282

Cisco Implementation of MPLS-VPN-MIB

SNMP agent code operating in conjunction with the notifications of the MPLS VPN--SNMP Notifications feature enables a standardized, SNMP-based approach to monitoring the MPLS-VPN-MIB notifications that aid in the management of MPLS VPNs in Cisco IOS.

The MPLS VPN--SNMP Notifications feature is based on the IETF draft specification *draft-ietf-ppvpn-mpls-vpn-mib-02.txt* , which includes notification objects that support MPLS VPN notification events. This IETF draft MIB, which undergoes revisions from time to time, is being evolved toward becoming a standard. Accordingly, the Cisco implementation of features of the MPLS-VPN-MIB is expected to track the evolution of the IETF draft MIB, and may change accordingly.

Some slight differences between the IETF draft MIB and the actual implementation of MPLS VPNs within Cisco IOS require some minor translations between the MPLS-VPN-MIB and the internal data structures of Cisco IOS. These translations are accomplished by means of the SNMP agent code. Also, while running as a low priority process, the SNMP agent provides a management interface to Cisco IOS. SNMP adds little overhead on the normal functions of the device.

The SNMP objects defined in the MPLS-VPN-MIB notifications can be viewed by any standard SNMP utility. The network administrator can retrieve information in the MPLS-VPN-MIB using standard SNMP **get** and **getnext** operations for SNMP v1, v2, and v3.

All MPLS-VPN-MIB objects are based on the IETF draft MIB; thus, no specific Cisco SNMP application is required to support the functions and operations pertaining to the MPLS VPN--SNMP Notifications feature.

This section contains the following information about the Cisco implementation of the MPLS-VPN-MIB:

- Capabilities Supported by MPLS VPN--SNMP Notifications, page 279
- Notification Generation Events for the MPLS-VPN-MIB, page 279

Capabilities Supported by MPLS VPN--SNMP Notifications

The following functionality is supported in this release for the MPLS VPN--SNMP Notifications feature. This feature provides you with the ability to do the following:

- Create and send notification messages that signal changes when critical MPLS VPN events occur.
- Enable, disable, and configure notification messages for MPLS VPN events by using extensions to existing SNMP CLI commands.
- Specify the IP address of a network management system (NMS) in the operating environment to which notification messages are sent.
- Write notification configurations into nonvolatile memory.

Notification Generation Events for the MPLS-VPN-MIB

The following notifications of the MPLS-VPN-MIB are implemented for this release:

- **mplsVrfIfUp** --Sent to an NMS when an interface comes up and is assigned a VPN routing/forwarding table instance (VRF).
- **mplsVrfIfDown** --Generated and sent to the NMS when a VRF is removed from an interface or the interface transitions from an operationally “up” state to a “down” state.

Note For the mplsVrfIfUp or mplsVrfIfDown notifications to be issued on ATM or Frame Relay subinterfaces, you must configure the **snmp-server traps atm subif** command or the **snmp-server traps frame-relay subif** command on the subinterfaces, respectively.

- **mplsNumVrfRouteMidThreshExceeded** --Generated and sent when the middle (warning) threshold is crossed. You can configure this threshold in the CLI by using the following commands:

```
Router(config)# ip vrf vrf-name
Router(config-vrf)# maximum routes max-thresh
mid-thresh (% of max)
```

This notification is sent to the NMS only at the time the threshold is exceeded. Whenever the number of routes falls below this threshold and exceeds the threshold again, a notification is sent to the NMS. (See the figure below for a comparison of the warning and maximum thresholds.)

- **mplsNumVrfRouteMaxThreshExceeded** --Generated and sent when you attempt to create a route on a VRF that already contains the maximum number of routes as defined by the following CLI commands:

```
Router(config)# ip vrf vrf-name
Router(config-vrf)# maximum routes max-thresh
mid-thresh (% of max)
```

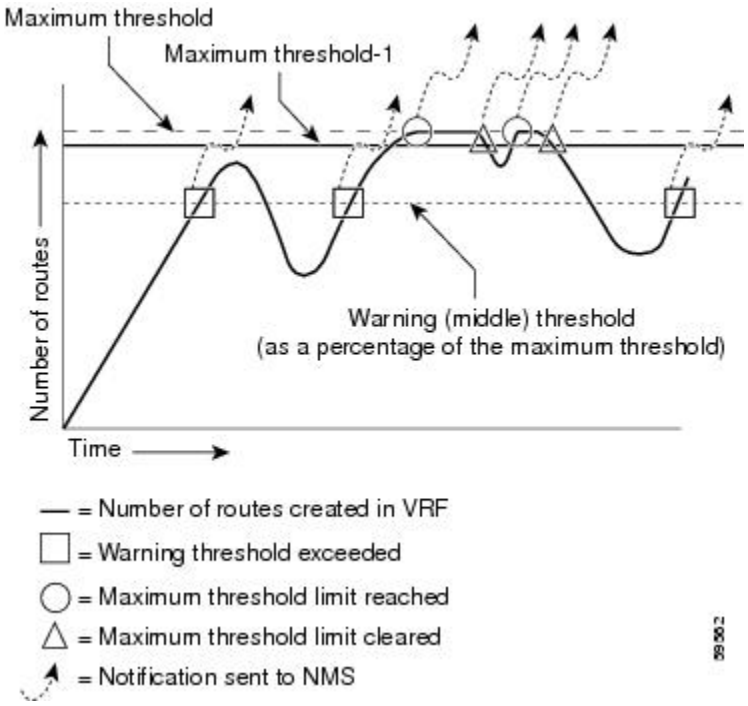
A trap notification is sent to the NMS when you attempt to exceed the maximum threshold. Another notification is not sent until the number of routes falls below the maximum threshold and reaches the maximum threshold again. (See the figure below for an example of how this notification works and for a comparison of the maximum and warning thresholds.)

Note The **maximum routes** command sets the number of routes for a VRF. You *cannot* exceed the number of routes in the VRF that you set with the **maximum routes** *max-thresh* CLI command. Prior to this implementation of the MPLS-VPN-MIB, you were not notified when this threshold (or the warning threshold) was reached.

- **mplsNumVrfSecIllegalLabelThreshExceeded** --Generated and sent when the amount of illegal labels received on a VRF interface exceeds the threshold *mplsVpnVrfSecIllegalLabelRcvThresh* . This threshold is defined with a value of 0. Therefore, a notification is sent when the first illegal label is received on a VRF. Labels are considered illegal if they are outside of the valid label range, do not

have a Label Forwarding Information Base (LFIB) entry, or the table ID of the message does not match the table ID for the label in the LFIB.

Figure 24 Comparison of Warning and Maximum Thresholds



For information on the Cisco IOS CLI commands for configuring MPLS-VPN-MIB notifications that are to be sent to an NMS, see the How to Configure the MPLS VPN--SNMP Notifications, page 282 and Command Reference, page 289 sections.

Notification Specification for MPLS-VPN-MIB

In an SNMPv1 notification, each VPN notification has a generic type identifier and an enterprise-specific type identifier for identifying the notification type.

- The generic type for all VPN notifications is “enterpriseSpecific” as this is not one of the generic notification types defined for SNMP.
- The enterprise-specific type is identified as follows:
  - 1 for *mplsVrfIfUp*
  - 2 for *mplsVrfIfDown*
  - 3 for *mplsNumVrfRouteMidThreshExceeded*
  - 4 for *mplsNumVrfRouteMaxThreshExceeded*
  - 5 for *mplsNumVrfSecIllegalLabelThreshExceeded*

In SNMPv2, the notification type is identified by an **SnmpTrapOID** varbind (variable binding consisting of an object identifier (OID) type and value) included within the notification message.

Each notification also contains two additional objects from the MPLS-VPN-MIB. These objects provide additional information about the event, as follows:

- The VRF interface up/down notifications provide additional variables--*mplsVpnInterfaceConfIndex* and *mplsVpnVrfName*-- in the notification. These variables describe the SNMP interface index and the VRF name, respectively.
- The mid and max threshold notifications include the *mplsVpnVrfName* variable (VRF name) as well as the *mplsVpnVrfPerfCurrNumRoutes* variable that indicates the current number of routes within the VRF.
- The illegal label notification includes the *mplsVpnVrfName* variable (VRF name) and the *mplsVpnVrfSecIllegalLabelViolations* variable that maintains the current count of illegal labels on a VPN.

Monitoring the MPLS VPN--SNMP Notifications

When MPLS-VPN-MIB notifications are enabled, notification messages relating to specific MPLS VPN events within Cisco IOS are generated and sent to a specified NMS in the network. Any utility that supports SNMPv1 or SNMPv2 notifications can receive notification messages.

To monitor MPLS-VPN-MIB notification messages, log in to an NMS that supports a utility that displays SNMP notifications, and start the display utility.

How to Configure the MPLS VPN--SNMP Notifications

This section contains the following procedures. Each task in the list is identified as either required or optional.

The MPLS VPN notifications are enabled or disabled using the extended CLI commands (see the Command Reference, page 289 section).

- Configuring an SNMP Communtity, page 282
- Configure the Router to Send SNMP Traps, page 283
- Configure Threshold Values for MPLS VPN--SNMP Notifications, page 286

Configuring an SNMP Communtity

An SNMP community string defines the relationship between the SNMP manager and the agent. The community string acts like a password to regulate access to the agent on the router.

Perform this task to configure an SNMP community.

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **configure terminal**
4. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*acl-number*]
5. **do copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** **Example:** `Router# show running-config`	Displays the running configuration to determine if an SNMP agent is already running. If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify the information or change it as needed.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*acl-number*] **Example:** `Router(config)# snmp-server community comaccess ro`	Sets up the community access string to permit access to the Simple Network Management Protocol protocol. • The *string* argument acts like a password and permits access to the SNMP protocol. • The **view***view-name* keyword argument pair specifies the name of a previously defined view. The view defines the objects available to the community. • The **ro** keyword specifies read-only access. Authorized management stations are only able to retrieve MIB objects. • The **rw** keyword specifies read-write access. Authorized management stations are able to both retrieve and modify MIB objects. • The *acl-number* argument is an integer from 1 to 99 that specifies an access list of IP addresses that are allowed to use the community string to gain access to the SNMP agent.
Step 5	**do copy running-config startup-config** **Example:** `Router(config)# do copy running-config startup-config`	Saves the modified configuration to nonvolatile memory (NVRAM) as the startup configuration file. (The **do** command allows you to perform Exec level commands in configuration mode.)

Configure the Router to Send SNMP Traps

Perform this task to configure the router to send traps to a host.

The **snmp-server host** command specifies which hosts receive traps. The **snmp-server enable traps** command globally enables the trap production mechanism for the specified traps.

Note Although you can set the *community-string* argument using the **snmp-server host** command by itself, we recommend you define this string using the **snmp-server community** command prior to using the **snmp-server host** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** *host-addr* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]
4. Do one of the following:
   - **snmp-server enable traps atm** [**pvc** | **subif**]
   - or
   - **snmp-server enable traps frame-relay** [**subif**]
5. **snmp-server enable traps mpls vpn**
6. **end**

DETAILED STEPS

	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 **snmp-server host** *host-addr* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] **Example:** `Router(config)# snmp-server host 172.20.2.160 traps comaccess mpls-vpn`	Specifies the recipient of an SNMP notification operation. • The *host-addr* argument specifies the name or Internet address of the host (the targeted recipient). • The **traps** keyword sends SNMP traps to this host. This is the default. • The **informs** keyword sends SNMP informs to this host. • The **version** keyword specifies the version of the SNMP used to send the traps. Version 3 is the most secure model, as it allows packet encryption with the **priv** keyword. If you use the **version** keyword, you must specify one of the following: ◦ **1**--SNMPv1. This option is not available with informs. ◦ **2c** --SNMPv2C. ◦ **3**--SNMPv3. The following three optional keywords can follow the **version 3** keyword (**auth**, **noauth**, **priv**). • The *community-string* argument is a password-like community string sent with the notification operation. • The **udp-port** *port* keyword argument pair names the UDP port of the host to use. The default is 162. • The *notification-type* argument specifies the type of notification to be sent to the host. If no type is specified, all notifications are sent. MPLS VPN notifications are specified with the **mpls-vpn** keyword.

Step 4 Do one of the following:
• **snmp-server enable traps atm** [**pvc** | **subif**]
• or
• **snmp-server enable traps frame-relay** [**subif**]
 	
Example:
 	
`Router(config)# snmp-server enable traps atm subif`
 	
Example:
 	
`Router(config)# snmp-server enable traps frame-relay subif`
	(For ATM subinterfaces only) Enables the sending of ATM SNMP notifications.
	• The **pvc** keyword enables SNMP ATM permanent virtual circuit (PVC) traps.
	• The **subif** keyword enables SNMP ATM subinterface traps.
	or
	(For Frame Relay subinterfaces only) Enables Frame Relay DLCI link status SNMP notifications.
	• The **subif** keyword enables SNMP Frame Relay subinterface traps.
	Note For mplsVrfIfUp or mplsVrfIfDown notifications to be issued on ATM or Frame Relay subinterfaces, you must configure the appropriate **snmp-server enable traps** command with the **subif** keyword.

	Command or Action	Purpose
Step 5	**snmp-server enable traps mpls vpn** **Example:** `Router(config)# snmp-server enable traps mpls vpn vrf-up vrf-down`	Enables the router to send MPLS VPN SNMP notifications.
Step 6	**end** **Example:** `Router(config)# end`	(Optional) Exits to user EXEC mode.

Configure Threshold Values for MPLS VPN--SNMP Notifications

Perform this task to configure threshold values for MPLS VPN SNMP notifications.

The **mplsNumVrfRouteMidThreshExceeded** notification event is generated and sent when the middle (warning) threshold is crossed. You can configure this threshold in the CLI by using the **maximum routes** command in VRF configuration mode. This notification is sent to the NMS only at the time the threshold is exceeded. Whenever the number of routes falls below this threshold and exceeds the threshold again, a notification is sent to the NMS.

The **mplsNumVrfRouteMaxThreshExceeded** notification event is generated and sent when you attempt to create a route on a VRF that already contains the maximum number of routes as defined by the **maximum routes** command in VRF configuration mode. A trap notification is sent to the NMS when you attempt to exceed the maximum threshold. Another notification is not sent until the number of routes falls below the maximum threshold and reaches the maximum threshold again.

(See the figure above for an example of how this notification works and for a comparison of the maximum and warning thresholds.)

Note The **maximum routes** command sets the number of routes for a VRF. You *cannot* exceed the number of routes in the VRF that you set with the **maximum routes** *max-thresh* CLI command. Prior to this implementation of the MPLS-VPN-MIB, you were not notified when this threshold (or the warning threshold) was reached.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **maximum routes** *limit* {*warn-threshold* | **warning-only**}
5. **end**

DETAILED STEPS

	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 vrf** *vrf-name* **Example:** `Router(config)# ip vrf vpn1`	Configures a VRF routing table. • The *vrf-name* argument specifies the name assigned to a VRF.
Step 4	**maximum routes** *limit* {*warn-threshold* | **warning-only**} **Example:** `Router(config-vrf)# maximum routes 10000 80`	Limits the maximum number of routes in a VRF to prevent a PE router from importing too many routes. • The *limit* argument specifies the maximum number of routes allowed in a VRF. You may select from 1 to 4,294,967,295 routes to be allowed in a VRF. • The *warn-threshold* argument specifies when the threshold limit is reached and routes are rejected. The threshold limit is a percentage of the *limit* specified, from 1 to 100 percent. • The **warning-only** keyword specifies that a SYSLOG error message is issued when the maximum number of routes allowed for a VRF exceeds the threshold. However, additional routes are still allowed.
Step 5	**end** **Example:** `Router(config-vrf)# end`	(Optional) Exits to privileged EXEC mode.

Configuration Examples for MPLS VPN--SNMP Notifications

- Configure the Community Example, page 288
- Configure the Router to Send SNMP Traps Examples, page 288
- Configure Threshold Values for Examples, page 288

Configure the Community Example

The following example shows enabling a simple SNMP community group. This configuration permits any SNMP client to access all MPLS-VPN-MIB objects with read-only access using the community string comaccess.

```
Router# configure terminal
Router(config)# snmp-server community comaccess ro
```

Verify that the SNMP master agent is enabled for the MPLS VPN--SNMP Notifications feature:

```
Router# show running-config | include snmp-server
Building configuration...
....
snmp-server community comaccess RO
....
```

Note If you do not see any “snmp-server” statements, SNMP has not been enabled on the router.

Configure the Router to Send SNMP Traps Examples

The following example shows you how to enable the router to send MPLS VPN notifications to host 172.20.2.160 using the comaccess community string if a VRF transitions from a down state to an up state or from an up state to a down state.

```
Router# configure terminal
Router(config)# snmp-server host 172.20.2.160 traps comaccess mpls-vpn
Router(config)# snmp-server enable traps mpls vpn vrf-up vrf-down
```

Configure Threshold Values for Examples

The following example shows how to set a maximum threshold or 10000 routes and a warning threshold that is 80 percent of the maximum threshold for a VRF named vpn1 on a router:

```
Router(config)# ip vrf vpn1
Router(config)# maximum routes 10000 80
```

Additional References

Related Documents

Related Topic	Document Title
MPLS Virtual Private Network (VPN) configuration tasks	MPLS Virtual Private Networks

Standards

Standards	Title
No new or modified standards are supported by this feature.	--

MIBs.

MIBs	MIBs Link
MPLS/BGP Virtual Private Network Management Information Base Using SMIv2 (*draft-ietf-ppvpn-mpls-vpn-mib-03.txt*) MPLS-VPN-MIB.my	To obtain lists of supported MIBs by platform and Cisco software release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 2233	*The Interfaces Group MIB using SMIv2*

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/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Multiprotocol Label Switching Command Reference* at http://www.cisco.com/en/US/docs/ios/mpls/command/reference/mp_book.html. For information about all Cisco IOS commands, go to the *Cisco IOS Master Commands List* .

- **snmp-server enable traps mpls vpn**
- **snmp-server host**

Glossary

ASN.1 --Abstract Syntax Notation One. OSI language for describing data types independent of particular computer structures and representation techniques. Described by ISO International Standard 8824.

BGP --Border Gateway Protocol. The exterior Border Gateway Protocol used to exchange routing information between routers in separate autonomous systems. BGP uses Transmission Control Protocol (TCP). Because TCP is a reliable protocol, BGP does not experience problems with dropped or fragmented data packets.

CEF --Cisco Express Forwarding. An advanced Layer 3 IP switching technology. CEF optimizes network performance and scalability for networks with large and dynamic traffic patterns.

CE router --customer edge router. A router on the border between a VPN provider and a VPN customer that belongs to the customer.

community --In SNMP, a logical group of managed devices and NMSs in the same administrative domain.

community name --*See* community string.

community string --Text string that acts as a password and is used to authenticate messages sent between a managed station and a router containing an SNMP agent. The community string is sent in every packet between the manager and the client. Also called a community name.

IETF --Internet Engineering Task Force. Task force consisting of over 80 working groups responsible for developing Internet standards. The IETF operates under the auspices of ISOC. *See also* ISOC.

informs --A type of notification message that is more reliable than a conventional trap notification message, because the informs message notification requires acknowledgment, and a trap notification does not.

ISOC --Internet Society. International nonprofit organization, founded in 1992, that coordinates the evolution and use of the Internet. In addition, ISOC delegates authority to other groups related to the Internet, such as the IAB. ISOC is headquartered in Reston, Virginia (United States).

label --A short, fixed-length data construct that tells switching nodes how to forward data (packets or cells).

label distribution protocol --*See* LDP.

label forwarding information base --*See* LFIB.

label switch router --*See* LSR.

LDP --label distribution protocol. A standard protocol between MPLS-enabled routers that is used for the negotiation of the labels (addresses) used to forward packets.

LFIB --label forwarding information base. In the Cisco Label Switching system, the data structure for storing information about incoming and outgoing tags (labels) and associated equivalent packets suitable for labeling.

LSR --label switch router. A device that forwards MPLS packets based on the value of a fixed-length label encapsulated in each packet.

MIB --Management Information Base. Database of network management information that is used and maintained by a network management protocol such as SNMP or CMIP. The value of a MIB object can be changed or retrieved using SNMP or CMIP commands, usually through a GUI network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

MPLS --Multiprotocol Label Switching. A method for forwarding packets (frames) through a network. It enables routers at the edge of a network to apply labels to packets (frames). ATM switches or existing routers in the network core can switch packets according to the labels with minimal lookup overhead.

MPLS interface --An interface on which MPLS traffic is enabled.

MPLS VPN --Multiprotocol Label Switching Virtual Private Network. Using MPLS VPNs in a Cisco IOS network provide the capability to deploy and administer scalable Layer 3 VPN backbone services including applications, data hosting network commerce, and telephony services, to business customers. A VPN is a secure IP-based network that shares resources on one or more physical networks. A VPN contains geographically dispersed sites that can communicate securely over a shared backbone.

For an MPLS VPN Solution, an MPLS VPN is a set of PEs that are connected by means of a common "backbone" network to supply private IP interconnectivity between two or more customer sites for a given customer. Each VPN has a set of provisioning templates and policies and can span multiple provider administrative domains (PADs).

Multiprotocol Label Switching --*See* MPLS.

notification --A message sent by an SNMP agent to a network management station, console, or terminal to indicate that a significant event within Cisco IOS has occurred. *See also* trap.

NMS --network management system. A powerful, well-equipped computer (typically an engineering workstation) that is used by a network administrator to communicate with other devices in the network. An NMS is typically used to manage network resources, gather statistics, and perform a variety of network administration and configuration tasks.

PE router --provider edge router. A router on the border between a VPN provider and a VPN customer that belongs to the provider.

PPVPN --Provider-Provisioned VPN. The name of the IETF working group that is developing the PPVPN-MPLS-VPN MIB (MPLS-VPN-MIB).

QoS --quality of service. Measure of performance for a transmission system that reflects its transmission quality and service availability.

RSVP --Resource Reservation Protocol. Protocol for reserving network resources to provide Quality of Service guarantees to application flows.

Simple Network Management Protocol --*See* SNMP.

SNMP --Simple Network Management Protocol. Network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices, and to manage configurations, statistics collection, performance, and security. See *also* SNMP2.

SNMP2 --SNMP Version 2. Version 2 of the popular network management protocol. SNMP2 supports centralized as well as distributed network management strategies, and includes improvements in the SMI, protocol operations, management architecture, and security. *See also* SNMP.

traffic engineering --The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

trap --A message sent by an SNMP agent to a network management station, console, or terminal, indicating that a significant event occurred. Traps (notifications) are less reliable than inform requests, because the receiver does not send an acknowledgment when it receives a trap. The sender cannot determine if the trap was received. *See also* notification.

VPN --Virtual Private Network. A group of sites that, as the result of a set of administrative policies, are able to communicate with each other over a shared backbone network. *See* MPLS VPN.

VRF --VPN routing/forwarding instance. A VRF 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 what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

Multi-VRF Selection Using Policy-Based Routing (PBR)

The Multi-VRF Selection Using Policy-Based Routing (PBR) feature allows a specified interface on a provider edge (PE) router to route packets to Virtual Private Networks (VPNs) based on packet length or match criteria defined in an IP access list.

You can enable VPN routing and forwarding (VRF) selection by policy routing packets through a route map, through the global routing table, or to a specified VRF.

You can enable policy-routing packets for VRF instances by using route map commands with **set** commands.

This feature and the Directing MPLS VPN Traffic Using a Source IP Address feature can be configured together on the same interface.

- Finding Feature Information, page 293
- Prerequisites for Multi-VRF Selection Using Policy-Based Routing, page 294
- Restrictions for Multi-VRF Selection Using Policy-Based Routing, page 294
- Information About Multi-VRF Selection Using Policy-Based Routing, page 294
- How to Configure Multi-VRF Selection Using Policy-Based Routing, page 297
- Configuration Examples for Multi-VRF Selection Using Policy-Based Routing, page 305
- Additional References, page 306
- Feature Information for Multi-VRF Selection Using Policy-Based Routing, page 307
- Glossary, page 308

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Multi-VRF Selection Using Policy-Based Routing

- The router must support policy-based routing (PBR) in order for you to configure this feature. For platforms that do not support PBR, use the Directing MPLS VPN Traffic Using a Source IP Address feature.
- A VRF must be defined before you configure this feature. An error message is displayed on the console if no VRF exists.

Restrictions for Multi-VRF Selection Using Policy-Based Routing

- All commands that aid in routing also support hardware switching, except for the **set ip next-hop verify availability** command because Cisco Discovery Protocol information is not available in the line cards.
- Protocol Independent Multicast (PIM) and multicast packets do not support PBR and cannot be configured for a source IP address that is a match criterion for this feature.
- The **set vrf** and **set ip global next-hop** commands can be configured with the **set default interface**, **set interface** , **set ip default next-hop**, and **set ip next-hop** commands. But the **set vrf** and **set ip global next-hop** commands take precedence over the **set default interface**, **set interface** , **set ip default next-hop**, and **set ip next-hop** commands. No error message is displayed if you attempt to configure the **set vrf** command with any of these three **set** commands.
- The Multi-VRF Selection Using Policy-Based Routing (PBR) feature cannot be configured with IP prefix lists.
- The **set global** and **set vrf** commands cannot be simultaneously applied to a route map.
- The Multi-VRF Selection Using Policy-Based Routing (PBR) feature supports VRF-lite; that is, only IP routing protocols run on the router. Multiprotocol Label Switching (MPLS) and VPN cannot be configured.

Information About Multi-VRF Selection Using Policy-Based Routing

- Policy Routing of VPN Traffic Based on Match Criteria, page 294
- Policy-Based Routing set Commands, page 295

Policy Routing of VPN Traffic Based on Match Criteria

The Multi-VRF Selection Using Policy-Based Routing feature is an extension of the VRF Selection Based on Source IP Address feature. The PBR implementation of the VRF selection feature allows you to policy route VPN traffic based on match criteria. Match criteria are defined in an IP access list and/or are based on packet length. The following match criteria are supported in Ciscosoftware:

- IP access lists—Define match criteria based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. All IP access list configuration options in Cisco software can be used to define match criteria.
- Packet lengths—Define match criteria based on the length of a packet, in bytes. The packet length filter is defined in a route map with the **match length** route-map configuration command.

Policy routing is defined in the route map. The route map is applied to the incoming interface with the **ip policy route-map** interface configuration command. An IP access list is applied to the route map with the **match ip address** route-map configuration command. Packet length match criteria are applied to the route map with the **match length** route-map configuration command. The **set** action is defined with the **set vrf** route-map configuration command. The match criteria are evaluated, and the appropriate VRF is selected by the **set** command. This combination allows you to define match criteria for incoming VPN traffic and policy route VPN packets out to the appropriate VRF.

Policy-Based Routing set Commands

- Policy-routing Packets for VRF Instances, page 295
- Change of Normal Routing and Forwarding Behavior, page 296
- Support of Inherit-VRF Inter-VRF and VRF-to-Global Routing, page 296

Policy-routing Packets for VRF Instances

To enable policy-routing packets for VRF instances, you can use route map commands with the following **set** commands. They are listed in the order in which the router uses them during the routing of packets.

- **set tos**--Sets the Type of Service (TOS) bits in the header of an IP packet.
- **set df**--Sets the Don't Fragment (DF) bit in the header of an IP packet.
- **set vrf**--Routes packets through the specified interface. The destination interface can belong only to a VRF instance.
- **set global**--Routes packets through the global routing table. This command is useful for routing ingress packets belonging to a specific VRF through the global routing table.
- **set ip vrf next-hop**--Indicates where to output packets that pass a match criteria of a route map for policy routing when the next hop must be under a specified VRF.
- **set ip global next-hop**--Indicates where to forward packets that pass a match criterion of a route map for policy routing and for which the Cisco IOS software uses the global routing table.
- **set interface**--When packets enter a VRF, routes the packets out of the egress interface under the same VRF according to the set interface policy, provided that the Layer 2 rewrite information is available.
- **set ip default vrf**--Provides inherit-VRF and inter-VRF routing. With inherit-VRF routing, packets arriving at a VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, packets arriving at a VRF interface are routed via any other outgoing VRF interface.
- **set ip default global**--Provides VRF to global routing.
- **set default interface**--Indicates where to output packets that pass a match criterion of a route map for policy routing and have no explicit route to the destination. The interface can belong to any VRF.
- **set ip global next-hop**--Routes packets through the global routing table, where the next hop lookup will be in the global routing table.
- **set ip default next-hop**--Indicates where to output packets that pass a match criterion of a route map for policy routing and for which the Cisco IOS software has no explicit route to a destination.

Change of Normal Routing and Forwarding Behavior

When you configure PBR, you can use the following four **set** commands to change normal routing and forwarding behavior. Configuring any of these **set** commands, with the potential exception of the **set ip next-hop** command, overrides the routing behavior of packets entering the interface if the packets do not belong to a VRF. The packets are routed from the egress interface across the global routing table.

- **set default interface**--Indicates where to output packets that pass a match criterion of a route map for policy routing and have no explicit route to the destination.
- **set interface**--When packets enter a VRF, routes the packets out of the egress interface under the same VRF according to the set interface policy, provided that the Layer 2 rewrite information is available.
- **set ip default next-hop**--Indicates where to output packets that pass a match criterion of a route map for policy routing and for which the Cisco IOS software has no explicit route to a destination.
- **set ip next-hop**--Indicates where to output packets that pass a match criterion of a route map for policy routing. If a packet is received on a VRF interface and is transmitted from another interface within the same VPN, the VRF context of the incoming packet will be inherited from the interface.

Support of Inherit-VRF Inter-VRF and VRF-to-Global Routing

The Multi-VRF Selection Using Policy-Based Routing (PBR) feature supports inherit-VRF and inter-VRF. With inherit-VRF routing, packets arriving at a VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, packets arriving at a VRF interface are routed via any other outgoing VRF interface.

VRF-to-global routing causes packets that enter any VRF interface to be routed via the global routing table. When a packet arrives on a VRF interface, the destination lookup normally is done only in the corresponding VRF table. If a packet arrives on a global interface, the destination lookup is done in the global routing table.

The Multi-VRF Selection Using Policy-Based Routing (PBR) feature modifies the following **set** commands to support inherit-VRF, inter-VRF, and VRF-to-global routing. The commands are listed in the order in which the router uses them during the routing of packets.

- **set global**—Routes packets through the global routing table. This command is useful for routing ingress packets belonging to a specific VRF through the global routing table.
- **set ip global next-hop**—Indicates where to forward packets that pass a match criterion of a route map for policy routing and for which the Cisco software uses the global routing table.
- **set ip vrf next-hop**—Causes the router to look up the next hop in the VRF table. If a packet arrives on an interface that belongs to a VRF and the packet needs to be routed via a different VRF, you can use the **set ip vrf next-hop** command.
- **set ip default vrf**—Provides inherit-VRF and inter-VRF routing. With inherit-VRF routing, packets arriving at a VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, packets arriving at a VRF interface are routed via any other outgoing VRF interface.
- **set interface**—When packets enter a VRF, routes the packets out of the egress interface under the same VRF, according to the set interface policy, provided that the Layer 2 rewrite information is available.
- **set default interface**—Indicates where to output packets that pass a match criterion of a route map for policy routing and have no explicit route to the destination. The interface can belong to any VRF.
- **set ip next-hop**—Routes packets through the global routing table in an IP-to-IP routing and forwarding environment.
- **set vrf**—Selects the appropriate VRF after a successful match occurs in the route map. VRS-aware PSV allows only inter-VRF (or VRF-to-VRF) switching.

How to Configure Multi-VRF Selection Using Policy-Based Routing

- Defining the Match Criteria for Multi-VRF Selection Using PBR, page 297
- Configuring Multi-VRF Selection in a Route Map, page 299
- Configuring Multi-VRF Selection Using PBR and IP VRF Receive on the Interface, page 302
- Verifying the Configuration of Multi-VRF Selection Using PBR, page 303

Defining the Match Criteria for Multi-VRF Selection Using PBR

Define the match criteria for multi-VRF selection using PBR so that you can selectively route the packets instead of using their default routing and forwarding.

The match criteria for multi-VRF selection using PBR are defined in an access list. Standard, named, and extended access lists are supported.

You can define the match criteria based on the packet length by configuring the **match length** route-map configuration command. This configuration option is defined entirely within a route map.

The following sections explain how to configure PBR route selection:

- Configuring Multi-VRF Selection Using PBR with a Standard Access List, page 297
- Configuring Multi-VRF Selection Using PBR with a Named Extended Access List, page 298

Configuring Multi-VRF Selection Using PBR with a Standard Access List

The tasks in the following sections assume that the VRF and associated IP address are already defined.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} [**source** *source-wildcard*] [**log**]

DETAILED STEPS

	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	**access-list** *access-list-number* {**deny** | **permit**} [**source** *source-wildcard*] [**log**] **Example:** `Router(config)# access-list 40 permit source 10.1.1.0/24 0.0.0.255`	Creates an access list and defines the match criteria for the route map. • Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. You can use all IP access list configuration options to define match criteria. • The example creates a standard access list numbered 40. This filter permits traffic from any host with an IP address in the 10.1.1.0/24 subnet.

Configuring Multi-VRF Selection Using PBR with a Named Extended Access List

To configure Multi-VRF Selection using PBR with a named extended access list, complete the following steps.

The tasks in the following sections assume that the VRF and associated IP address are already defined.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*]
4. [*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos***tos*] [**ttl** *operator-vaue*] [**log**] [**time-range** *time-range-name*] [**fragments**]

DETAILED STEPS

	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	**ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] **Example:** `Router(config)# ip access-list extended NAMEDACL`	Specifies the IP access list type and enters the corresponding access list configuration mode. • You can specify a standard, extended, or named access list.
Step 4	[*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos***tos*] [**ttl** *operator-vaue*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config-ext-nacl)# permit ip any any option any-options`	Defines the criteria for which the access list will permit or deny packets. • Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. You can use all IP access list configuration options to define match criteria. • The example creates a named access list that permits any configured IP option.

Configuring Multi-VRF Selection in a Route Map

Incoming packets are filtered through the match criteria that are defined in the route map. After a successful match occurs, the **set vrf** command configuration determines the VRF through which the outbound VPN packets will be policy routed.

You must define the VRF before you configure the route map; otherwise an error message appears on the console.

A receive entry must be added to the VRF selection table with the **ip vrf receive** command. If a match and set operation occurs in the route map but there is no receive entry in the local VRF table, the packet will be dropped if the packet destination is local.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. Do one of the following:
   - **set ip vrf** *vrf-name* **next-hop** *ip-address* [*...ip-address*]
   - or
   - **set ip next-hop recursive vrf** *ip-address* [*...ip-address*]
   - or
   - **set ip global next-hop** *ip-address* [*...ip-address*]
5. Do one of the following:
   - **match ip address** {*acl-number* [*acl-name* | *acl-number*]}
   - or
   - **match length** *minimum-lengthmaximum-length*
6. **end**

DETAILED STEPS

	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	**route-map** *map-tag* [**permit**	**deny**] [*sequence-number*] **Example:** `Router(config)# route-map map1 permit 10`	Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing. • Enters route-map configuration mode.

Command or Action	Purpose

Step 4 Do one of the following:
• **set ip vrf** *vrf-name* **next-hop** *ip-address* [*...ip-address*]
• or
• **set ip next-hop recursive vrf** *ip-address* [*...ip-address*]
• or
• **set ip global next-hop** *ip-address* [*...ip-address*]
Example:
`Router(config-route-map)# set ip vrf myvrf next-hop 10.0.0.0`
Example:
`Router(config-route-map)# set ip next-hop recursive vrf 10.0.0.0`
Example:
`Router(config-route-map)# set ip global next-hop 10.0.0.0`
	Indicates where to forward packets that pass a match criterion of a route map for policy routing when the next hop must be under a specified VRF.
	or
	Indicates which destination or next hop will be used for packets that pass the match criterion configured in the route map.
	or
	Indicates where to forward packets that pass a match criterion of a route map for policy routing and for which the software uses the global routing table.

Step 5 Do one of the following:
• **match ip address** {*acl-number* [*acl-name* | *acl-number*]}
• or
• **match length** *minimum-lengthmaximum-length*
Example:
`Router(config-route-map)# match ip address 1 or`
Example:
`Router(config-route-map)# match length 3 200`
	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets. IP access lists are supported.
	• The example configures the route map to use standard access list 1 to define match criteria.
	or
	Specifies the Layer 3 packet length in the IP header as a match criterion in a class map.
	• The example configures the route map to match packets that are 3 to 200 bytes in length.

Step 6 **end** **Example:** `Router(config-route-map)# end`	Exits route-map configuration mode and returns to privileged EXEC mode.

Configuring Multi-VRF Selection Using PBR and IP VRF Receive on the Interface

The route map is attached to the incoming interface with the **ip policy route-map** interface configuration command.

The source IP address must be added to the VRF selection table. VRF selection is a one-way (unidirectional) feature. It is applied to the incoming interface. If a **match** and **set** operation occurs in the route map but there is no receive entry in the local VRF table, the packet is dropped if the packet destination is local.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip policy route-map** *map-tag*
5. **ip vrf receive** *vrf-name*
6. **end**

DETAILED STEPS

	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* [*name-tag*] **Example:** `Router(config)# interface FastEthernet 0/1/0`	Configures an interface and enters interface configuration mode.
Step 4	**ip policy route-map** *map-tag* **Example:** `Router(config-if)# ip policy route-map map1`	Identifies a route map to use for policy routing on an interface. • The configuration example attaches the route map named map1 to the interface.

	Command or Action	Purpose
Step 5	**ip vrf receive** *vrf-name* **Example:** `Router(config-if)# ip vrf receive VRF-1`	Adds the IP addresses that are associated with an interface into the VRF table. • This command must be configured for each VRF that will be used for VRF selection.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying the Configuration of Multi-VRF Selection Using PBR

To verify the configuration of the Multi-VRF Selection Using Policy-Based Routing (PBR) feature, perform the following steps. You can enter the commands in any order.

SUMMARY STEPS

1. **show ip access-list** [*access-list-number* | *access-list-name*]
2. **show route-map** [*map-name*]
3. **show ip policy**

DETAILED STEPS

Step 1 **show ip access-list** [*access-list-number* | *access-list-name*]

To verify the configuration of match criteria for PBR multi-VRF selection, use the **show ip access-list** command. The following **show ip access-list** command output displays three subnet ranges defined as match criteria in three standard access lists:

Example:

```
Router# show ip access-list

Standard IP access list 40
 10 permit 10.1.0.0, wildcard bits 0.0.255.255
Standard IP access list 50
 10 permit 10.2.0.0, wildcard bits 0.0.255.255
Standard IP access list 60
 10 permit 10.3.0.0, wildcard bits 0.0.255.255
```

Step 2 **show route-map** [*map-name*]

Use this command to verify **match** and **set** commands within the route map:

Example:

```
Router# show route-map
```

To verify the route-map configuration, use the **show route-map** command. The output displays the match criteria and set action for each route-map sequence. The output also displays the number of packets and bytes that have been policy routed per each route-map sequence.

Example:

```
Router# show route-map map1
route-map map1, permit, sequence 10
Match clauses:
Set clauses:
 ip next-hop vrf myvrf 10.5.5.5  10.6.6.6  10.7.7.7
 ip next-hop global 10.8.8.8  10.9.9.9
Policy routing matches: 0 packets, 0 bytes
Router# show route-map map2
route-map map2, permit, sequence 10
Match clauses:
Set clauses:
 vrf myvrf
Policy routing matches: 0 packets, 0 bytes
Router# show route-map map3
route-map map3, permit, sequence 10
Match clauses:
Set clauses:
 global
Policy routing matches: 0 packets, 0 bytes
```

The following **show route-map** command displays output from the **set ip vrf next-hop** command:

Example:

```
Router(config)# route-map test

Router(config-route-map)# set ip vrf myvrf next-hop
Router(config-route-map)# set ip vrf myvrf next-hop 192.168.3.2
Router(config-route-map)# match ip address 255 101
Router(config-route-map)# end
Router# show route-map

route-map test, permit, sequence 10
 Match clauses:
  ip address (access-lists): 101
 Set clauses:
  ip vrf myvrf next-hop 192.168.3.2
 Policy routing matches: 0 packets, 0 bytes
```

The following **show route-map** command displays output from the **set ip global** command:

Example:

```
Router(config)# route-map test
Router(config-route-map)# match ip address 255 101
Router(config-route-map)# set ip global next-hop 192.168.4.2
Router(config-route-map)# end
Router# show route-map

*May 25 13:45:55.551: %SYS-5-CONFIG_I: Configured from console by consoleout-map
route-map test, permit, sequence 10
 Match clauses:
  ip address (access-lists): 101
 Set clauses:
  ip global next-hop 192.168.4.2
 Policy routing matches: 0 packets, 0 bytes
```

Step 3 **show ip policy**

To verify the PBR multi-VRF selection policy, use the **show ip policy** command:

Example:

```
Router# show ip policy
```

The following **show ip policy** command output displays the interface and associated route map that is configured for policy routing:

Example:

```
Router# show ip policy
Interface                Route map
FastEthernet0/1/0        PBR-VRF-Selection
```

Configuration Examples for Multi-VRF Selection Using Policy-Based Routing

- Defining the Match Criteria for Multi-VRF Selection Using PBR Example, page 305
- Configuring Multi-VRF Selection in a Route Map Example, page 306

Defining the Match Criteria for Multi-VRF Selection Using PBR Example

In the following example, three standard access lists are created to define match criteria for three different subnetworks. Any packets received on FastEthernet interface 0/1/0 will be policy routed through the PBR-VRF-Selection route map to the VRF that is matched in the same route-map sequence. If the source IP address of the packet is part of the 10.1.0.0/24 subnet, VRF1 will be used for routing and forwarding.

```
access-list 40 permit source 10.1.0.0 0.0.255.255
access-list 50 permit source 10.2.0.0 0.0.255.255
access-list 60 permit source 10.3.0.0 0.0.255.255
route-map PBR-VRF-Selection permit 10
 match ip address 40
 set vrf VRF1
 !
route-map PBR-VRF-Selection permit 20
 match ip address 50
 set vrf VRF2
 !
route-map PBR-VRF-Selection permit 30
 match ip address 60
 set vrf VRF3
 !
interface FastEthernet 0/1/0
 ip address 192.168.1.6 255.255.255.252
 ip vrf forwarding VRF4
 ip policy route-map PBR-VRF-Selection
 ip vrf receive VRF1
 ip vrf receive VRF2
 ip vrf receive VRF3
```

Configuring Multi-VRF Selection in a Route Map Example

The following example shows a **set ip vrf next-hop** command that applies policy-based routing to the VRF interface named myvrf and specifies that the IP address of the next hop is 10.0.0.2:

```
Router(config)# route-map map1 permit
Router(config)# set vrf myvrf
Router(config-route-map)# set ip vrf myvrf next-hop 10.0.0.2
Router(config-route-map)# match ip address 101
Router(config-route-map)# end
```

The following example shows a **set ip global** command that specifies that the router should use the next hop address 10.0.0.1 in the global routing table:

```
Router(config-route-map)# set ip global next-hop 10.0.0.1
```

Additional References

Related Documents

Related Topic	Document Title
MPLS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Multiprotocol Label Switching Command Reference*
IP access list commands	*Cisco IOS Security 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 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 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 Multi-VRF Selection Using Policy-Based Routing

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 17 Feature Information for Multi-VRF Selection Using Policy-Based Routing

Feature Name	Releases	Feature Information
Multi-VRF Selection Using Policy-Based Routing (PBR)	12.2(33)SRB1 12.2(33)SXH1 12.4(24)T	The Multi-VRF Selection Using Policy-Based Routing (PBR) feature allows a specified interface on a provider edge (PE) router to route packets to VPNs based on packet length or match criteria defined in an IP access list. This feature and the *Directing MPLS VPN Traffic Using a Source IP Address* feature can be configured together on the same interface. In 12.2(33)SRB1, this feature was introduced. In 12.2(33)SXH1, support was added. The following commands were modified: **set ip global next-hop** and **set ip vrf next-hop**. In 12.4(24)T, this feature was integrated.

Glossary

CE router—customer edge router. A router that is part of a customer network and that interfaces to a provider edge (PE) router.

Inherit-VRF routing—Packets arriving at a VRF interface are routed by the same outgoing VRF interface.

Inter-VRF routing—Packets arriving at a VRF interface are routed via any other outgoing VRF interface.

IP —Internet Protocol. Network layer protocol in the TCP/IP stack offering a connectionless internetwork service. IP provides features for addressing, type-of-service specification, fragmentation and reassembly, and security. Defined in RFC 791.

PBR —policy-based routing. PBR allows a user to manually configure how received packets should be routed.

PE router—provider edge router. A router that is part of a service provider's network and that is connected to a CE router. It exchanges routing information with CE devices by using static routing or a routing protocol such as BGP, RIPv1, or RIPv2.

VPN —Virtual Private Network. A collection of sites sharing a common routing table. A VPN provides a secure way for customers to share bandwidth over an ISP backbone network.

VRF —A VPN routing and forwarding instance. A VRF 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 what goes into the forwarding table.

VRF-lite —A feature that enables a service provider to support two or more VPNs, where IP addresses can be overlapped among the VPNs.