MPLS: Layer 3 VPNs: Inter-AS and CSC Configuration Guide, Cisco IOS Release 15M&T

First Published: November 21, 2012

Last Modified: March 15, 2013

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

CHAPTER 1 **MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses** 1

Finding Feature Information 1

Prerequisites for MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses 2

Restrictions for MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses 2

Information About MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses 2

MPLS VPN Inter-AS Introduction 2

Benefits of MPLS VPN Inter-AS 2

Use of Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses 3

Information Exchange in an MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses 4

Transmission of Information in an MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses 4

Exchange of VPN Routing Information in an MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses 6

Packet Forwarding Between MPLS VPN Inter-AS Systems with ASBRs Exchanging VPN-IPv4 Addresses 7

Use of a Confederation for MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses 9

How to Configure MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses 11

Configuring the ASBRs to Exchange VPN-IPv4 Addresses 11

Configuring EBGP Routing to Exchange VPN Routes Between Subautonomous Systems in a Confederation 13

Verifying Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses 15

Configuration Examples for MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses 16

Example: Configuring MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses 16

Example: Configuration for Autonomous System 1 CE1 17

Example: Configuration for Autonomous System 1 PE1 17
Example: Configuration for Autonomous System 1 P1 18
Example: Configuration for Autonomous System 1 EBGP1 19
Example: Configuration for Autonomous System 2 EBGP2 19
Example: Configuration for Autonomous System 2 P2 20
Example: Configuration for Autonomous System 2 PE2 21
Example: Configuration for Autonomous System 2 CE2 22
Example: Configuring MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses in a Confederation 22
Example: Configuration for Autonomous System 1 CE1 23
Example: Configuration for Autonomous System 1 PE1 23
Example: Configuration for Autonomous System 1 P1 24
Example: Configuration for Autonomous System 1 ASBR1 25
Example: Configuration for Autonomous System 2 ASBR2 26
Example: Configuration for Autonomous System 2 P2 27
Example: Configuration for Autonomous System 2 PE2 28
Example: Configuration for Autonomous System 2 CE2 29
Additional References 29
Feature Information for MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses 30

CHAPTER 2 **MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels 33**
Finding Feature Information 33
Prerequisites for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels 34
Restrictions for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels 35
Information About MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels 35
MPLS VPN Inter-AS Introduction 35
Benefits of MPLS VPN Inter-AS 36
Information About Using MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels 36
Benefits of MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels 36

How the Inter-AS Works When ASBRs Exchange IPv4 Routes with MPLS Labels 37
BGP Routing Information 37
Types of BGP Messages and MPLS Labels 38
How BGP Sends MPLS Labels with Routes 38
How to Configure MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels 38
Configuring the ASBRs to Exchange IPv4 Routes and MPLS Labels 39
Configuring the Route Reflectors to Exchange VPN-IPv4 Routes 41
Configuring the Route Reflector to Reflect Remote Routes in Its Autonomous System 43
Verifying the MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels Configuration 46
Verifying the Route Reflector Configuration 47
Verifying that CE1 Can Communicate with CE2 47
Verifying that PE1 Can Communicate with CE2 48
Verifying that PE2 Can Communicate with CE2 50
Verifying the ASBR Configuration 52
Verifying the ASBR Configuration 52
Configuration Examples for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels 53
Configuring MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels over an MPLS VPN Service Provider Examples 53
Route Reflector 1 Configuration Example (MPLS VPN Service Provider) 54
ASBR1 Configuration Example (MPLS VPN Service Provider) 55
Route Reflector 2 Configuration Example (MPLS VPN Service Provider) 56
ASBR2 Configuration Example (MPLS VPN Service Provider) 57
Configuring MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels over a Non-MPLS VPN Service Provider Examples 58
Route Reflector 1 Configuration Example (Non-MPLS VPN Service Provider) 59
ASBR1 Configuration Example (Non-MPLS VPN Service Provider) 60
Route Reflector 2 Configuration Example (Non-MPLS VPN Service Provider) 61
ASBR2 Configuration Example (Non-MPLS VPN Service Provider) 62
ASBR3 Configuration Example (Non-MPLS VPN Service Provider) 63
Route Reflector 3 Configuration Example (Non-MPLS VPN Service Provider) 64
ASBR4 Configuration Example (Non-MPLS VPN Service Provider) 65
Additional References 66

Feature Information for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels 68

CHAPTER 3 MPLS VPN Multipath Support for Inter-AS VPNs 69

Finding Feature Information 69

Restrictions for MPLS VPN Multipath Support for Inter-AS VPNs 69

Information About MPLS VPN Multipath Support for Inter-AS VPNs 70

Load Sharing with MPLS VPN Inter-AS ASBRs 70

How to Configure MPLS VPN Multipath Support for Inter-AS VPNs 71

Configuring eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs 71

Example 76

Verifying eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs 76

Configuration Examples for MPLS VPN Multipath Support for Inter-AS VPNs 78

Example: Configuring eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs 78

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 CE1 79

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 PE1 80

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 P1 80

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 ASBR1 81

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 ASBR2 82

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 ASBR3 83

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 P2 83

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 PE2 84

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 CE2 85

Additional References 85

Feature Information for MPLS VPN Multipath Support for Inter-AS VPNs 87

Glossary 87

CHAPTER 4 MPLS VPN--Inter-AS Option AB 89

Finding Feature Information 90

Prerequisites for MPLS VPN--Inter-AS Option AB 90

Restrictions for MPLS VPN--Inter-AS Option AB 90

Information About MPLS VPN--Inter-AS Option AB 90

MPLS VPN--Inter-AS Option AB Introduction 90

Benefits of MPLS VPN--Inter-AS Option AB 91

Option B Style Peering with Shared Link Forwarding 91

Route Distribution and Packet Forwarding in Non-CSC Networks 91

Route Distribution for VPN 1 92

Packet Forwarding for VPN 1 93

Route Distribution for VPN 2 94

Route Distribution and Packet Forwarding for CSC 94

Route Distribution for VPN 1 95

Packet Forwarding for VPN 1 96

Shared Link Forwarding in Non-CSC Networks 96

Route Distribution for VPN 1 97

Packet Forwarding for VPN1 98

How to Configure Inter-AS Option AB 98

Configuring an Inter-AS Option AB Connection 98

Configuring the VRFs on the ASBR Interface for Each VPN Customer 99

Configuring the MP-BGP Session Between ASBR Peers 100

Configuring the Routing Policy for VPNs that Need Inter-AS Connections 102

Changing an Inter-AS Option A Deployment to an Option AB Deployment 105

Configuration Examples for MPLS VPN--Inter-AS Option AB 107

Examples Inter-AS AB Network Configuration 107

Example CE1 107

Example CE2 107

Example PE1 108

Example Route Reflector 1 109

Example ASBR1 110

Example ASBR 3 111

Example PE2 112

Example CE3 114

Example CE4 114
Examples Inter-AS AB CSC Configuration 115
Example CE1 115
Example CE2 115
Example CE3 116
Example CE4 116
Example PE1 116
Example CSC-CE1 117
Example CSC-PE1 118
Example PE 2 119
Example CSC-CE2 120
Example ASBR1 121
Example CSC-PE 3 124
Example CSC-CE3 125
Example CSC-CE 4 125
Example PE 3 126
Example PE 4 127
Additional References 128
Feature Information for MPLS VPN--Inter-AS Option AB 130
Glossary 131

CHAPTER 5 **MPLS VPN Carrier Supporting Carrier Using LDP and an IGP** 133
Finding Feature Information 133
Prerequisites for MPLS VPN CSC with LDP and IGP 134
Restrictions for MPLS VPN CSC with LDP and IGP 134
Information About MPLS VPN CSC with LDP and IGP 135
MPLS VPN CSC Introduction 135
Benefits of Implementing MPLS VPN CSC 135
Configuration Options for MPLS VPN CSC with LDP and IGP 136
Customer Carrier Is an ISP 136
Customer Carrier Is a BGP MPLS VPN Service Provider 139
How to Configure MPLS VPN CSC with LDP and IGP 141
Configuring the Backbone Carrier Core 141
Prerequisites 141
Verifying IP Connectivity and LDP Configuration in the CSC Core 142

Troubleshooting Tips 144
Configuring VRFs for CSC-PE Routers 144
Troubleshooting Tips 146
Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier 146
Troubleshooting Tips 148
Configuring the CSC-PE and CSC-CE Routers 148
Prerequisites 148
Configuring LDP on the CSC-PE and CSC-CE Routers 148
Enabling MPLS Encapsulation on the CSC-PE and CSC-CE Routers 150
Verifying the Carrier Supporting Carrier Configuration 151
Configuration Examples for MPLS VPN CSC with LDP and IGP 152
MPLS VPN CSC Network with a Customer Who Is an ISP Example 152
CSC-CE1 Configuration 152
CSC-PE1 Configuration 153
CSC-PE2 Configuration 154
CSC-CE2 Configuration 156
MPLS VPN CSC Network with a Customer Who Is an MPLS VPN Provider Example 157
CE1 Configuration 157
PE1 Configuration 158
CSC-CE1 Configuration 159
CSC-PE1 Configuration 159
CSC-PE2 Configuration 161
CSC-CE2 Configuration 162
PE2 Configuration 163
CE2 Configuration 164
MPLS VPN CSC Network That Contains Route Reflectors Example 165
Backbone Carrier Configuration 166
Route Reflector 1 (72K-37-1) Configuration 166
Route Reflector 2 (72K-38-1) Configuration 167
CSC-PE1 (75K-37-3) Configuration 168
CSC-PE2 (75K-38-3) Configuration 169
Customer Carrier Site 1 Configuration 171
PE1 (72K-36-8) Configuration 171
CSC-CE1 (72K-36-9) Configuration 172
PE2 (72K-36-7) Configuration 173

Route Reflector 3 (36K-38-4) Configuration 174
CE1 (36K-36-1) Configuration 175
Customer Carrier Site 2 Configuration 175
CSC-CE3 (72K-36-6) Configuration 175
PE3 (72K-36-4) Configuration 176
CSC-CE4 (72K-36-5) Configuration 177
Route Reflector 4 (36K-38-5) Configuration 178
CE2 (36K-36-2) Configuration 179
CE3 (36K-36-3) Configuration 179
MPLS VPN CSC Network with a Customer Who Has VPNs at the Network Edge Example 180
Backbone Carrier Configuration 181
CSC-PE1 (72K-36-9) Configuration 181
P1 (75K-37-3) Configuration 182
P2 (75K-38-3) Configuration 184
CSC-PE2 (72K-36-5) Configuration 185
Customer Carrier Site 1 Configuration 187
CSC-CE1 (72K-36-8) Configuration 187
PE2 (72K-36-7) Configuration 188
CE1 (36K-36-1) Configuration 189
Customer Carrier Site 2 Configuration 189
CSC-CE2 (72K-36-4) Configuration 189
PE2 (72K-36-6) Configuration 191
CE2 (36K-38-4) Configuration 192
CE3 (36K-38-5) Configuration 192
Additional References for MPLS VPN Carrier Supporting Carrier Using LDP and an IGP 193
Feature Information for MPLS VPN CSC with LDP and IGP 194
Glossary 194

CHAPTER 6 **MPLS VPN Carrier Supporting Carrier with BGP** 197
Finding Feature Information 197
Prerequisites for MPLS VPN CSC with BGP 198
Restrictions for MPLS VPN CSC with BGP 198
Information About MPLS VPN CSC with BGP 198
MPLS VPN CSC Introduction 198

Benefits of Implementing MPLS VPN CSC 198
Benefits of Implementing MPLS VPN CSC with BGP 199
Configuration Options for MPLS VPN CSC with BGP 200
Customer Carrier Is an ISP with an IP Core 200
Customer Carrier Is an MPLS Service Provider With or Without VPN Services 201
How to Configure MPLS VPN CSC with BGP 201
Identifying the Carrier Supporting Carrier Topology 201
What to Do Next 202
Configuring the Backbone Carrier Core 202
Prerequisites 203
Verifying IP Connectivity and LDP Configuration in the CSC Core 203
Troubleshooting Tips 205
Configuring VRFs for CSC-PE Routers 205
Troubleshooting Tips 207
Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier 207
Troubleshooting Tips 209
Configuring the CSC-PE and CSC-CE Routers 209
Configuring CSC-PE Routers 209
Troubleshooting Tips 211
Configuring CSC-CE Routers 212
Verifying Labels in the CSC-PE Routers 214
Verifying Labels in the CSC-CE Routers 216
Configuring the Customer Carrier Network 218
Prerequisites 218
Verifying IP Connectivity in the Customer Carrier 218
Configuring a Customer Carrier Core Router as a Route Reflector 219
Troubleshooting Tips 221
Configuring the Customer Site for Hierarchical VPNs 221
Defining VPNs on PE Routers for Hierarchical VPNs 222
Configuring BGP Routing Sessions on the PE Routers for Hierarchical VPNs 223
Verifying Labels in Each PE Router for Hierarchical VPNs 225
Configuring CE Routers for Hierarchical VPNs 226
Verifying IP Connectivity in the Customer Site 228
Configuration Examples for MPLS VPN CSC with BGP 230
Configuring the Backbone Carrier Core Examples 231

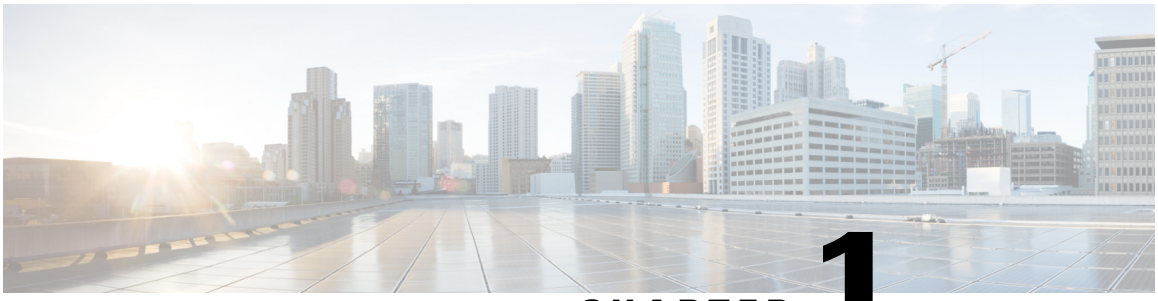
Verifying IP Connectivity and LDP Configuration in the CSC Core Example 231
Configuring VRFs for CSC-PE Routers Example 232
Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier Example 232
Configuring the Links Between CSC-PE and CSC-CE Routers Examples 233
Configuring the CSC-PE Routers Examples 233
Configuring the CSC-CE Routers Examples 234
Verifying Labels in the CSC-PE Routers Examples 235
Verifying Labels in the CSC-CE Routers Examples 237
Configuring the Customer Carrier Network Examples 239
Verifying IP Connectivity in the Customer Carrier Example 239
Configuring a Customer Carrier Core Router as a Route Reflector Example 240
Configuring the Customer Site for Hierarchical VPNs Examples 240
Configuring PE Routers for Hierarchical VPNs Examples 240
Verifying Labels in Each PE Router for Hierarchical VPNs Examples 241
Configuring CE Routers for Hierarchical VPNs Examples 242
Verifying IP Connectivity in the Customer Site Examples 242
Additional References 243
Feature Information for MPLS VPN CSC with BGP 244
Glossary 245

CHAPTER 7 **MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs 247**
Finding Feature Information 247
Prerequisites for MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs 248
Restrictions for MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs 248
Information About MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs 250
Load Sharing Using Directly Connected Loopback Peering 250
How to Configure MPLS VPN Load Balancing Support for Inter-AS and CSC VPN 251
Configuring Directly Connected Loopback Peering for MPLS VPN Inter-AS using ASBRs to Exchange VPN-IPv4 Addresses 251
Configuring Loopback Interface Addresses for Directly Connected ASBRs 251
Configuring /32 Static Routes to the eBGP Neighbor Loopback 252
Configuring Forwarding on Connecting Loopback Interfaces 254
Configuring an eBGP Session Between the Loopbacks 255
Verifying That Load Sharing Occurs Between Loopbacks 258

Configuring Directly Connected Loopback Peering for MPLS VPN Inter-AS Using ASBRs to Exchange IPv4 Routes and Labels 258
Configuring Loopback Interface Addresses for Directly Connected ASBRs 259
Configuring /32 Static Routes to the eBGP Neighbor Loopback 260
Configuring Forwarding on Connecting Loopback Interfaces 261
Configuring an eBGP Session Between the Loopbacks 263
Verifying That Load Sharing Occurs Between Loopbacks 266
Configuring Directly Connected Loopback Peering on MPLS VPN Carrier Supporting Carrier 267
Configuring Loopback Interface Addresses on CSC-PE Devices 267
Configuring Loopback Interface Addresses for CSC-CE Routers 268
Configuring /32 Static Routes to the eBGP Neighbor Loopback on the CSC-PE Device 269
Configuring /32 Static Routes to the eBGP Neighbor Loopback on the CSC-CE Device 271
Configuring Forwarding on CSC-PE Interfaces That Connect to the CSC-CE Loopback 272
Configuring Forwarding on CSC-CE Interfaces That Connect to the CSC-PE Loopback 273
Configuring an eBGP Session Between the CSC-PE Device and the CSC-CE Loopback 275
Configuring an eBGP Session Between the CSC-CE Device and the CSC-PE Loopback 277
Verifying That Load Sharing Occurs Between Loopbacks 280
Configuration Examples for MPLS VPN Load Balancing Support for Inter-AS and CSC VPN 281
Examples: Configuring a 32 Static Route from an ASBR to the Loopback Address of Another ASBR 281
Example: Configuring BGP MPLS Forwarding on the Interfaces Connecting ASBRs 281
Example: Configuring VPNv4 Sessions on an ASBR 281
Additional References 282
Feature Information for MPLS VPN Load Balancing Support for Inter-AS and CSC VPN 283

CHAPTER 8 **MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs 285**
Finding Feature Information 285
Prerequisites for MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs 286

Restrictions for MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs 286
Information About MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs 288
Overview of MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs 288
How to Configure MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs 288
Configuring MPLS VPN eBGP Multipath Load Sharing with Inter-AS MPLS VPNs 288
Configuring MPLS VPN eBGP Multipath Load Sharing with Carrier Supporting Carrier on the CSC-PE Devices 291
Configuring MPLS VPN eBGP Multipath Load Sharing with Carrier Supporting Carrier on the CSC-CE Devices 293
Configuration Examples for MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs 296
Example: Configuring MPLS VPN eBGP Multipath Load Sharing with MPLS VPN Inter-AS 296
Example: Configuring MPLS VPN eBGP Multipath Load Sharing with MPLS VPN Carrier Supporting Carrier on the CSC-PE Devices 296
Example: Configuring MPLS VPN eBGP Multipath Load Sharing with MPLS VPN Carrier Supporting Carrier on the CSC-CE Devices 296
Additional References 297
Feature Information for MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs 298



CHAPTER 1

MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

The Multiprotocol Label Switching (MPLS) VPN Inter-AS with Autonomous System Boundary Routers (ASBRs) Exchanging VPN-IPv4 Addresses feature allows a MPLS VPN to span service providers and autonomous systems. This module explains how to enable ASBRs to use Exterior Border Gateway Protocol (EBGP) to exchange IPv4 Network Layer Reachability Information (NLRI) in the form of VPN-IPv4 addresses.

- Finding Feature Information, page 1
- Prerequisites for MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses, page 2
- Restrictions for MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses, page 2
- Information About MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses, page 2
- How to Configure MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses, page 11
- Configuration Examples for MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses, page 16
- Additional References, page 29
- Feature Information for MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses, page 30

Finding Feature Information

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

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 Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

- Before you configure Exterior Border Gateway Protocol (EBGP) routing between autonomous systems or subautonomous systems in an Multiprotocol Label Switching (MPLS) VPN, ensure that you have properly configured all MPLS VPN routing instances and sessions. The configuration tasks outlined in this section build from those configuration tasks. Perform the following tasks as described in the Configuring MPLS Layer 3 VPNs module:
  - Define VPN routing instances
  - Configure BGP routing sessions in the MPLS core
  - Configure provider-edge-provider-edge (PE-to-PE) routing sessions in the MPLS core
  - Configure BGP provider-edge-customer-edge (PE-to-CE) routing sessions
  - Configure a VPN-IPv4 EBGP session between directly connected Autonomous System Boundary Routers (ASBRs)

Restrictions for MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

Multihop VPN-IPv4 Exterior Border Gateway Protocol (EBGP) is not supported.

Information About MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

MPLS VPN Inter-AS Introduction

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.

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.

Benefits of MPLS VPN Inter-AS

An MultiprotocolLabel Switching (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 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. Previously, MPLS VPN could traverse only a single Border Gateway Protocol (BGP) autonomous system service provider backbone. This 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. An autonomous system can be divided 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 Network Layer Reachability Information (NLRI) between the subautonomous systems that form the confederation.

Use of Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

Separate autonomous systems from different service providers can communicate by exchanging IPv4 Network Layer Reachability Information (NLRI) in the form of VPN-IPv4 addresses. The Autonomous System Border Routers (ASBRs) use Exterior Border Gateway Protocol (EBGP) to exchange network reachability 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 EBGP border edge devices to distribute the routes, which include label switching information. Each border edge device rewrites the next hop and labels. See the Information Exchange in an MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses, on page 4 section for more information.

Interautonomous system configurations supported in an MPLS VPN are as follows:

- **Interprovider VPN--** MPLS VPNs that include two or more autonomous systems, connected by separate border edge devices. 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.

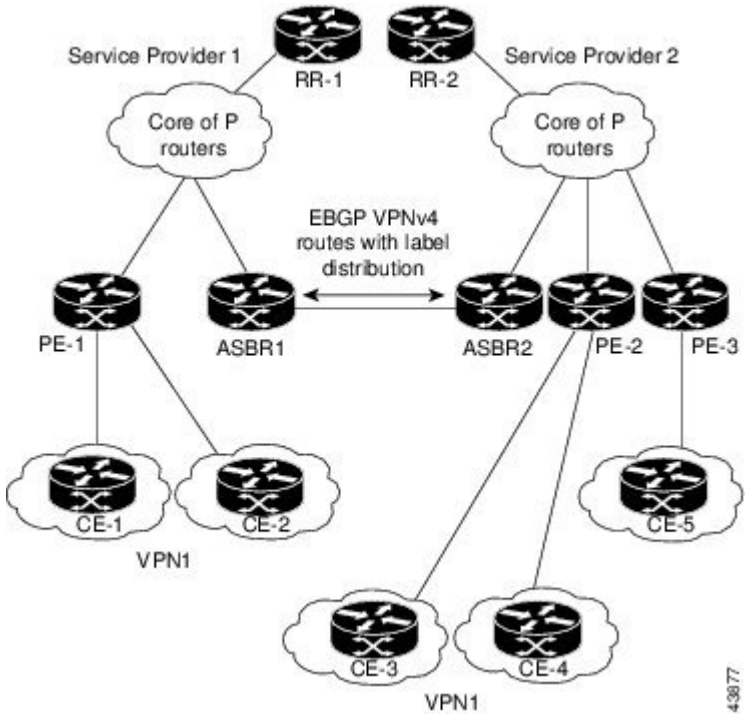
Information Exchange in an MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

This section contains the following topics:

Transmission of Information in an MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

The figure below illustrates an Multiprotocol Label Switching (MPLS) VPN consisting of two separate autonomous systems. Each autonomous system operates under different administrative control and runs a different Interior Gateway Protocol (IGP). Service providers exchange routing information through Exterior Border Gateway Protocol (EBGP) border edge devices (ASBR1, ASBR2).

Figure 1: EBGP Connection Between Two MPLS VPN Inter-AS Systems with ASBRs Exchanging VPN-IPv4 Addresses



This configuration uses the following process to transmit information:

SUMMARY STEPS

1. The provider edge device (PE-1) assigns a label for a route before distributing that route. The PE device uses the multiprotocol extensions of Border Gateway Protocol (BGP) to transmit label mapping information. The PE device distributes the route as a VPN-IPv4 address. The address label and the VPN identifier are encoded as part of the IPv4 Network Layer Reachability Information (NLRI).
2. The two route reflectors (RR-1 and RR-2) reflect VPN-IPv4 internal routes within the autonomous system. The border edge devices (ASBR1 and ASBR2) of the autonomous systems advertise the VPN-IPv4 external routes.
3. The EBGP border edge device (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:
4. The EBGP border edge device (ASBR2) redistributes the route in one of the following ways, depending on its configuration:

DETAILED STEPS

Step 1 The provider edge device (PE-1) assigns a label for a route before distributing that route. The PE device uses the multiprotocol extensions of Border Gateway Protocol (BGP) to transmit label mapping information. The PE device distributes the route as a VPN-IPv4 address. The address label and the VPN identifier are encoded as part of the IPv4 Network Layer Reachability Information (NLRI).

Step 2 The two route reflectors (RR-1 and RR-2) reflect VPN-IPv4 internal routes within the autonomous system. The border edge devices (ASBR1 and ASBR2) of the autonomous systems advertise the VPN-IPv4 external routes.

Step 3 The EBGP border edge device (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:

- The next-hop device is always reachable in the service provider (P) backbone network.
- Tthe label assigned by the distributing device is properly interpreted. (The label associated with a route must be assigned by the corresponding next-hop device.)

Step 4 The EBGP border edge device (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 change. 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 devices in different autonomous systems.

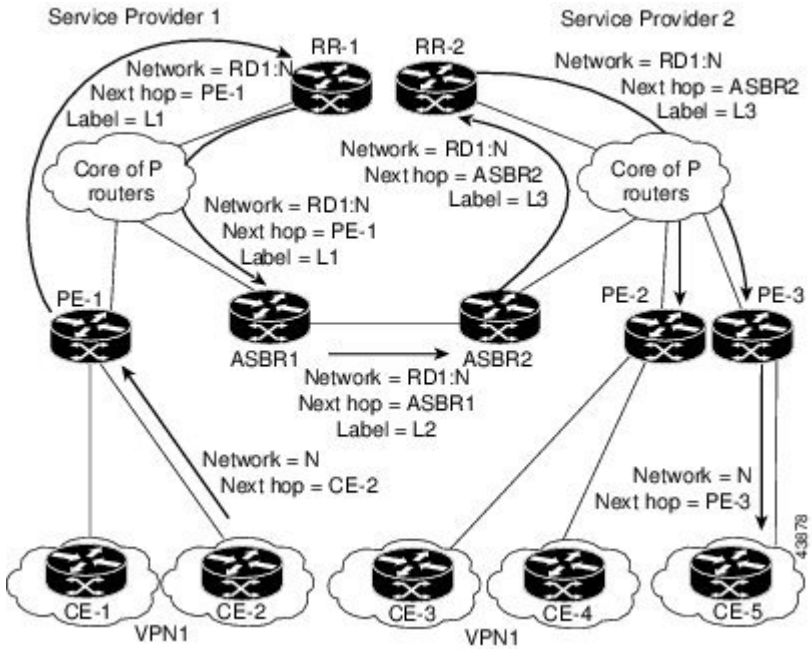
Exchange of VPN Routing Information in an MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

Autonomous systems exchange VPN routing information (routes and labels) to establish connections. To control connections between autonomous systems, the provider edge (PE) devices and Exterior Border Gateway Protocol (EBGP) border edge devices maintain a Label Forwarding Information Base (LFIB). The LFIB manages the labels and routes that the PE devices and EBGP border edge devices 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 conditions to exchange VPN routing information:

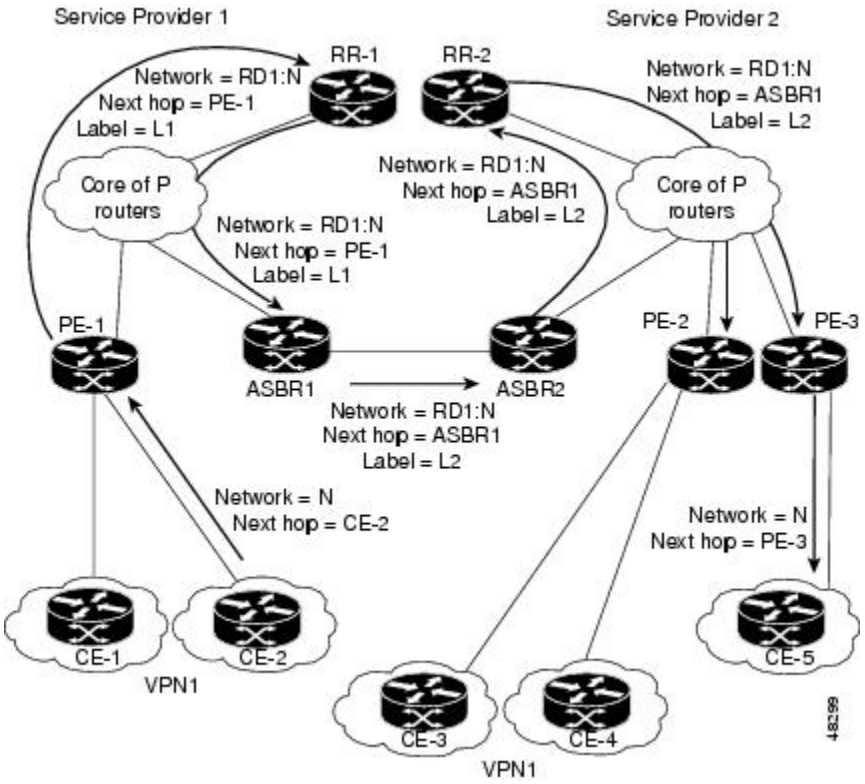
- Routing information includes:
  - The destination network (N)
  - The next-hop field associated with the distributing device
  - A local MPLS label (L)
- An RD1: route distinguisher is part of a destination network address. It makes the VPN-IPv4 route globally unique in the VPN service provider environment.
- The Autonomous System Border Routers (ASBRs) are configured to change the next-hop (next hop-self) when sending VPN-IPv4 Network Layer Reachability Information (NLRI) to the Internal Border Gateway Protocol (IBGP) neighbors. Therefore, the ASBRs must allocate a new label when they forward the NLRI to the IBGP neighbors.

Figure 2: 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 3: Exchanging Routes and Labels with the redistribute connected Command in an MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses



Packet Forwarding Between MPLS VPN Inter-AS Systems with ASBRs Exchanging VPN-IPv4 Addresses

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 Multiprotocol Label Switching (MPLS). Packets use the routing information stored in the Label Forwarding Information Base (LFIB) of each provider edge (PE) device and Exterior Border Gateway Protocol (EBGP) border edge device.

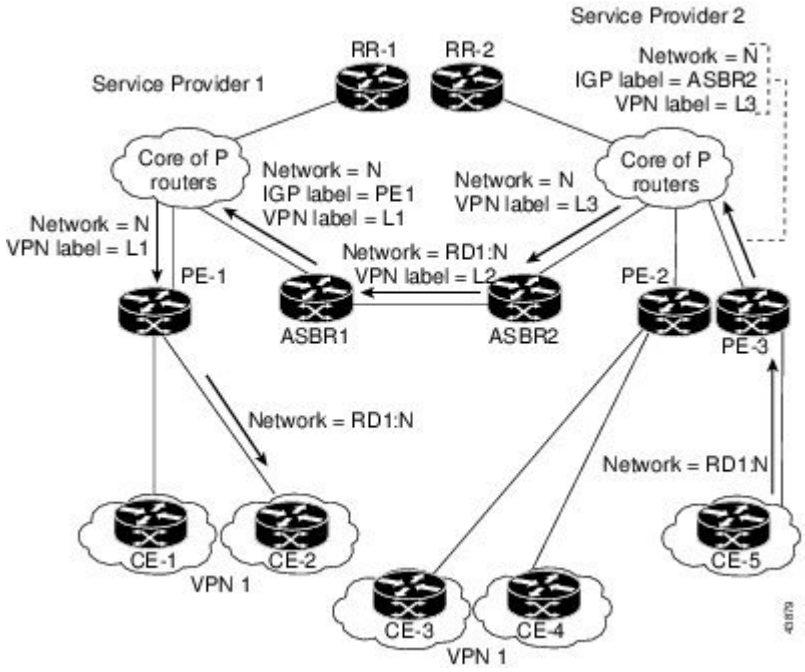
The service provider VPN backbone uses dynamic label switching to forward labels.

Each autonomous system uses standard multilevel labeling to forward packets between the edges of the autonomous system devices (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:

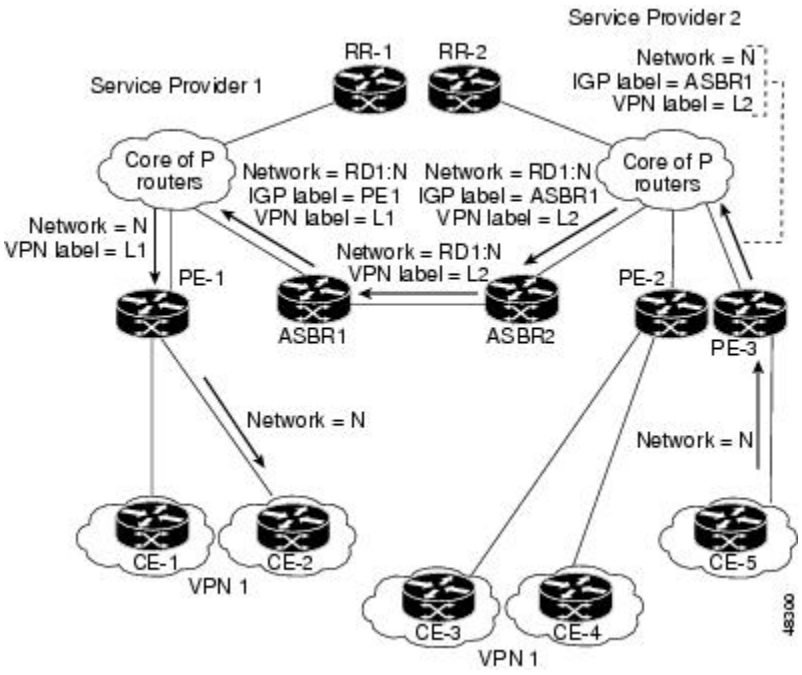
- The first label (IGP route label) directs the packet to the correct PE device or EBGP border edge device. (For example, the Interior Gateway Protocol (IGP) label of ASBR2 points to the ASBR2 border edge device.)
- The second label (VPN route label) directs the packet to the appropriate PE device or EBGP border edge device.

Figure 4: Forwarding Packets Between MPLS VPN Inter-AS Systems with ASBRs Exchanging VPN-IPv4 Addresses



The figure below shows the same packet forwarding method as described in the figure above, except the EBGP device (ASBR1) forwards the packet without reassigning it a new label.

Figure 5: Forwarding Packets Without a New Label Assignment Between MPLS VPN Inter-AS Systems with ASBRs Exchanging VPN-IPv4 Addresses



Use of a Confederation for MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

A confederation is a collection of multiple subautonomous systems that are 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 Interior Gateway Protocol (IGP), such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS). Each subautonomous system also has an Exterior Border Gateway Protocol (EBGP) connection to the other subautonomous systems. The confederation EBGP (CEBGP) border edge devices forward next-hop-self addresses between the specified subautonomous systems. The next-hop-self address forces the Border Gateway Protocol (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:

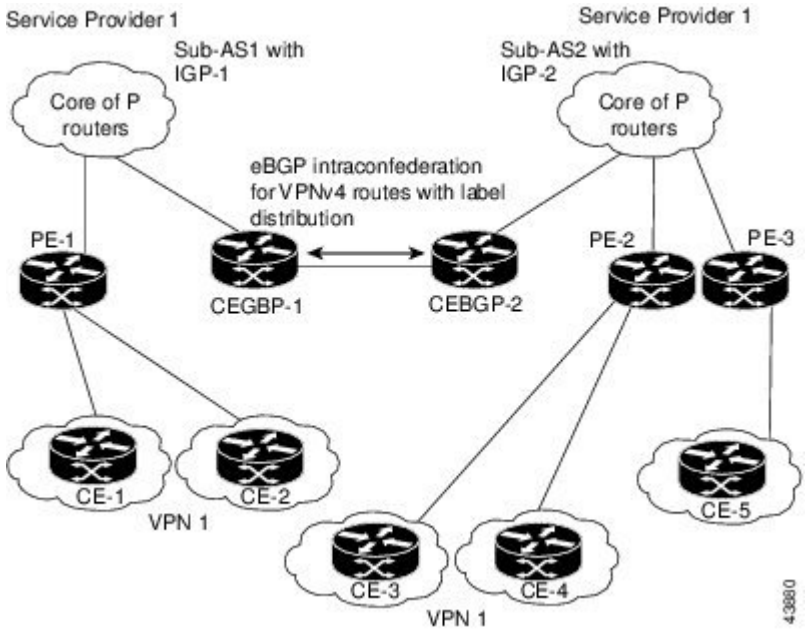
- You can configure a device to forward next-hop-self addresses between only the CEBGP border edge devices (both directions). The subautonomous systems (IBGP peers) at the subautonomous system border do not forward the next-hop-self address. Each subautonomous system runs as a single IGP domain. However, the CEBGP border edge device addresses are known in the IGP domains.
- You can configure a device to forward next-hop-self addresses between the CEBGP border edge devices (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 devices in the domain. The CEBGP border edge device addresses are known in the IGP domains.

The figure below illustrates a typical MPLS VPN confederation configuration. In this confederation configuration:

- The two CEBGP border edge devices exchange VPN-IPv4 addresses with labels between the two subautonomous systems.
- The distributing device 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 6: EBGP Connection Between Two Subautonomous Systems in a Confederation



In this confederation configuration:

- CEBGP border edge devices function as neighboring peers between the subautonomous systems. The subautonomous systems use EBGP to exchange route information.
- Each CEBGP border edge device (CEBGP-1, CEBGP-2) assigns a label for the route before distributing the route to the next subautonomous system. The CEBGP border edge device 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 IPv4 Network Layer Reachability Information (NLRI).
- Each provider edge (PE) and CEBGP border edge device assigns its own label to each VPN-IPv4 address prefix before redistributing the routes. The CEBGP border edge devices 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 device address is distributed throughout the IBGP neighbors, and the two CEBGP border edge devices are known to both confederations.

How to Configure MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

Configuring the ASBRs to Exchange VPN-IPv4 Addresses

To configure an Exterior Border Gateway Protocol (EBGP) Autonomous System Border Router (ASBR) to exchange VPN-IPv4 routes with another autonomous system, perform this task.

Note Issue the **redistribute connected subnets** command in the Interior Gateway Protocol (IGP) configuration portion of the device to propagate host routes for VPN-IPv4 EBGP neighbors to other devices and provider edge devices. Alternatively, you can specify the next-hop-self address when you configure Internal Border Gateway Protocol (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** *peer-group-name* **remote-as** *as-number*
7. **neighbor** *peer-group-name* **activate**
8. **exit-address-family**
9. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 1`	Creates an EBGP routing process and assigns it an autonomous system number. • The autonomous system number is passed along and identifies the device to EBGP devices in another autonomous system.
Step 4	**no bgp default route-target filter** **Example:** `Device(config)# no bgp default route-target filter`	Disables BGP route-target filtering and places the device in configuration mode. • All received BGP VPN-IPv4 routes are accepted by the device.
Step 5	**address-family vpnv4** [**unicast**] **Example:** `Device(config-router)# address-family vpnv4`	Configures a routing session to carry VPNv4 addresses across the VPN backbone and places the device in address family configuration mode. • 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 6	**neighbor** *peer-group-name* **remote-as** *as-number* **Example:** `Device(config-router-af)# neighbor 1 remote-as 2`	Enters the address family configuration mode and specifies a neighboring EBGP peer group. • This EBGP peer group is identified to the specified autonomous system.
Step 7	**neighbor** *peer-group-name* **activate** **Example:** `Device(config-router-af)# neighbor 1 activate`	Activates the advertisement of the VPNv4 address family to a neighboring EBGP device.
Step 8	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits from the address family submode of the router configuration mode.
Step 9	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring EBGP Routing to Exchange VPN Routes Between Subautonomous Systems in a Confederation

Perform this task to configure EBGP routing to exchange 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 devices and provider edge devices, specify the **redistribute connected** command in the IGP configuration portion of the CEBGP device. 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 devices 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 conferderation 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *sub-autonomous-system* **Example:** `Device(config)# router bgp 2`	Creates an EBGP routing process and assigns it an autonomous system number and enters the device in configuration mode. • The subautonomous system number is passed along to identify the device to EBGP devices in other subautonomous systems.
Step 4	**bgp confederation identifier** *as-number* **Example:** `Device(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 conferderation peers** *sub-autonomous-system* **Example:** `Device(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:** `Device(config-router)# no bgp default route-target filter`	Disables BGP route-target community filtering. All received BGP VPN-IPv4 routes are accepted by the device.
Step 7	**address-family vpnv4** [**unicast**] **Example:** `Device(config-router)# address-family vpnv4`	Configures a routing session to carry VPNv4 addresses across the VPN backbone. Each address is made globally unique by the addition of an 8-byte RD. Enters address family configuration mode. • The **unicast** keyword specifies a unicast prefix.
Step 8	**neighbor** *peer-group-name* **remote-as** *as-number* **Example:** `Device(config-router-af)# neighbor 1 remote-as 1`	Enters the address family configuration mode and 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**	Advertises the device as the next hop for the specified neighbor.

	Command or Action	Purpose
	Example: `Device(config-router-af)# neighbor 1 next-hop-self`	• If a next-hop-self address is specified as part of the router configuration, the **redistribute connected** command need not be used.
Step 10	**neighbor** *peer-group-name* **activate** **Example:** `Device(config-router-af)# neighbor R activate`	Activates the advertisement of the VPNv4 address family to a neighboring PE device in the specified subautonomous system.
Step 11	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits from the address family submode of the router configuration mode.
Step 12	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Verifying Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

Perform this task to display the VPN-IPv4 Label Forwarding Information Base (LFIB) entries.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**]
3. **show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [**-***label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]
4. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**] **Example:** `Device# show ip bgp vpnv4 all labels`	Displays VPN address information from the BGP table. • Use the **all** and **labels** keywords to display information about all VPNv4 labels.
Step 3	**show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [-*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**] **Example:** `Device# show mpls forwarding-table`	Displays the contents of the MPLS LFIB (such as VPNv4 prefix/length and BGP next-hop destination for the route).
Step 4	**disable** **Example:** `Device# disable`	Returns to user EXEC mode.

Configuration Examples for MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

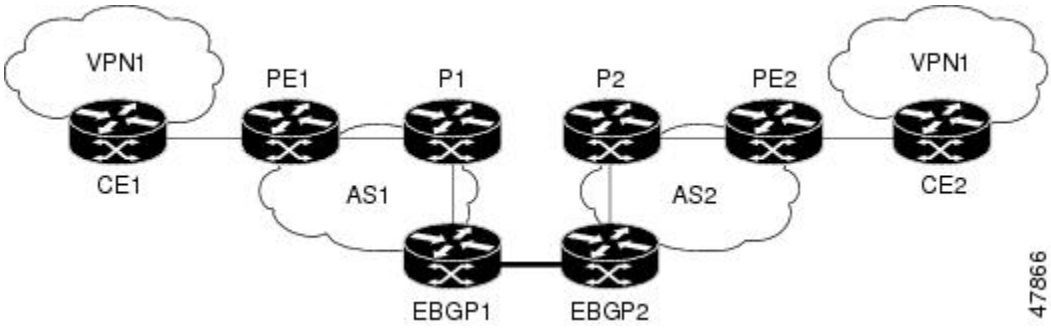
Example: Configuring MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

The network topology in the figure below shows two autonomous systems, which are configured as follows:

- Autonomous system 1 (AS1) includes provider edge 1 (PE1), P1, and Exterior Border Gateway Protocol 1(EBGP1). The Interior Gateway Protocol (IGP) is Open Shortest Path First(OSPF).
- Autonomous system 2 (AS2) includes PE2, P2, and EBGP2. The IGP is Intermediate System to Intermediate System (IS-IS).
- Customer edge 1 (CE1) and CE2 belong to the same VPN, which is called VPN1.
- The P devices are route reflectors.
- EBGP1 is configured with the **redistribute connected subnets** command.

- EBGP2 is configured with the **neighbor next-hop-self** command.

Figure 7: Configuring Two Autonomous Systems



Example: Configuration for Autonomous System 1 CE1

The following example shows how to configure CE1 in VPN1 in a topology with two autonomous systems:

```
interface Loopback1
 ip address 10.1.0.4 255.0.0.0
!
interface GigabitEthernet0/0/0
   no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
!
interface GigabitEthernet0/5/3 point-to-point
  ip address 10.1.0.2 255.0.0.0
 frame-relay interface-dlci 22
!
router ospf 1
 network 192.168.3.0 255.255.0.0 area 0
```

Example: Configuration for Autonomous System 1 PE1

The following example shows how to configure PE1 in AS1 in a topology with two autonomous systems:

```
ip cef
!
ip vrf V1
 rd 1:105
 route-target export 1:100
 route-target import 1:100
!
interface GigabitEthernet0/0/0
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 2000000
!
interface GigabitEthernet0/0/0.3 point-to-point
 ip vrf forwarding V1
 ip address 192.168.2.4 255.255.0.0
 frame-relay interface-dlci 22
!
interface GigabitEthernet0/5/3
 ip address 192.168.3.5 255.255.0.0
 tag-switching ip
!
```

```
router ospf 1
 log-adjacency-changes
 network 192.168.41.0 255.255.0.0 area 0
!
router ospf 10 vrf V1
 log-adjacency-changes
 redistribute bgp 1 metric 100 subnets
 network 192.168.41.0 255.255.0.0 area 0
!
router bgp 1
 no synchronization
 neighbor 1 peer-group
 neighbor 1 remote-as 1
 neighbor 1 update-source Loopback0
 neighbor 192.168.11.10 peer-group R
 no auto-summary
 !
 address-family ipv4 vrf V1
  redistribute ospf 10
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor R activate
  neighbor R send-community extended
  neighbor 192.168.11.10 peer-group R
  no auto-summary
  exit-address-family
```

Example: Configuration for Autonomous System 1 P1

The following example shows how to configure P1 in AS1 in a topology with two autonomous systems:

```
ip cef
!
interface Loopback0
 ip address 10.1.2.1 255.0.0.0
!
interface GigabitEthernet0/4/7
 ip address 10.1.0.4 255.0.0.0
 tag-switching ip
!
interface GigabitEthernet0/5/3
 ip address 10.2.0.3 255.0.0.0
 duplex auto
 speed auto
 tag-switching ip
!
router ospf 1
 log-adjacency-changes
 network 10.1.0.2 255.0.0.0 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 1
 neighbor R update-source Loopback0
 neighbor R route-reflector-client
 neighbor 192.168.3.4 peer-group R
 neighbor 192.168.3.5 peer-group R
 !
 address-family vpnv4
  neighbor R activate
  neighbor R route-reflector-client
  neighbor R send-community extended
  neighbor 192.168.3.4 peer-group R
```

```
 neighbor 192.168.3.5 peer-group R
 exit-address-family
```

Example: Configuration for Autonomous System 1 EBGP1

The following example shows how to configure EBGP1 in AS1 in a topology with two autonomous systems:

```
ip cef
!
interface Loopback0
 ip address 10.2.2.1 255.0.0.0
!
!
ip cef
!
interface Loopback0
 ip address 10.2.2.1 255.0.0.0
!
interface GigabitEthernetEthernet0/5/3
 ip address 10.1.0.5 255.0.0.0
 tag-switching ip
!
interface GigabitEthernet0/0/0
!
interface GigabitEthernet0/0/0.1 point-to-point
!
router ospf 1
 log-adjacency-changes
 redistribute connected subnets
 network 10.1.0.5 255.0.0.0 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
 neighbor R update-source Loopback0
 neighbor 10.1.0.2 remote-as 2
 neighbor 10.1.0.2 peer-group R
 no auto-summary
 !
 address-family vpnv4
  neighbor R activate
  neighbor R send-community extended
  neighbor 10.1.0.2 activate
  neighbor 10.1.0.2 send-community extended
  neighbor 10.1.0.2 peer-group R
  no auto-summary
  exit-address-family
```

Example: Configuration for Autonomous System 2 EBGP2

The following example shows how to configure EBGP2 in AS2 in a topology with two autonomous systems:

```
ip cef
!
ip vrf V1
 rd 2:103
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 10.1.1.2 255.0.0.0
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
```

```
 ip address 10.1.1.2 255.0.0.0
!
interface GigabitEthernet0/4/7
 no ip address
 encapsulation frame-relay
 load-interval 30
 no fair-queue
 clockrate 2000000
!
interface GigabitEthernet0/0/3 point-to-point
 ip unnumbered Loopback0
 ip router isis
 tag-switching ip
 frame-relay interface-dlci 23
!
interface GigabitEthernet0/0/4
 no ip address
 atm clock INTERNAL
 no atm scrambling cell-payload
 no atm ilmi-keepalive
!
interface GigabitEthernet0/0/4.1 point-to-point
 ip address 10.1.0.5 255.0.0.0
 pvc 1/100
!
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 10.1.0.1 remote-as 1
 neighbor 10.1.1.2 remote-as 2
 neighbor 10.1.1.2 update-source Loopback0
 neighbor 10.1.1.2 next-hop-self
!
 address-family ipv4 vrf V1
  redistribute connected
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor 10.1.0.1 activate
  neighbor 10.1.0.1 send-community extended
  neighbor 10.1.1.2 activate
  neighbor 10.1.1.2 next-hop-self
  neighbor 10.1.1.2 send-community extended
  exit-address-family
```

Example: Configuration for Autonomous System 2 P2

The following example shows how to configure P2 in AS2 in a topology with two autonomous systems:

```
ip cef
!
ip vrf V1
 rd 2:108
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 10.1.0.2 255.0.0.0
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
 ip address 10.1.0.2 255.0.0.0
```

```
!
interface GigabitEthernet0/0/0
 ip address 10.2.1.4 255.0.0.0
 ip router isis
 tag-switching ip
!
interface GigabitEthernet0/0/3
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
!
interface GigabitEthernet0/0/3.1 point-to-point
 ip unnumbered Loopback0
 ip router isis
 tag-switching ip
 frame-relay interface-dlci 23
!
router isis
 net aa.0002.0000.0000.0008.00
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 2
 neighbor R update-source Loopback0
 neighbor R route-reflector-client
 neighbor 10.1.2.1 peer-group R
 neighbor 10.0.1.2 peer-group R
 !
 address-family ipv4 vrf V1
  redistribute connected
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor R activate
  neighbor R route-reflector-client
  neighbor R send-community extended
  neighbor 10.1.2.1 peer-group R
  neighbor 10.0.1.2 peer-group R
  exit-address-family
```

Example: Configuration for Autonomous System 2 PE2

The following example shows how to configure PE2 in AS2 in a topology with two autonomous systems:

```
ip cef
!
ip vrf V1
 rd 2:109
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 192.168.11.10 255.255.0.0
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
 ip address 192.168.11.10 255.255.0.0
!
interface GigabitEthernet0/5/3
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
 no fair-queue
 clockrate 2000000
!
```

```
interface GigabitEthernet0/5/3.1 point-to-point
 ip vrf forwarding V1
 ip unnumbered Loopback1
 frame-relay interface-dlci 24
!
interface GigabitEthernet0/0/0
 ip address 192.168.2.10 255.255.0.0
 ip router isis
 tag-switching ip
!
router ospf 10 vrf V1
 log-adjacency-changes
 redistribute bgp 2 subnets
 network 192.168.2.2 255.255.0.0 area 0
!
router isis
 net 49.0002.0000.0000.0009.00
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor 192.168.3.2 remote-as 2
 neighbor 192.168.3.2 update-source Loopback0
 !
 address-family ipv4 vrf V1
  redistribute connected
  redistribute ospf 10
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor 192.168.3.2 activate
  neighbor 192.168.3.2 send-community extended
  exit-address-family v
```

Example: Configuration for Autonomous System 2 CE2

The following example shows how to configure CE2 in VPN1 in a topology with two autonomous systems:

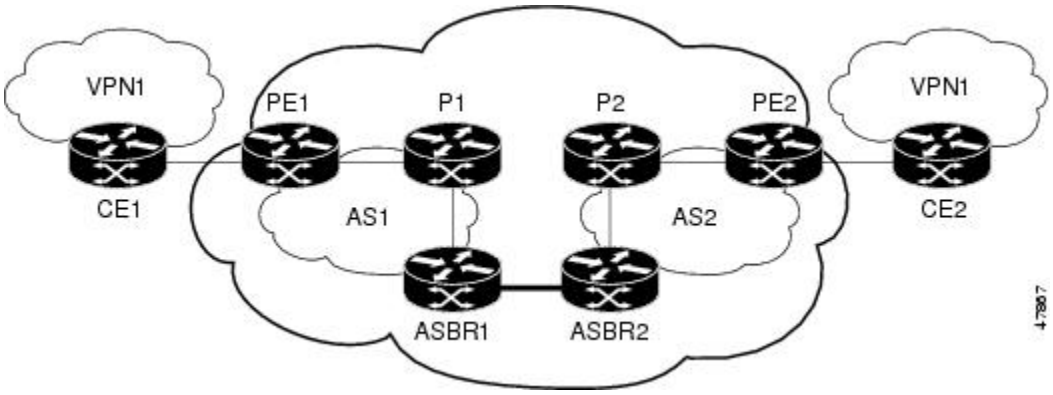
```
interface Loopback0
 ip address 192.168.2.2 255.255.0.0
!
interface GigabitEthernet0/0/0
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 2000000
!
interface GigabitEthernet0/0/0.1 point-to-point
 ip unnumbered Loopback0
 frame-relay interface-dlci 24
!
router ospf 1
 network 192.168.4.6 255.255.0.0 area 0
```

Example: Configuring MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses in a Confederation

The network topology in the figure below shows a single internet service provider, 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) includes provider edge 1 (PE1), P1, Autonomous System Border Router 1 (ASBR1). The Interior Gateway Protocol (IGP) is Open Shortest Path First (OSPF).
- Autonomous system 2 (AS2) includes PE2, P2, ASBR2. The IGP is Intermediate System to Intermediate System (IS-IS).
- Customer edge 1 (CE1) and CE2 belong to the same VPN, which is called VPN1.
- The P devices are route reflectors.
- ASBR1 is configured with the **redistribute connected subnets** command.
- ASBR2 is configured with the **neighbor next-hop-self**command.

Figure 8: Configuring Two Autonomous Systems in a Confederation



Example: Configuration for Autonomous System 1 CE1

The following example shows how to configure CE1 in VPN1 in a confederation topology:

```
interface Loopback1
 ip address 192.168.3.4 255.255.255.255
!
interface GigabitEthernet0/4/7
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
!
interface GigabitEthernet0/4/7.1 point-to-point
 ip address 192.168.1.3 255.255.0.0
 frame-relay interface-dlci 22
!
router ospf 1
 network 192.168.0.1 255.255.0.0 area 0
```

Example: Configuration for Autonomous System 1 PE1

The following example shows how to configure PE1 in AS1 in a confederation topology:

```
ip cef
!
ip vrf V1
 rd 1:105
 route-target export 1:100
```

```
 route-target import 1:100
!
interface GigabitEthernet0/0/0
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 2000000
!
interface GigabitEthernet0/0/0.3 point-to-point
 ip vrf forwarding V1
 ip address 10.0.2.4 255.0.0.0
 frame-relay interface-dlci 22
!
interface GigabitEthernet0/4/7
 ip address 10.1.2.6 255.0.0.0
 tag-switching ip
!
router ospf 1
 log-adjacency-changes
 network 10.1.8.4 255.0.0.0 area 0
!
router ospf 10 vrf V1
 log-adjacency-changes
 redistribute bgp 1 metric 100 subnets
 network 10.1.8.4 255.0.0.0 area 0
!
router bgp 1
 no synchronization
 bgp confederation identifier 100
 bgp confederation identifier 100
 neighbor 1 peer-group
 neighbor 1 remote-as 1
 neighbor 1 update-source Loopback0
 neighbor 10.2.1.2 peer-group R
 no auto-summary
 !
 address-family ipv4 vrf V1
  redistribute ospf 10
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor R activate
  neighbor R send-community extended
  neighbor 10.2.1.2 peer-group R
  no auto-summary
  exit-address-family
```

Example: Configuration for Autonomous System 1 P1

The following example shows how to configure P1 in AS1 in a confederation topology:

```
ip cef
!
interface Loopback0
 ip address 10.0.0.2 255.0.0.0
!
interface GigabitEthernet0/0/0
 ip address 10.2.1.1 255.0.0.0
 tag-switching ip
!
interface GigabitEthernet0/4/7
 ip address 10.2.2.1 255.0.0.0
 duplex auto
 speed auto
 tag-switching ip
!
router ospf 1
 log-adjacency-changes
```

```
 network 10.1.2.2 255.0.0.0 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 100
 neighbor R peer-group
 neighbor R remote-as 1
 neighbor R update-source Loopback0
 neighbor R route-reflector-client
 neighbor 10.0.0.4 peer-group R
 neighbor 10.0.0.5 peer-group R
 !
 address-family vpnv4
  neighbor R activate
  neighbor R route-reflector-client
  neighbor R send-community extended
  neighbor 10.1.0.4 peer-group R
  neighbor 10.1.0.5 peer-group R
  exit-address-family
```

Example: Configuration for Autonomous System 1 ASBR1

The following example shows how to configure ASBR1 in AS1 in a confederation topology:

```
ip cef
!
interface Loopback0
 ip address 10.0.0.4 255.0.0.0
!
interface GigabitEthernet0/0/0
 ip address 10.2.1.40 255.255.255.0
 tag-switching ip
!
interface GigabitEthernet0/5/3
 no ip address
 no atm scrambling cell-payload
 no atm ilmi-keepalive
!
interface GigabitEthernet0/5/3.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 pvc 1/100
!
router ospf 1
 log-adjacency-changes
 redistribute connected subnets
 network 10.0.0.3 255.0.0.0 area 0
!
router bgp 1
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 bgp confederation identifier 100
 bgp confederation peers 1
 neighbor R peer-group
 neighbor R remote-as 1
 neighbor R update-source Loopback0
 neighbor 10.0.0.2 remote-as 2
 neighbor 10.0.0.2 next-hop-self
 neighbor 10.0.0.2 peer-group R
 no auto-summary
 !
 address-family vpnv4
  neighbor R activate
  neighbor R send-community extended
  neighbor 10.0.0.2 activate
  neighbor 10.0.0.2 next-hop-self
  neighbor 10.0.0.2 send-community extended
  neighbor 10.0.0.2 peer-group R
```

```
 no auto-summary
 exit-address-family
```

Example: Configuration for Autonomous System 2 ASBR2

The following example shows how to configure ASBR2 in AS2 in a confederation topology:

```
ip cef
!
ip vrf V1
 rd 2:103
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 10.0.0.3 255.0.0.0
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
 ip address 10.0.0.3 255.0.0.0
!
interface GigabitEthernet0/4/7
 no ip address
 encapsulation frame-relay
 load-interval 30
 no fair-queue
 clockrate 2000000
!
interface GigabitEthernet0/4/7.2 point-to-point
 ip unnumbered Loopback0
 ip router isis
 tag-switching ip
 frame-relay interface-dlci 23
!
interface GigabitEthernet0/5/3
 no ip address
 atm clock INTERNAL
 no atm scrambling cell-payload
 no atm ilmi-keepalive
!
interface GigabitEthernet0/5/3.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 pvc 1/100
!
router isis
 net aa.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 10.0.0.1 remote-as 1
 neighbor 10.0.0.1 next-hop-self
 neighbor 10.0.0.8 remote-as 2
 neighbor 10.0.0.8 update-source Loopback0
 neighbor 10.0.0.8 next-hop-self
 !
 address-family ipv4 vrf V1
  redistribute connected
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor 10.0.0.1 activate
  neighbor 10.0.0.1 next-hop-self
  neighbor 10.0.0.1 send-community extended
```

```
 neighbor 10.0.0.8 activate
 neighbor 10.0.0.8 next-hop-self
 neighbor 10.0.0.8 send-community extended
 exit-address-family
```

Example: Configuration for Autonomous System 2 P2

The following example shows how to configure P2 in AS2 in a confederation topology:

```
ip cef
!
ip vrf V1
 rd 2:108
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 10.0.0.8 255.0.0.0
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
 ip address 10.0.0.8 255.0.0.0
!
interface GigabitEthernet0/0/0
 ip address 10.9.1.2 255.0.0.0
 ip router isis
 tag-switching ip
!
interface GigabitEthernet0/5/3
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
!
interface GigabitEthernet0/5/3.1 point-to-point
 ip unnumbered Loopback0
 ip router isis
 tag-switching ip
 frame-relay interface-dlci 23
!
router isis
 net aa.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
 neighbor R update-source Loopback0
 neighbor R route-reflector-client
 neighbor 10.0.0.3 peer-group R
 neighbor 10.0.0.9 peer-group R
 !
 address-family ipv4 vrf V1
  redistribute connected
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor R activate
  neighbor R route-reflector-client
  neighbor R send-community extended
  neighbor 10.0.0.3 peer-group R
  neighbor 10.0.0.9 peer-group R
  exit-address-family
```

Example: Configuration for Autonomous System 2 PE2

The following example shows how to configure PE2 in AS2 in a confederation topology:

```
ip cef
!
ip vrf V1
 rd 2:109
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 10.0.0.9 255.0.0.0
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
 ip address 10.0.0.9 255.0.0.0
!
interface GigabitEthernet0/0/4
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
 no fair-queue
 clockrate 2000000
!
interface GigabitEthernet0/0/4.1 point-to-point
 description Bethel
 ip vrf forwarding V1
 ip unnumbered Loopback1
 frame-relay interface-dlci 24
!
interface GigabitEthernet0/4/7
 ip address 10.9.1.1 255.0.0.0
 ip router isis
 tag-switching ip
!
router ospf 10 vrf V1
 log-adjacency-changes
 redistribute bgp 2 subnets
 network 10.0.0.2 255.0.0.0 area 0
!
router isis
 net aa.0002.0000.0000.0009.00
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 100
 neighbor 10.0.0.8 remote-as 2
 neighbor 10.0.0.8 update-source Loopback0
 !
 address-family ipv4 vrf V1
  redistribute connected
  redistribute ospf 10
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor 10.0.0.8 activate
  neighbor 10.0.0.8 send-community extended
  exit-address-family
```

Example: Configuration for Autonomous System 2 CE2

The following example shows how to configure CE2 in VPN1 in a confederation topology:

```
interface Loopback0
 ip address 10.0.0.11 255.0.0.0
!
interface GigabitEthernet0/0/7
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 2000000
!
interface GigabitEthernet0/0/7.1 point-to-point
ip unnumbered Loopback0
 frame-relay interface-dlci 24
!
router ospf 1
 network 10.0.1.2 255.0.0.0 area 0
```

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 1700	*Assigned Numbers*
RFC 1966	*BGP Route Reflection: An Alternative to Full Mesh IBGP*
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

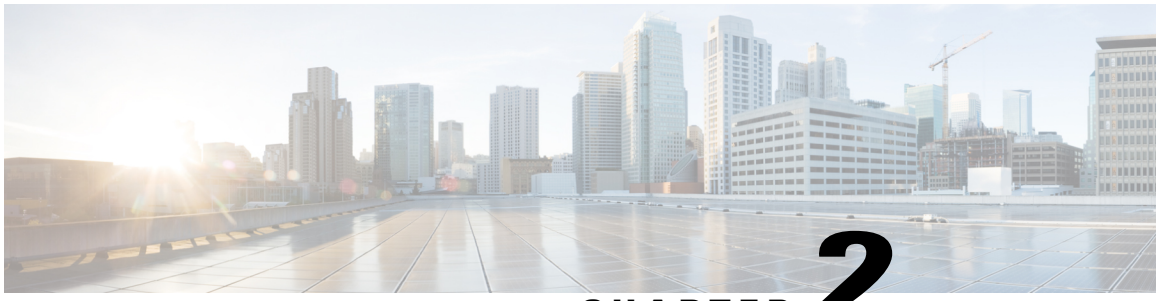
Feature Information for MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

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 VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

Feature Name	Releases	Feature Information
MPLS VPN Interautonomous System Support	Cisco IOS XE Release 3.7S	The MPLS VPN Interautonomous System Support feature enables an MPLS VPN to span service providers and autonomous systems. This feature explains how to configuring the Inter-AS using the ASBRs to exchange VPN-IPv4 Addresses. In Cisco IOS XE Release 3.7S, support was added for the Cisco ASR 903 Router. This feature uses no new or modified commands.



CHAPTER 2

MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

The MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels feature allows a Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) to span service providers and autonomous systems. This module explains how to configure an MPLS VPN Inter-AS network so that the Autonomous System Boundary Routers (ASBRs) exchange IPv4 routes with MPLS labels of the provider edge (PE) routers. Route reflectors (RRs) exchange VPN-IPv4 routes by using multihop, multiprotocol, external Border Gateway Protocol (eBGP).

- Finding Feature Information, page 33
- Prerequisites for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels, page 34
- Restrictions for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels, page 35
- Information About MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels, page 35
- How to Configure MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels, page 38
- Configuration Examples for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels, page 53
- Additional References, page 66
- Feature Information for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels, page 68

Finding Feature Information

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

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 Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

The network must be properly configured for MPLS VPN operation before you configure MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels.

The table below lists the Cisco 12000 series line card support in Cisco IOS S releases.

Table 2: Cisco 12000 Series Line Card Support in Cisco IOS S Releases

Type	Line Cards	Cisco IOS Release Supported
ATM	4-Port OC-3 ATM 1-Port OC-12 ATM 4-Port OC-12 ATM 8-Port OC-3 ATM	12.0(22)S 12.0(23)S 12.0(27)S
Channelized interface	2-Port CHOC-3 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	12.0(22)S 12.0(23)S 12.0(27)S
Electrical interface	6-Port DS3 12-Port DS3 6-Port E3 12-Port E3	12.0(22)S 12.0(23)S 12.0(27)S
Ethernet	3-Port GbE	12.0(23)S 12.0(27)S

Type	Line Cards	Cisco IOS Release Supported
Packet over SONET (POS)	4-Port OC-3 POS 8-Port OC-3 POS 16-Port OC-3 POS 1-Port OC-12 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	12.0(22)S 12.0(23)S 12.0(27)S

Restrictions for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

- For networks configured with eBGP multihop, you must configure a label switched path (LSP) between nonadjacent routers.
- The physical interfaces that connect the BGP speakers must support Cisco Express Forwarding or distributed Cisco Express Forwarding and MPLS.

Information About MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

MPLS VPN Inter-AS Introduction

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.

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.

Benefits of MPLS VPN Inter-AS

An MultiprotocolLabel Switching (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 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. Previously, MPLS VPN could traverse only a single Border Gateway Protocol (BGP) autonomous system service provider backbone. This 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. An autonomous system can be divided 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 Network Layer Reachability Information (NLRI) between the subautonomous systems that form the confederation.

Information About Using MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

This feature can configure a MPLS VPN Inter-AS network so that the ASBRs exchange IPv4 routes with MPLS labels of the PE routers. RRs exchange VPN-IPv4 routes by using multihop, multiprotocol, External Border Gateway Protocol (eBGP). This method of configuring the Inter-AS system is often called MPLS VPN Inter-AS--IPv4 BGP Label Distribution.

Benefits of MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

An Inter-AS system can be configured so that the ASBRs exchange the IPv4 routes and MPLS labels has the following benefits:

- Saves the ASBRs from having to store all the VPN-IPv4 routes. Using the route reflectors to store the VPN-IPv4 routes and forward them to the PE routers results in improved scalability compared wtih configurations where the ASBR holds all of the VPN-IPv4 routes and forwards the routes based on VPN-IPv4 labels.
- Simplifies the configuration at the border of the network by having the route reflectors hold the VPN-IPv4 routes.
- Enables a non-VPN core network to act as a transit network for VPN traffic. You can transport IPv4 routes with MPLS labels over a non-MPLS VPN service provider.

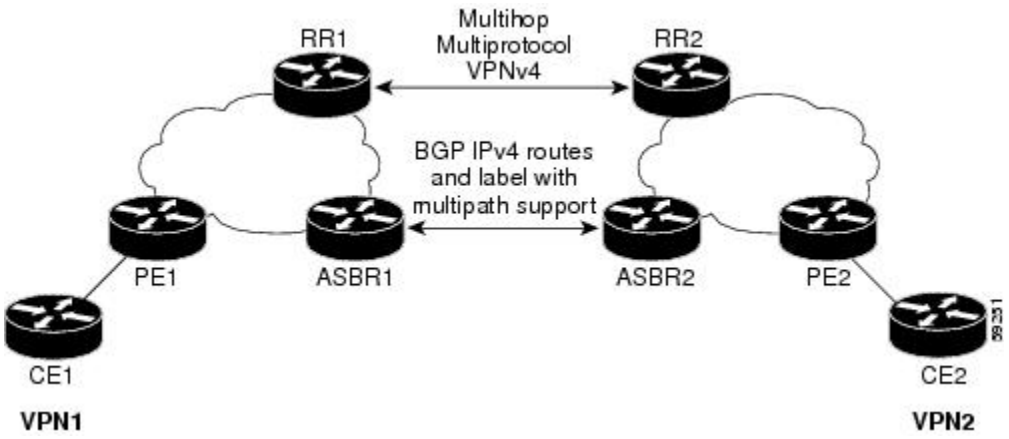
- Eliminates the need for any other label distribution protocol between adjacent LSRs. If two adjacent label switch routers (LSRs) are also BGP peers, BGP can handle the distribution of the MPLS labels. No other label distribution protocol is needed between the two LSRs.

How the Inter-AS Works When ASBRs Exchange IPv4 Routes with MPLS Labels

A VPN service provider network to exchange IPv4 routes with MPLS labels can be configured. The VPN service provider network can be configured as follows:

- Route reflectors exchange VPN-IPv4 routes by using multihop, multiprotocol eBGP. This configuration also preserves the next-hop information and the VPN labels across the autonomous systems.
- A local PE router (for example, PE1 in the figure below) needs to know the routes and label information for the remote PE router (PE2). This information can be exchanged between the PE routers and ASBRs in one of two ways:
  - Internal Gateway Protocol (IGP) and Label Distribution Protocol (LDP): The ASBR can redistribute the IPv4 routes and MPLS labels it learned from eBGP into IGP and LDP and vice versa.
  - Internal Border Gateway Protocol (iBGP) IPv4 label distribution:The ASBR and PE router can use direct iBGP sessions to exchange VPN-IPv4 and IPv4 routes and MPLS labels.

Alternatively, the route reflector can reflect the IPv4 routes and MPLS labels learned from the ASBR to the PE routers in the VPN. This is accomplished by the ASBR exchanging IPv4 routes and MPLS labels with the route reflector. The route reflector also reflects the VPN-IPv4 routes to the PE routers in the VPN. For example, in VPN1 of the figure below, RR1 reflects to PE1 the VPN-IPv4 routes it learned and IPv4 routes and MPLS labels learned from ASBR1. Using the route reflectors to store the VPN-IPv4 routes and forward them through the PE routers and ASBRs allows for a scalable configuration.



BGP Routing Information

BGP routing information includes the following items:

- A network number (prefix), which is the IP address of the destination.

- Autonomous system path, which is a list of the other autonomous systems through which a route passes on its way to the local router. The first autonomous system in the list is closest to the local router; the last autonomous system in the list is farthest from the local router and usually the autonomous system where the route began.
- Path attributes, which provide other information about the autonomous system path, for example, the next hop.

Types of BGP Messages and MPLS Labels

MPLS labels are included in the update messages that a router sends. Routers exchange the following types of BGP messages:

- Keepalive messages--Routers exchange keepalive messages to determine if a neighboring router is still available to exchange routing information. The router sends these messages at regular intervals. (Sixty seconds is the default for Cisco routers.) The keepalive message does not contain routing data; it contains only a message header.
- Notification messages--When a router detects an error, it sends a notification message.
- Open messages--After a router establishes a TCP connection with a neighboring router, the routers exchange open messages. This message contains the number of the autonomous system to which the router belongs and the IP address of the router that sent the message.
- Update messages--When a router has a new, changed, or broken route, it sends an update message to the neighboring router. This message contains the NLRI, which lists the IP addresses of the usable routes. The update message includes any routes that are no longer usable. The update message also includes path attributes and the lengths of both the usable and unusable paths. Labels for VPN-IPv4 routes are encoded in the update message as specified in RFC 2858. The labels for the IPv4 routes are encoded in the update message as specified in RFC 3107.

How BGP Sends MPLS Labels with Routes

When BGP (eBGP and iBGP) distributes a route, it can also distribute an MPLS label that is mapped to that route. The MPLS label mapping information for the route is carried in the BGP update message that contains the information about the route. If the next hop is not changed, the label is preserved.

When you issue the **neighbor send-label** command on both BPG routers, the routers advertise to each other that they can then send MPLS labels with the routes. If the routers successfully negotiate their ability to send MPLS labels, the routers add MPLS labels to all outgoing BGP updates.

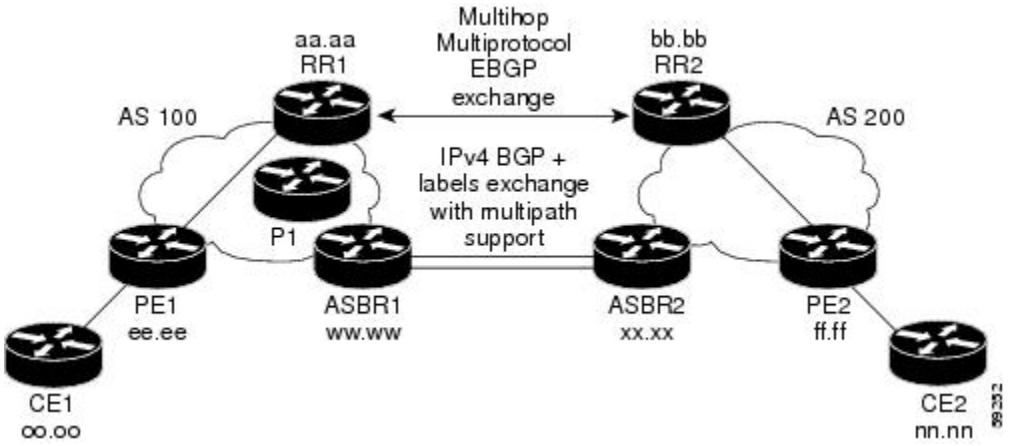
How to Configure MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

To configure MPLS VPN Inter-AS with ASBRs exchanging IPv4 routes and MPLS labels, perform the tasks in the following sections:

The figure below shows the following sample configuration:

- The configuration consists of two VPNs.

- The ASBRs exchange the IPv4 routes with MPLS labels.
- The route reflectors exchange the VPN-IPv4 routes using multihop MPLS eBGP.
- The route reflectors reflect the IPv4 and VPN-IPv4 routes to the other routers in their autonomous system.



Configuring the ASBRs to Exchange IPv4 Routes and MPLS Labels

Perform this task to configure the ASBRs to exchange IPv4 routes and MPLS labels. This configuration procedure uses ASBR1 as an example.

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 ipv4** [**multicast** | **unicast** | **mdt** | **vrf** *vrf-name*]
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
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.

	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 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 hh.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 5	**address-family ipv4** [**multicast** | **unicast** | **mdt** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4`	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv4 address prefixes. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **mdt** keyword specifies an IPv4 multicast distribution tree (MDT) address family session. • The **vrf** *vrf-name* keyword and argument specify the name of the VPN routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor hh.0.0.1 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 7	**neighbor***ip-address* **send-label** **Example:** `Router(config-router-af)# neighbor hh.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.

	Command or Action	Purpose
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-af)# end`	(Optional) Exits to privileged EXEC mode.

Configuring the Route Reflectors to Exchange VPN-IPv4 Routes

Perform this task to enable the route reflectors to exchange VPN-IPv4 routes by using multihop, multiprotocol eBGP.

This procedure also specifies that the next hop information and the VPN label are to be preserved across the autonomous systems. This procedure uses RR1 as an example of the route reflector.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]
6. **address-family vpnv4** [**unicast**]
7. **neighbor** {*ip-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *peer-group-name*} **next-hop unchanged**
9. **exit-address-family**
10. **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	**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. The autonomous system number identifies RR1 to routers in other autonomous systems.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor bb.bb.bb.bb 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 5	**neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop** [*ttl*] **Example:** `Router(config-router)# neighbor bb.bb.bb.bb ebgp-multihop 255`	Accepts and attempts BGP connections to external peers residing on networks that are not directly connected. • 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 *ttl* argument specifies the time-to-live in the range from 1 to 255 hops.
Step 6	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP sessions, that use standard VPNv4 address prefixes. • The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **activate**	Enables the exchange of information with a neighboring router. • The *ip-address* argument specifies the IP address of the neighbor.

	Command or Action	Purpose
	Example: `Router(config-router-af)# neighbor bb.bb.bb.bb activate`	• The *peer-group-name* argument specifies the name of a BGP peer group.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **next-hop unchanged** **Example:** `Router(config-router-af)# neighbor ip-address next-hop unchanged`	Enables an eBGP multihop peer to propagate the next hop unchanged. • The *ip-address* argument specifies the IP address of the next hop. • The *peer-group-name* argument specifies the name of a BGP peer group that is the next hop.
Step 9	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 10	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Configuring the Route Reflector to Reflect Remote Routes in Its Autonomous System

Perform this task to enable the RR to reflect the IPv4 routes and labels learned by the ASBR to the PE routers in the autonomous system.

This is accomplished by making the ASBR and PE router route reflector clients of the RR. This procedure also explains how to enable the RR to reflect the VPN-IPv4 routes.

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* **activate**
6. **neighbor** *ip-address* **route-reflector-client**
7. **neighbor** *ip-address* **send-label**
8. **exit-address-family**
9. **address-family vpnv4** [**unicast**]
10. **neighbor** {*ip-address* | *peer-group-name*} **activate**
11. **neighbor** *ip-address* **route-reflector-client**
12. **exit-address-family**
13. **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	**address-family ipv4** [**multicast**	**unicast**	**vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4`	Enters address family configuration mode for configuring routing sessions, such as BGP sessions, that use standard IPv4 address prefixes. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes.

	Command or Action	Purpose
		• The **vrf** *vrf-name* keyword and argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor ee.ee.ee.ee 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 6	**neighbor** *ip-address* **route-reflector-client** **Example:** `Router(config-router-af)# neighbor ee.ee.ee.ees route-reflector-client`	Configures the router as a BGP route reflector and configures the specified neighbor as its client. • The *ip-address* argument specifies the IP address of the BGP neighbor being configured as a client.
Step 7	**neighbor** *ip-address* **send-label** **Example:** `Router(config-router-af)# neighbor ee.ee.ee.ee 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 address family configuration mode.
Step 9	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP sessions, that use standard VPNv4 address prefixes. • The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 10	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor ee.ee.ee.ee 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.

	Command or Action	Purpose
Step 11	**neighbor** *ip-address* **route-reflector-client** **Example:** `Router(config-router-af)# neighbor ee.ee.ee.ee route-reflector-client`	Enables the RR to pass iBGP routes to the neighboring router.

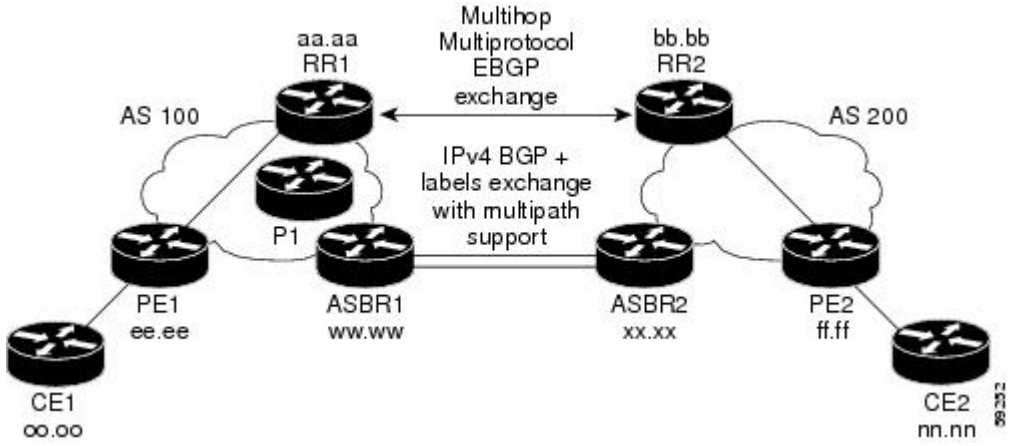
Step 12	**exit-address-family**
	
	Example:
	`Router(config-router-af)#`
	`exit-address-family`
		Exits address family configuration mode.

Step 13	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Verifying the MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels Configuration

If you use ASBRs to distribute the IPv4 labels and route reflectors to distribute the VPN-IPv4 routes, use the following procedures to help verify the configuration:

The figure below shows the configuration that is referred to in the next several sections.



Verifying the Route Reflector Configuration

Perform this task to verify the route reflector configuration.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name* } [**summary**] [**labels**]
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	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name* } [**summary**] [**labels**] **Example:** `Router# show ip bgp vpnv4 all summary`	(Optional) Displays VPN address information from the BGP table. • Use the **all** and **summary** keywords to verify that a multihop, multiprotocol eBGP session exists between the route reflectors and that the VPNv4 routes are being exchanged between the route reflectors. The last two lines of the command output show the following information: • • Prefixes are being learned from PE1 and then passed to RR2. • Prefixes are being learned from RR2 and then passed to PE1. • Use the **all** and **labels** keywords to verify that the route reflectors exchange VPNv4 label information.
Step 3	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Verifying that CE1 Can Communicate with CE2

Perform this task to verify that router CE1 has NLRI for router CE2.

SUMMARY STEPS

1. **enable**
2. **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | [**protocol** [*protocol-id*]] | [**list** [*access-list-number* | *access-list-name*]
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	**show ip route** [*ip-address* [*mask*] [**longer-prefixes**]]	[**protocol** [*protocol-id*]]	[**list** [*access-list-number*	*access-list-name*] **Example:** `Router# show ip route nn.nn.nn.nn`	Displays the current state of the routing table. • Use the *ip-address* argument to verify that CE1 has a route to CE2. • Use this command to verify the routes learned by CE1. Make sure that the route for CE2 is listed.
Step 3	**disable** **Example:** `Router# disable`	(Optional) Exits to privileged EXEC mode.

Verifying that PE1 Can Communicate with CE2

Perform this task to verify that router PE1 has NLRI for router CE2.

SUMMARY STEPS

1. **enable**
2. **show ip route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*] [*tag*] [*output-modifiers*]] [**list** *number* [*output-modifiers*]] [**profile**] [**static** [[]] [**summary***output-modifiers*]] [**supernets-only** [*output-modifiers*]] [**traffic-engineering** [*output-modifiers*]]
3. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [*ip-prefix* | *length* [**longer-prefixes**] [*output-modifiers*]]] [**network-address** *mask*]] **longer-prefixes** [*output-modifiers*]] [**cidr-only**] [**community**] [**community-list**] [**dampened-paths**] [**filter-list**] [**flap-statistics**] [**inconsistent-as**] [**neighbors**] [**paths** [*line*]] [**peer-group**] [**quote-regexp**] [**regexp**] [**summary**] [**tags**]
4. **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]
5. **show mpls forwarding-table** [{*network* {*mask* | *length*} | **labels** *label* [-*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
6. **show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**]
7. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**]
8. **disable**

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 route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*] [*tag*] [*output-modifiers*]] [**list** *number* [*output-modifiers*]] [**profile**] [**static** [[]] [**summary***output-modifiers*]] [**supernets-only** [*output-modifiers*]] [**traffic-engineering** [*output-modifiers*]] **Example:** `Router# show ip route vrf vpn1 nn.nn.nn.nn`	(Optional) Displays the IP routing table associated with a VRF. • Use this command to verify that router PE1 learns routes from router CE2 (nn.nn.nn.nn).
Step 3	**show ip bgp vpnv4** {**all**	**rd** *route-distinguisher*	**vrf** *vrf-name*} [*ip-prefix*	*length* [**longer-prefixes**] [*output-modifiers*]]] [**network-address** *mask*]] **longer-prefixes** [*output-modifiers*]] [**cidr-only**] [**community**] [**community-list**] [**dampened-paths**] [**filter-list**] [**flap-statistics**] [**inconsistent-as**] [**neighbors**] [**paths** [*line*]] [**peer-group**] [**quote-regexp**] [**regexp**] [**summary**] [**tags**] **Example:** `Router# show ip bgp vpnv4 vrf vpn1 nn.nn.nn.nn`	(Optional) Displays VPN address information from the BGP table. • Use the **vrf** or **all** keyword to verify that router PE2 is the BGP next-hop to router CE2.

	Command or Action	Purpose
	Example: `Router# show ip bgp vpnv4 all nn.nn.nn.nn`	
Step 4	**show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**] **Example:** `Router# show ip cef vrf vpn1 nn.nn.nn.nn`	(Optional) Displays entries in the Forwarding Information Base (FIB) or displays a summary of the FIB. • Use this command to verify that the Cisco Express Forwarding entries are correct.
Step 5	**show mpls forwarding-table** [{*network* {*mask* | *length*} | **labels** *label* [-*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:** `Router# show mpls forwarding-table`	(Optional) Displays the contents of the MPLS LFIB. • Use this command to verify the IGP label for the BGP next hop router (autonomous system boundary).
Step 6	**show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**] **Example:** `Router# show ip bgp ff.ff.ff.ff`	(Optional) Displays entries in the BGP routing table. • Use the **show ip bgp** command to verify the label for the remote egress PE router (PE2).
Step 7	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**] **Example:** `Router# show ip bgp vpnv4 all labels`	(Optional) Displays VPN address information from the BGP table. • Use the **all** and **summary** keywords to verify the VPN label of CE2, as advertised by PE2.
Step 8	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Verifying that PE2 Can Communicate with CE2

Perform this task to ensure that PE2 can access CE2.

SUMMARY STEPS

1. **enable**
2. **show ip route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*] [*tag*] [*output-modifiers*]] [**list** *number* [*output-modifiers*]] [**profile**] [**static** [*output-modifiers*]] [**summary**[*output-modifiers*]] [**supernets-only** [*output-modifiers*]] [**traffic-engineering** [*output-modifiers*]]
3. **show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [*-label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
4. **show ip bgp vpnv4** { **all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**]
5. **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]
6. **disable**

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 route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*] [*tag*] [*output-modifiers*]] [**list** *number* [*output-modifiers*]] [**profile**] [**static** [*output-modifiers*]] [**summary**[*output-modifiers*]] [**supernets-only** [*output-modifiers*]] [**traffic-engineering** [*output-modifiers*]] **Example:** `Router# show ip route vrf vpn1 nn.nn.nn.nn`	(Optional) Displays the IP routing table associated with a VRF. • Use this command to check the VPN routing and forwarding table for CE2. The output provides next-hop information.
Step 3	**show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask*	*length*}	**labels** *label* [*-label*]	**interface** *interface*	**next-hop** *address*	**lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:** `Router# show mpls forwarding-table vrf vpn1 nn.nn.nn.nn`	(Optional) Displays the contents of the LFIB. • Use the **vrf** keyword to check the VPN routing and forwarding table for CE2. The output provides the label for CE2 and the outgoing interface.
Step 4	**show ip bgp vpnv4** { **all**	**rd** *route-distinguisher*	**vrf** *vrf-name*} [**summary**] [**labels**] **Example:** `Router# show ip bgp vpnv4 all labels`	(Optional) Displays VPN address information from the BGP table. • Use the **all** and **labels** keywords to check the VPN label for CE2 in the multiprotocol BGP table.
Step 5	**show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]	(Optional) Displays entries in the FIB or displays a summary of the FIB.

	Command or Action	Purpose
	Example: `Router# show ip cef vpn1 nn.nn.nn.nn`	• Use this command to check the Cisco Express Forwarding entry for CE2. The command output shows the local label for CE2 and the outgoing interface.
Step 6	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Verifying the ASBR Configuration

Perform this task to verify that the ASBRs exchange IPv4 routes with MPLS labels or IPv4 routes without labels as prescribed by a route map.

Verifying the ASBR Configuration

SUMMARY STEPS

1. **enable**
2. **show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**]
3. **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]
4. **disable**

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 bgp** [*network*] [*network-mask*] [**longer-prefixes**] **Example:** `Router# show ip bgp ff.ff.ff.ff`	(Optional) Displays entries in the BGP routing table. • Use this command to check that: • ASBR1 receives an MPLS label for PE2 from ASBR2. • ASBR1 receives IPv4 routes for RR2 without labels from ASBR2. • ASBR2 distributes an MPLS label for PE2 to ASBR1.

	Command or Action	Purpose
		• ASBR2 does not distribute a label for RR2 to ASBR1.

Step 3	**show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]
	
	Example:
	`Router# show ip cef ff.ff.ff.ff`
	
	Example:
	`Router# show ip cef bb.bb.bb.bb`
		(Optional) Displays entries in the FIB or displays a summary of the FIB.
		• Use thiscommand from ASBR1 and ASBR2 to check that:
		• The Cisco Express Forwarding entry for PE2 is correct.
		• The Cisco Express Forwarding entry for RR2 is correct.

Step 4	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Configuration Examples for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

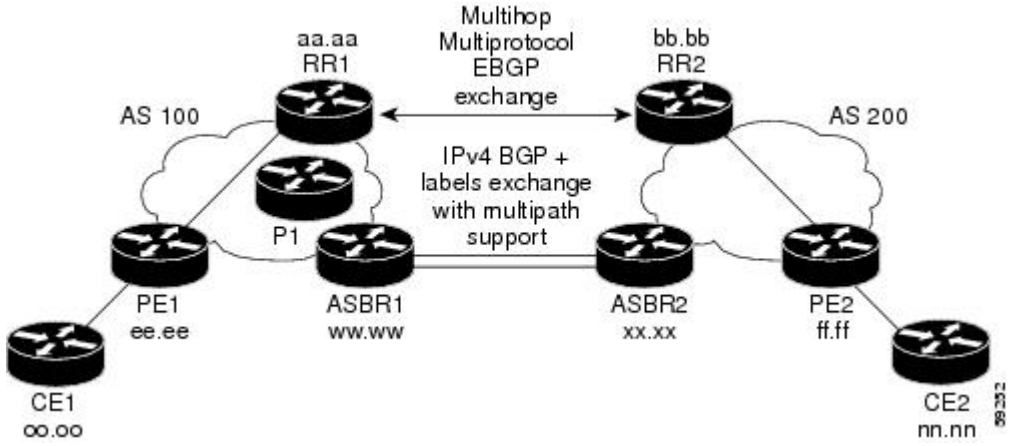
Configuring MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels over an MPLS VPN Service Provider Examples

Configuration examples for Inter-AS using BGP to distribute routes and MPLS labels over an MPLS VPN service provider included in this section are as follows:

The figure below shows two MPLS VPN service providers. The service provider distributes the VPN-IPv4 routes between the route reflectors. The MPLS VPN service providers distribute the IPv4 routes with MPLS labels between the ASBRs.

The configuration example shows the following two techniques you can use to distribute the VPN-IPv4 routes and the IPv4 routes with MPLS labels of the remote RRs and PEs to the local RRs and PEs:

- Autonomous system 100 uses the RRs to distribute the VPN-IPv4 routes learned from the remote RRs. The RRs also distribute the remote PE address and label learned from ASBR1 using IPv4 labels.
- In Autonomous system 200, the IPv4 routes that ASBR2 learned are redistributed into IGP.



Route Reflector 1 Configuration Example (MPLS VPN Service Provider)

The configuration example for RR1 specifies the following:

- RR1 exchanges VPN-IPv4 routes with RR2 using multiprotocol, multihop eBGP.
- The VPN-IPv4 next-hop information and the VPN label are preserved across the autonomous systems.
- RR1 reflects to PE1:
  - The VPN-IPv4 routes learned from RR2
  - The IPv4 routes and MPLS labels learned from ASBR1

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address aa.aa.aa.aa 255.255.255.255
!
interface Ethernet0/3
 ip address dd.0.0.2 255.0.0.0
!
router ospf 10
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 network aa.aa.aa.aa 0.0.0.0 area 100
 network dd.0.0.0 0.255.255.255 area 100
!
router bgp 100
 bgp cluster-id 1
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor ee.ee.ee.ee remote-as 100
 neighbor ee.ee.ee.ee update-source Loopback0
 neighbor ww.ww.ww.ww remote-as 100
 neighbor ww.ww.ww.ww update-source Loopback0
 neighbor bb.bb.bb.bb remote-as 200
 neighbor bb.bb.bb.bb ebgp-multihop 255
 neighbor bb.bb.bb.bb update-source Loopback0
 no auto-summary
 !
address-family ipv4
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee route-reflector-client                !IPv4+labels session to PE1
 neighbor ee.ee.ee.ee send-label
```

```
 neighbor ww.ww.ww.ww activate
 neighbor ww.ww.ww.ww route-reflector-client                !IPv4+labels session to ASBR1
 neighbor ww.ww.ww.ww send-label
 no neighbor bb.bb.bb.bb activate
 no auto-summary
 no synchronization
 exit-address-family
 !
address-family vpnv4
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee route-reflector-client                !VPNv4 session with PE1
 neighbor ee.ee.ee.ee send-community extended
 neighbor bb.bb.bb.bb activate
 neighbor bb.bb.bb.bb next-hop-unchanged                    !MH-VPNv4 session with RR2
 neighbor bb.bb.bb.bb send-community extended                 !with next hop unchanged

 exit-address-family
!
ip default-gateway 3.3.0.1
no ip classless
!
snmp-server engineID local 00000009020000D0584B25C0
snmp-server community public RO
snmp-server community write RW
no snmp-server ifindex persist
snmp-server packetsize 2048
!
end
```

ASBR1 Configuration Example (MPLS VPN Service Provider)

ASBR1 exchanges IPv4 routes and MPLS labels with ASBR2.

In this example, ASBR1 uses route maps to filter routes:

- A route map called OUT specifies that ASBR1 should distribute the PE1 route (ee.ee) with labels and the RR1 route (aa.aa) without labels.
- A route map called IN specifies that ASBR1 should accept the PE2 route (ff.ff) with labels and the RR2 route (bb.bb) without labels.

```
ip subnet-zero
mpls label protocol ldp
!
interface Loopback0
 ip address ww.ww.ww.ww 255.255.255.255
!
interface Ethernet0/2
 ip address hh.0.0.2 255.0.0.0
!
interface Ethernet0/3
 ip address dd.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls 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
```

Route Reflector 2 Configuration Example (MPLS VPN Service Provider)

RR2 exchanges VPN-IPv4 routes with RR1 through multihop, multiprotocol eBGP. This configuration also specifies that the next-hop information and the VPN label are preserved across the autonomous systems:

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address bb.bb.bb.bb 255.255.255.255
!
interface Serial1/1
 ip address ii.0.0.2 255.0.0.0
!
router ospf 20
 log-adjacency-changes
 network bb.bb.bb.bb 0.0.0.0 area 200
 network ii.0.0.0 0.255.255.255 area 200
!
router bgp 200
 bgp cluster-id 1
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor aa.aa.aa.aa remote-as 100
```

```
 neighbor aa.aa.aa.aa ebgp-multihop 255
 neighbor aa.aa.aa.aa update-source Loopback0
 neighbor ff.ff.ff.ff remote-as 200
 neighbor ff.ff.ff.ff update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor aa.aa.aa.aa activate
 neighbor aa.aa.aa.aa next-hop-unchanged              !Multihop VPNv4 session with RR1
 neighbor aa.aa.aa.aa send-community extended              !with next-hop-unchanged
 neighbor ff.ff.ff.ff activate
 neighbor ff.ff.ff.ff route-reflector-client          !VPNv4 session with PE2
 neighbor ff.ff.ff.ff send-community extended
 exit-address-family
!
ip default-gateway 3.3.0.1
no ip classless
!
end
```

ASBR2 Configuration Example (MPLS VPN Service Provider)

ASBR2 exchanges IPv4 routes and MPLS labels with ASBR1. However, in contrast to ASBR1, ASBR2 does not use the RR to reflect IPv4 routes and MPLS labels to PE2. ASBR2 redistributes the IPv4 routes and MPLS labels learned from ASBR1 into IGP. PE2 can now reach these prefixes.

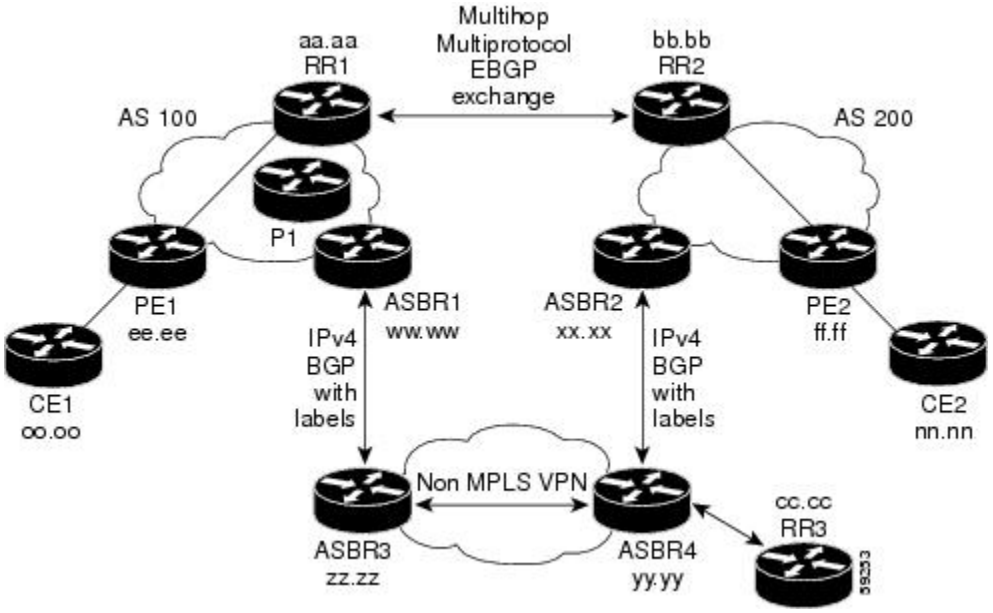
```
ip subnet-zero
ip cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address xx.xx.xx.xx 255.255.255.255
!
interface Ethernet1/0
 ip address hh.0.0.1 255.0.0.0
!
interface Ethernet1/2
 ip address jj.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
 !
router ospf 20
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 redistribute bgp 200 subnets           ! Redistributing the routes learned from
 passive-interface Ethernet1/0               ! ASBR1(eBGP+labels session) into IGP
 network xx.xx.xx.xx 0.0.0.0 area 200         ! so that PE2 will learn them
 network jj..0.0 0.255.255.255 area 200
 !
router bgp 200
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor bb.bb.bb.bb remote-as 200
 neighbor bb.bb.bb.bb update-source Loopback0
 neighbor hh.0.0.2 remote-as 100
 no auto-summary
 !
address-family ipv4
 redistribute ospf 20                         ! Redistributing IGP into BGP
 neighbor hh.0.0.2 activate                   ! so that PE2 & RR2 loopbacks
 neighbor hh.0.0.2 advertisement-interval 5   ! will get into the BGP-4 table.
 neighbor hh.0.0.2 route-map IN in
 neighbor hh.0.0.2 route-map OUT out
 neighbor hh.0.0.2 send-label
 neighbor kk.0.0.2 activate
 neighbor kk.0.0.2 advertisement-interval 5
 neighbor kk.0.0.2 route-map IN in
```

```
 neighbor kk.0.0.2 route-map OUT out
 neighbor kk.0.0.2 send-label
 no auto-summary
 no synchronization
 exit-address-family
!
address-family vpnv4
 neighbor bb.bb.bb.bb activate
 neighbor bb.bb.bb.bb send-community extended
 exit-address-family
 !
ip default-gateway 3.3.0.1
ip classless
!
access-list 1 permit ff.ff.ff.ff log          !Setting up the access lists
access-list 2 permit ee.ee.ee.ee log
access-list 3 permit bb.bb.bb.bb log
access-list 4 permit aa.aa.aa.aa log
route-map IN permit 11                       !Setting up the route maps
 match ip address 2
 match mpls-label
!
route-map IN permit 12
 match ip address 4
!
route-map OUT permit 10
 match ip address 1
 set mpls-label
!
route-map OUT permit 13
 match ip address 3
end
```

Configuring MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels over a Non-MPLS VPN Service Provider Examples

Configuration examples for Inter-AS using BGP to distribute routes and MPLS labels over a non MPLS VPN service provider included in this section are as follows:

The figure below shows two MPLS VPN service providers that are connected through a non MPLS VPN service provider. The autonomous system in the middle of the network is configured as a backbone autonomous system that uses LDP or Tag Distribution Protocol (TDP) to distribute MPLS labels. Traffic engineering tunnels can also be used instead of TDP or LDP to build the LSP across the non MPLS VPN service provider.



Route Reflector 1 Configuration Example (Non-MPLS VPN Service Provider)

The configuration example for RR1 specifies the following:

- RR1 exchanges VPN-IPv4 routes with RR2 using multiprotocol, multihop eBGP.
- The VPN-IPv4 next-hop information and the VPN label are preserved across the autonomous systems.
- RR1 reflects to PE1:
  - The VPN-IPv4 routes learned from RR2
  - The IPv4 routes and MPLS labels learned from ASBR1

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address aa.aa.aa.aa 255.255.255.255
!
interface Serial1/2
 ip address dd.0.0.2 255.0.0.0
 clockrate 124061
!
router ospf 10
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 network aa.aa.aa.aa 0.0.0.0 area 100
 network dd.0.0.0 0.255.255.255 area 100
!
router bgp 100
 bgp cluster-id 1
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor ee.ee.ee.ee remote-as 100
 neighbor ee.ee.ee.ee update-source Loopback0
 neighbor ww.ww.ww.ww remote-as 100
 neighbor ww.ww.ww.ww update-source Loopback0
 neighbor bb.bb.bb.bb remote-as 200
```

```
 neighbor bb.bb.bb.bb ebgp-multihop 255
 neighbor bb.bb.bb.bb update-source Loopback0
 no auto-summary
 !
address-family ipv4
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee route-reflector-client               !IPv4+labels session to PE1
 neighbor ee.ee.ee.ee send-label
 neighbor ww.ww.ww.ww activate
 neighbor ww.ww.ww.ww route-reflector-client               !IPv4+labels session to ASBR1
 neighbor ww.ww.ww.ww send-label
 no neighbor bb.bb.bb.bb activate
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee route-reflector-client              !VPNv4 session with PE1
 neighbor ee.ee.ee.ee send-community extended
 neighbor bb.bb.bb.bb activate
 neighbor bb.bb.bb.bb next-hop-unchanged                   !MH-VPNv4 session with RR2
 neighbor bb.bb.bb.bb send-community extended                with next-hop-unchanged
 exit-address-family
!
ip default-gateway 3.3.0.1
no ip classless
!
snmp-server engineID local 00000009020000D0584B25C0
snmp-server community public RO
snmp-server community write RW
no snmp-server ifindex persist
snmp-server packetsize 2048
!
end
```

ASBR1 Configuration Example (Non-MPLS VPN Service Provider)

ASBR1 exchanges IPv4 routes and MPLS labels with ASBR2.

In this example, ASBR1 uses route maps to filter routes:

- A route map called OUT specifies that ASBR1 should distribute the PE1 route (ee.ee) with labels and the RR1 route (aa.aa) without labels.
- A route map called IN specifies that ASBR1 should accept the PE2 route (ff.ff) with labels and the RR2 route (bb.bb) without labels.

```
ip subnet-zero
ip cef distributed
mpls label protocol ldp
!
interface Loopback0
 ip address ww.ww.ww.ww 255.255.255.255
!
interface Serial3/0/0
 ip address kk.0.0.2 255.0.0.0
 ip route-cache distributed
!
interface Ethernet0/3
 ip address dd.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
!
router ospf 10
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
```

```
 passive-interface Serial3/0/0
 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 kk.0.0.1 remote-as 200
 no auto-summary
!
 address-family ipv4
 redistribute ospf 10                      ! Redistributing IGP into BGP
 neighbor aa.aa.aa.aa activate             ! so that PE1 & RR1 loopbacks
 neighbor aa.aa.aa.aa send-label           ! get into BGP table
 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 specified in route map IN
 neighbor kk.0.0.1 route-map OUT out  ! Distributing routes specified 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
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
 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
```

Route Reflector 2 Configuration Example (Non-MPLS VPN Service Provider)

RR2 exchanges VPN-IPv4 routes with RR1 using multihop, multiprotocol eBGP. This configuration also specifies that the next-hop information and the VPN label are preserved across the autonomous systems:

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address bb.bb.bb.bb 255.255.255.255
!
interface Serial1/1
 ip address ii.0.0.2 255.0.0.0
!
router ospf 20
 log-adjacency-changes
 network bb.bb.bb.bb 0.0.0.0 area 200
 network ii.0.0.0 0.255.255.255 area 200
!
router bgp 200
```

```
 bgp cluster-id 1
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor aa.aa.aa.aa remote-as 100
 neighbor aa.aa.aa.aa ebgp-multihop 255
 neighbor aa.aa.aa.aa update-source Loopback0
 neighbor ff.ff.ff.ff remote-as 200
 neighbor ff.ff.ff.ff update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor aa.aa.aa.aa activate
 neighbor aa.aa.aa.aa next-hop-unchanged              !MH vpnv4 session with RR1
 neighbor aa.aa.aa.aa send-community extended              !with next-hop-unchanged
 neighbor ff.ff.ff.ff activate
 neighbor ff.ff.ff.ff route-reflector-client          !vpnv4 session with PE2
 neighbor ff.ff.ff.ff send-community extended
 exit-address-family
!
ip default-gateway 3.3.0.1
no ip classless
!
end
```

ASBR2 Configuration Example (Non-MPLS VPN Service Provider)

ASBR2 exchanges IPv4 routes and MPLS labels with ASBR1. However, in contrast to ASBR1, ASBR2 does not use the RR to reflect IPv4 routes and MPLS labels to PE2. ASBR2 redistributes the IPv4 routes and MPLS labels learned from ASBR1 into IGP. PE2 can now reach these prefixes.

```
ip subnet-zero
ip cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address xx.xx.xx.xx 255.255.255.255
!
interface Ethernet0/1
 ip address qq.0.0.2 255.0.0.0
!
interface Ethernet1/2
 ip address jj.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
 !
router ospf 20
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 redistribute bgp 200 subnets            !redistributing the routes learned from
 passive-interface Ethernet0/1                !ASBR2 (eBGP+labels session) into IGP
 network xx.xx.xx.xx 0.0.0.0 area 200         !so that PE2 will learn them
 network jj.0.0.0 0.255.255.255 area 200
 !
router bgp 200
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor bb.bb.bb.bb remote-as 200
 neighbor bb.bb.bb.bb update-source Loopback0
 neighbor qq.0.0.1 remote-as 100
 no auto-summary
!
address-family ipv4                          ! Redistributing IGP into BGP
                                                          redistribute ospf 20
                    ! so that PE2 & RR2 loopbacks
 neighbor qq.0.0.1 activate                  !  will get into the BGP-4 table
 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
 no auto-summary
 no synchronization
 exit-address-family
 !
address-family vpnv4
 neighbor bb.bb.bb.bb activate
 neighbor bb.bb.bb.bb send-community extended
 exit-address-family
 !
ip default-gateway 3.3.0.1
ip classless
!
access-list 1 permit ff.ff.ff.ff log
access-list 2 permit ee.ee.ee.ee log
access-list 3 permit bb.bb.bb.bb log
access-list 4 permit aa.aa.aa.aa log
!
route-map IN permit 11
 match ip address 2
 match mpls-label
!
route-map IN permit 12
 match ip address 4
!
route-map OUT permit 10
 match ip address 1
 set mpls-label
!
route-map OUT permit 13
 match ip address 3
!
end
```

ASBR3 Configuration Example (Non-MPLS VPN Service Provider)

ASBR3 belongs to a non MPLS VPN service provider. ASBR3 exchanges IPv4 routes and MPLS labels with ASBR1. ASBR3 also passes the routes learned from ASBR1 to ASBR4 through RR3.

Note Do not redistribute eBGP routes learned into iBGP if you are using iBGP to distribute the routes and labels. This is not a supported configuration.

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address yy.yy.yy.yy 255.255.255.255
interface Hssi4/0
 ip address mm.0.0.0.1 255.0.0.0
 mpls ip
 hssi internal-clock
 !
interface Serial5/0
 ip address kk.0.0.1 255.0.0.0
 load-interval 30
 clockrate 124061
!
router ospf 30
log-adjacency-changes
auto-cost reference-bandwidth 1000
redistribute connected subnets
network yy.yy.yy.yy 0.0.0.0 area 300
network mm.0.0.0 0.255.255.255 area 300
!
```

```
router bgp 300
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor cc.cc.cc.cc remote-as 300
 neighbor cc.cc.cc.cc update-source Loopback0
 neighbor kk.0.0.2 remote-as 100
 no auto-summary
 !
 address-family ipv4
 neighbor cc.cc.cc.cc activate            ! iBGP+labels session with RR3
 neighbor cc.cc.cc.cc send-label
 neighbor kk.0.0.2 activate               ! eBGP+labels session with ASBR1
 neighbor kk.0.0.2 advertisement-interval 5
 neighbor kk.0.0.2 send-label
 neighbor kk.0.0.2 route-map IN in
 neighbor kk.0.0.2 route-map OUT out
 no auto-summary
 no synchronization
 exit-address-family
!
ip classless
!
access-list 1 permit ee.ee.ee.ee log
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
 match ip address 1
  match mpls-label
!
route-map IN permit 11
   match ip address 3
!
route-map OUT permit 12
 match ip address 2
  set mpls-label
!
route-map OUT permit 13
   match ip address 4
!
ip default-gateway 3.3.0.1
ip classless
!
end
```

Route Reflector 3 Configuration Example (Non-MPLS VPN Service Provider)

RR3 is a non MPLS VPN RR that reflects IPv4 routes with MPLS labels to ASBR3 and ASBR4.

```
ip subnet-zero
mpls label protocol ldp
mpls traffic-eng auto-bw timers
no mpls ip
!
interface Loopback0
 ip address cc.cc.cc.cc 255.255.255.255
!
interface POS0/2
 ip address pp.0.0.1 255.0.0.0
 crc 16
 clock source internal
!
router ospf 30
 log-adjacency-changes
 network cc.cc.cc.cc 0.0.0.0 area 300
 network pp.0.0.0 0.255.255.255 area 300
!
router bgp 300
 bgp log-neighbor-changes
```

```
 neighbor zz.zz.zz.zz remote-as 300
 neighbor zz.zz.zz.zz update-source Loopback0
 neighbor yy.yy.yy.yy remote-as 300
 neighbor yy.yy.yy.yy update-source Loopback0
 no auto-summary
 !
address-family ipv4
 neighbor zz.zz.zz.zz activate
 neighbor zz.zz.zz.zz route-reflector-client
 neighbor zz.zz.zz.zz send-label               ! iBGP+labels session with ASBR3
 neighbor yy.yy.yy.yy activate
 neighbor yy.yy.yy.yy route-reflector-client
 neighbor yy.yy.yy.yy send-label               ! iBGP+labels session with ASBR4
 no auto-summary
 no synchronization
 exit-address-family
!
ip default-gateway 3.3.0.1
ip classless
!
end
```

ASBR4 Configuration Example (Non-MPLS VPN Service Provider)

ASBR4 belongs to a non MPLS VPN service provider. ASBR4 and ASBR3 exchange IPv4 routes and MPLS labels by means of RR3.

Note Do not redistribute eBGP routes learned into iBGP if you are using iBGP to distribute the routes and labels. This is not a supported configuration.

```
ip subnet-zero
ip cef distributed
!
interface Loopback0
 ip address zz.zz.zz.zz 255.255.255.255
!
interface Ethernet0/2
 ip address qq.0.0.1 255.0.0.0
!
interface POS1/1/0
 ip address pp.0.0.2 255.0.0.0
 ip route-cache distributed
 !
interface Hssi2/1/1
 ip address mm.0.0.2 255.0.0.0
 ip route-cache distributed
 mpls label protocol ldp
 mpls ip
 hssi internal-clock
!
router ospf 30
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Ethernet0/2
 network zz.zz.zz.zz 0.0.0.0 area 300
 network pp.0.0.0 0.255.255.255 area 300
 network mm.0.0.0 0.255.255.255 area 300
 !
router bgp 300
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor cc.cc.cc.cc remote-as 300
 neighbor cc.cc.cc.cc update-source Loopback0
 neighbor qq.0.0.2 remote-as 200
```

```
 no auto-summary
 !
 address-family ipv4
 neighbor cc.cc.cc.cc activate
 neighbor cc.cc.cc.cc send-label
 neighbor qq.0.0.2 activate
 neighbor qq.0.0.2 advertisement-interval 5
 neighbor qq.0.0.2 send-label
 neighbor qq.0.0.2 route-map IN in
 neighbor qq.0.0.2 route-map OUT out
 no auto-summary
 no synchronization
 exit-address-family
!
ip classless
!
access-list 1 permit ff.ff.ff.ff log
access-list 2 permit ee.ee.ee.ee log
access-list 3 permit bb.bb.bb.bb log
access-list 4 permit aa.aa.aa.aa log
!
route-map IN permit 10
 match ip address 1
  match mpls-label
!
route-map IN permit 11
   match ip address 3
!
route-map OUT permit 12
 match ip address 2
  set mpls-label
!
route-map OUT permit 13
   match ip address 4
!
ip default-gateway 3.3.0.1
ip classless
!
end
```

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 1700	*Assigned Numbers*
RFC 1966	*BGP Route Reflection: An Alternative to Full Mesh IBGP*
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

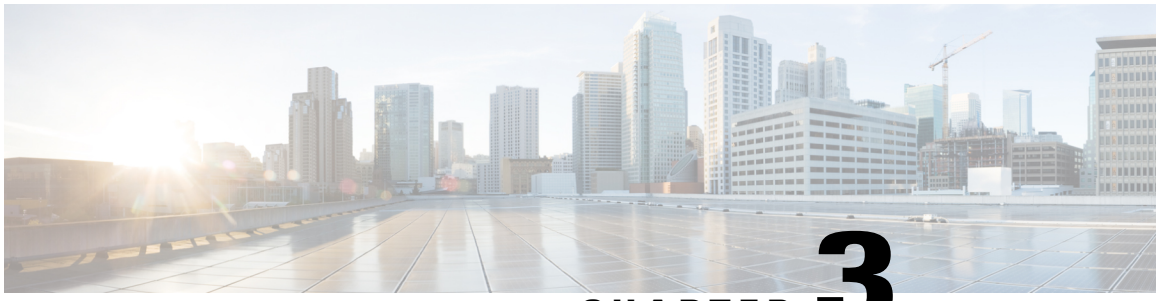
Feature Information for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

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 MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

Feature Name	Releases	Feature Configuration Information
MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels	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 Cisco IOS XE Release 2.5	This module explains how to configure an MPLS VPN Inter-AS network so that the ASBRs exchange IPv4 routes with MPLS labels of the provider edge (PE) routers. Route reflectors (RRs) exchange VPN-IPv4 routes by using multihop, multiprotocol, external Border Gateway Protocol (eBGP). In Cisco IOS XE Release 2.5, this feature was implemented on the Cisco ASR 1000 Series Routers. This feature uses no new or modified commands.



CHAPTER 3

MPLS VPN Multipath Support for Inter-AS VPNs

The MPLS VPN Multipath Support for Inter-AS VPNs feature supports Virtual Private Network (VPN)v4 multipath for Autonomous System Boundary 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.

- Finding Feature Information, page 69
- Restrictions for MPLS VPN Multipath Support for Inter-AS VPNs, page 69
- Information About MPLS VPN Multipath Support for Inter-AS VPNs, page 70
- How to Configure MPLS VPN Multipath Support for Inter-AS VPNs, page 71
- Configuration Examples for MPLS VPN Multipath Support for Inter-AS VPNs, page 78
- Additional References, page 85
- Feature Information for MPLS VPN Multipath Support for Inter-AS VPNs, page 87
- Glossary, page 87

Finding Feature Information

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

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 MPLS VPN Multipath Support for Inter-AS VPNs

The following restrictions apply to configuring multipath load sharing for Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) interautonomous system (Inter-AS) Autonomous System Boundary Routers (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.

Information About MPLS VPN Multipath Support for Inter-AS VPNs

Load Sharing with MPLS VPN Inter-AS ASBRs

Before the MPLS VPN Interautonomous System Support feature, if multiple paths existed across Autonomous System Boundary Routers (ASBRs), the Border Gateway Protocol (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 Virtual Private Network (VPN)-IPv4 prefixes.

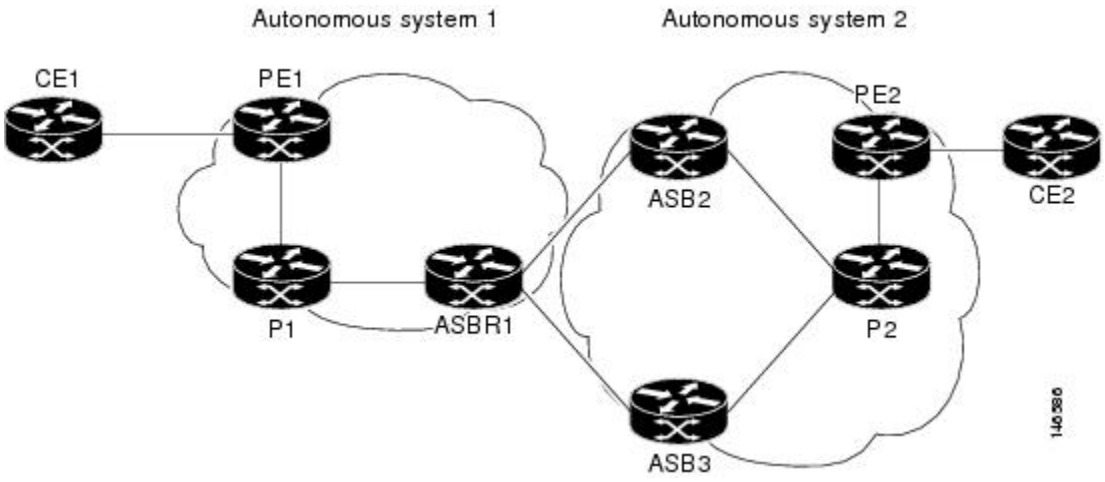
The MPLS VPN Multipath Support for Inter-AS VPNs feature applies to ASBRs that do not have a VPN routing and forwarding (VRF) instance configuration. BGP installs a number of learned VPN-IPv4 prefixes into the Multiprotocol Label Switching (MPLS) forwarding table (LFIB). VPN-IPv4 entries in the LFIB consist of the Route Distinguisher (RD) and the IPv4 prefix and are called VPNv4 entries.

The 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. This 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 software.

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 device CE1 to CE2 using the two available links—ASBR2 and ASBR3.

Figure 9: 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.

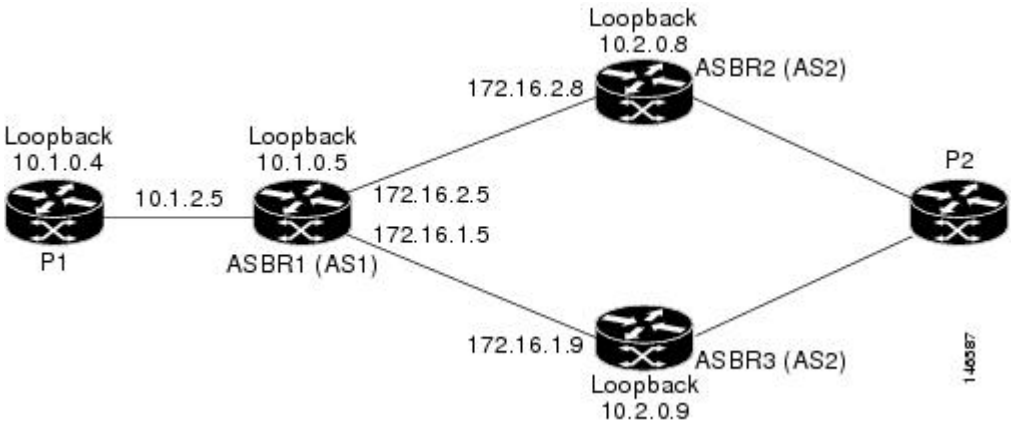
How to Configure MPLS VPN Multipath Support for Inter-AS VPNs

Configuring eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs

Perform this task to configure the external Border Gateway Protocol (eBGP) multipath load sharing for MPLS VPN Inter-AS ASBRs exchanging Virtual Private Network (VPN)-IPv4 routes. This allows for more efficient use of the label switched paths (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 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.

Figure 10: eBGP Multipath Configuration for Three VPN-IPv4 ASBRs



The configurations in the figure above is used as an example for this task.

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 1`	Configures an eBGP routing process and places the device in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP devices and tags the routing information passed along. The range is 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**no bgp default route-target filter** **Example:** `Device(config-router)# no bgp default route-target filter`	Disables BGP route-target community filtering. • All received VPN-IPv4 routes are accepted by the configured device. Accepting VPN-IPv4 routes is the desired behavior for a device configured as an ASBR.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 10.1.0.4 remote-as 1`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • *ip-address*—IP address of the neighbor. • *peer-group-name*—Name of a BGP peer group. • *as-number*—The autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 10.1.0.4 update-source loopback 0`	Allows BGP sessions to use any operational interface for TCP connections. • *ip-address*—IP address of the neighbor. • *peer-group-name*—Name of a BGP peer group. • *interface-typeinterface-number*—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.

	Command or Action	Purpose
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self** **Example:** `Device(config-router)# neighbor 10.1.0.4 next-hop-self`	Configures the device as the next hop for a BGP neighbor or peer group. • *ip-address*—IP address of the BGP neighbor. • *peer-group-name*—Name of a BGP peer group.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 172.16.1.9 remote-as 2`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • *ip-address*—IP address of the neighbor. • *peer-group-name*—Name of a BGP peer group. • *as-number*—Autonomous system to which the neighbor belongs.
Step 9	Repeat Step 8 for each BGP neighbor.	—
Step 10	**address-family vpnv4** [**unicast**] **Example:** `Device(config-router)# address-family vpnv4`	Enters address family configuration mode. • **unicast**—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:** `Device(config-router-af)# neighbor 10.1.0.4 activate`	Enables the exchange of information with a neighboring device. • *ip-address*—IP address of the neighbor. • *peer-group-name*—Name of a BGP peer group.
Step 12	**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self** **Example:** `Device(config-router-af)# neighbor 10.1.0.4 next-hop-self`	Configures the device as the next hop for a BGP neighbor or peer group. • *ip-address*—IP address of the BGP neighbor. • *peer-group-name*—Name of a BGP peer group.
Step 13	**neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **Example:** `Device(config-router-af)# neighbor 10.1.0.4 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • *ip-address*—IP address of the neighboring device. • *peer-group-name*—Name of a BGP peer group. • **both**—Specifies that both standard and extended communities will be sent. • **standard**—Specifies that only standard communities will be sent.

	Command or Action	Purpose
		• **extended—**Specifies that only extended communities will be sent.
Step 14	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Device(config-router-af)# neighbor 172.16.1.9 activate`	Enables the exchange of information with a BGP neighbor. • *ip-address***—**IP address of the neighboring device. • *peer-group-name***—**Name of a BGP peer group. • *ipv6-address***—**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 15	**neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **Example:** `Device(config-router-af)# neighbor 172.16.1.9 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • *ip-address***—**IP address of the neighboring device. • *peer-group-name***—**Name of a BGP peer group. • **both—**Specifies that both standard and extended communities will be sent. • **standard—**Specifies that only standard communities will be sent. • **extended—**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:** `Device(config-router-af)# maximum-paths 2`	Configures the maximum number of parallel routes that an IP routing protocol will install into the routing table. • *number-paths***—**Number of routes to install to the routing table.
Step 18	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits from address family configuration mode.
Step 19	**end** **Example:** `Device(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Example

The following example shows the configuration for external Border Gateway Protocol (eBGP) multipath for VPNv4 sessions on the ASBR1 device:

```
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 the external Border Gateway Protocol (eBGP) multipath load sharing for Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) interautonomous system (Inter-AS) Autonomous System Boundary Routers (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**

Enables privileged EXEC mode. Enter your password if required.

Example:

```
Device> enable
Device#
```

Step 2 **show ip bgp vpnv4 all** [**summary**]
Verifies that all peers are up.

Example:

```
Device# 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**
Verifies that BGP has paths from both remote ASBRs.

Example:

```
Device# 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*]
Verifies that paths are marked as multipath.

Example:

```
Device# 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**

Verifies that MPLS forwarding is properly set up and counters are increasing when traffic is present.

Example:

```
Device# 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**

Exits to user EXEC mode. For example:

Example:

```
Device# exit
Device>
```

Configuration Examples for MPLS VPN Multipath Support for Inter-AS VPNs

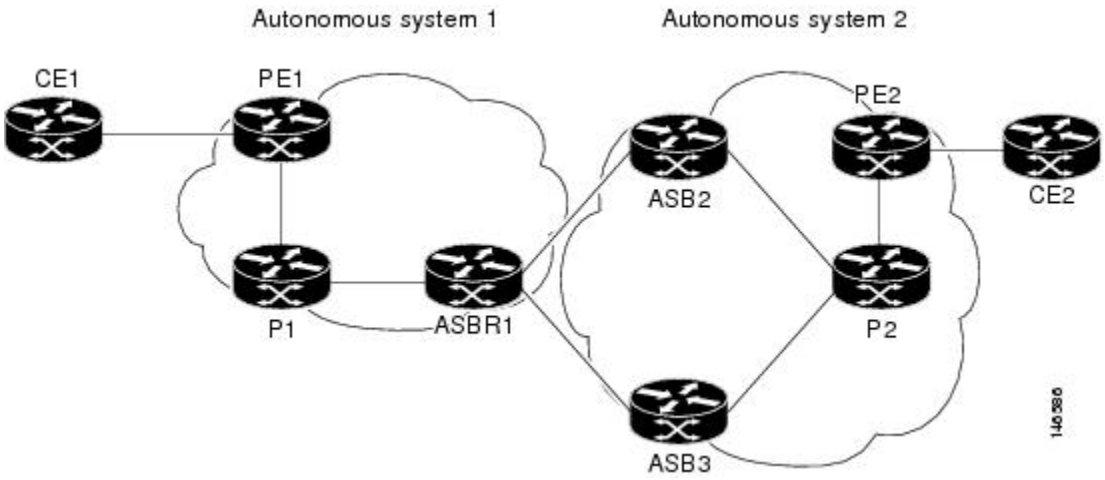
Example: Configuring eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs

This section includes examples that show how to configure the external Border Gateway Protocol (eBGP) multipath load sharing for Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) interautonomous system (Inter-AS) Autonomous System Boundary Routers (ASBRs) that exchange VPN-IPv4 routes.

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 devices are route reflectors.
- ASBR1 and ASBR2 are configured with the **neighbor next-hop-self** command for the internal BGP (iBGP) neighbors.
- ASBR1 and ASBR2 are configured with the **maximum paths** commands to set up eBGP multipath load sharing.

Figure 11: 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 devices.

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 CE1

The following example shows how to configure CE1 in VPN1:

```
!
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
```

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 PE1

The following example shows how to configure PE1 in autonomous system 1:

```
!
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
```

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 P1

The following example shows how to configure P1 in autonomous system 1:

```
!
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
```

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 1 ASBR1

The following example shows how to configure ASBR1 in autonomous system 1:

```
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
```

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 ASBR2

The following example shows how to configure ASBR2 in autonomous system 2:

```
!
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
```

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 ASBR3

The following example shows how to configure ASBR3 in autonomous system 2:

```
!
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
```

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 P2

The following example shows how to configure P2 in autonomous system 2:

```
!
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
!
```

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 PE2

The following example shows how to configure PE2 in autonomous system 2:

```
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
```

Example: Multipath Support for Inter-AS VPNs Configuration for Autonomous System 2 CE2

The following example shows how to configure CE2 in VPN1:

```
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
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference

Related Topic	Document Title
Configuration tasks for basic MPLS VPNs	"Configuring MPLS VPNs" module in the *MPLS: Layer 3 VPNs Configuration Guide*
Configuration tasks for MPLS VPN Inter-AS system exchanging IPv4 routes and MPLS labels	"MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels" module in the *MPLS: Layer 3 VPNs Inter-AS and CSC Configuration Guide*
Information about monitoring MPLS VPNs with MIBs	"MPLS VPN SNMP MIB Notifications" module in the *MPLS: Embedded Management and MIBs Configuration Guide*

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 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 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 Multipath Support for Inter-AS 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 MPLS VPN Multipath Support for Inter-AS VPNs

Feature Name	Releases	Feature Information
MPLS VPN Multipath Support for Inter-AS VPNs	12.2(30)S 12.2(33)SRA 12.2(33)SXH 12.4(20)T	The MPLS VPN Multipath Support for Inter-AS VPNs feature supports Virtual Private Network (VPN)v4 multipath for Autonomous System Boundary 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. No commands were introduced or modified.

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).

CE device—customer edge device. A device that is part of a customer network and that interfaces to a provider edge (PE) device. CE devices do not recognize associated MPLS VPNs.

eBGP —exterior Border Gateway Protocol. A BGP between devices located within different autonomous systems. When two devices, located in different autonomous systems, are more than one hop away from one another, the eBGP session between the two devices is considered a multihop BGP.

iBGP —interior Border Gateway Protocol. A BGP between devices within the same autonomous system.

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.

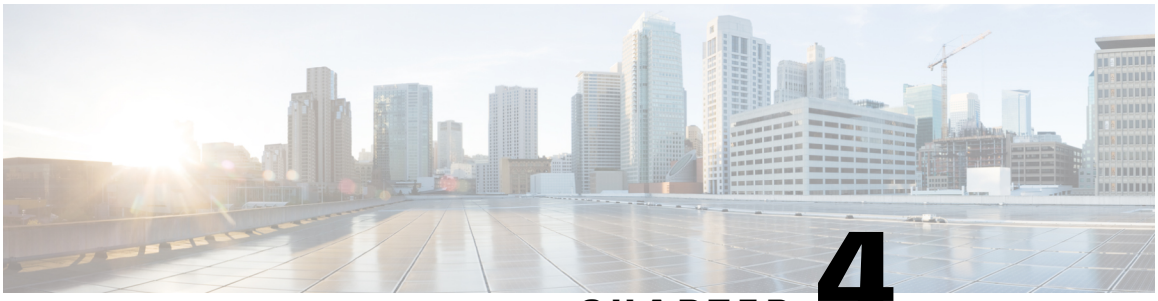
MPLS —Multiprotocol Label Switching. The name of the IETF working group responsible for label switching, and the name of the label switching approach it has standardized.

PE device—provider edge device. A device that is part of a service provider's network. It is connected to a customer edge (CE) device and all MPLS VPN processing occurs in the PE device.

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) device. 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.



CHAPTER 4

MPLS VPN--Inter-AS Option AB

The MPLS VPN--Inter-AS Option AB feature combines the best functionality of an Inter-AS Option (10) A and Inter-AS Option (10) B network to allow a Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) service provider to interconnect different autonomous systems to provide VPN services. These networks are defined in RFC 4364 section 10 "Multi-AS Backbones," subsections a and b, respectively.

When different autonomous systems are interconnected in an MPLS VPN--Inter-AS Option AB configuration, the entire network configuration is scaled and simplified, and maintains IP quality of service (QoS) functions between Autonomous System Boundary Router (ASBR) peers.

In an Inter-AS Option A network, ASBR peers are connected by multiple subinterfaces with at least one interface VPN that spans the two autonomous systems. These ASBRs associate each subinterface with a VPN routing and forwarding (VRF) instance and a Border Gateway Protocol (BGP) session to signal unlabeled IP prefixes. As a result, traffic between the back-to-back VRFs is IP. In this scenario, the VPNs are isolated from each other, and because the traffic is IP, QoS mechanisms that operate on IP traffic can be applied to achieve customer Service Level Agreements (SLAs). The downside of this configuration is that one BGP session is needed for each subinterface (and at least one subinterface for each VPN), which causes scalability concerns as this network grows.

In an Inter-AS Option B network, ASBR peers are connected by one or more subinterfaces that are enabled to receive MPLS traffic. A Multiprotocol Border Gateway Protocol (MP-BGP) session is used to distribute labeled VPN prefixes between the ASBR. As a result, the traffic that flows between them is labeled. The downside of this configuration is that, because the traffic is MPLS, QoS mechanisms that can be applied only to IP traffic cannot be applied and the VRFs cannot be isolated.

- Finding Feature Information, page 90
- Prerequisites for MPLS VPN--Inter-AS Option AB, page 90
- Restrictions for MPLS VPN--Inter-AS Option AB, page 90
- Information About MPLS VPN--Inter-AS Option AB, page 90
- How to Configure Inter-AS Option AB, page 98
- Configuration Examples for MPLS VPN--Inter-AS Option AB, page 107
- Additional References, page 128
- Feature Information for MPLS VPN--Inter-AS Option AB, page 130
- Glossary, page 131

Finding Feature Information

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

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--Inter-AS Option AB

Follow the appropriate configuration tasks outlined in the following documents:

- Configuring MPLS Layer 3 VPNs
- MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses
- MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

Before configuring the MPLS VPN--Inter-AS Option AB feature, perform these tasks:

- Enable Cisco Express Forwarding, which is required for the MPLS VPN routing and forwarding operation.
- Identify the VPNs for the MPLS VPN--Inter-AS Option AB network and configure the VRFs to which these VPNs belong. These VRFs are used for Inter-AS Option AB connections on the ASBR interface.

Restrictions for MPLS VPN--Inter-AS Option AB

- The In Service Software Upgrade (ISSU) feature can be configured only on the active Route Processor (RP) if the standby RP supports this feature. The ISSU feature can be configured if both the active and standby RP support this feature.
- Carrier Supporting Carrier (CSC) MPLS load-balancing on ASBR Option AB VRF interfaces is not supported.
- VPNv6 is not supported.

Information About MPLS VPN--Inter-AS Option AB

MPLS VPN--Inter-AS Option AB Introduction

MPLS VPN service providers need to interconnect different autonomous systems to provide service for multiple VPN customers. The MPLS VPN--Inter-AS Option AB feature allows the different autonomous systems to interconnect by using a single MP-BGP session in the global routing table to carry control plane traffic. This MP-BGP session signals VPN prefixes between two ASBRs for each VRF instance. The data plane traffic is on a VRF interface. This traffic can either be IP or MPLS.

Note Inter-AS connections can be configured between ASBRs that either have or do not have connections between different providers.

Benefits of MPLS VPN--Inter-AS Option AB

The MPLS VPN--Inter-AS Option AB feature provides the following benefits for service providers:

- Network configuration can be simplified because only one BGP session is configured for each VRF on the ASBR.
- One BGP session reduces CPU utilization.
- Networks can be scaled because a single MP-BGP session, which is enabled globally on the router, reduces the number of sessions required by multiple VPNs, while continuing to keep VPNs isolated and secured from each other.
- IP QoS functions between ASBR peers are maintained for customer SLAs.
- Dataplane traffic is isolated on a per-VRF basis for security purposes.

Option B Style Peering with Shared Link Forwarding

An enhancement to Inter-AS Option AB is the MPLS VPN—Inter-AS Option AB+ feature. This feature addresses the scalability concerns of MPLS VPN—Inter-AS Option A by using a single BGP session in the global routing table to signal VPN prefixes (as described in Inter-AS Option B).

The key difference between Option AB+ and Option B is in the route distribution between ASBRs. In Option AB+, at the ASBR, the route that is imported into the VRF (with the route distinguisher and route targets of the VRF) is distributed to the neighboring ASBR. In Option B, the original pre-import route (with the original RD and RTs) is distributed to the neighboring ASBR and not the imported route.

With Option AB+, the PE and ASBRs deploy MPLS forwarding over a global interface, similar to what is done in Option B, and the signaling is handled by a single MP-eBGP VPNv4 session. The provider edge and ASBRs thus use regular Option B style peering between them. They receive MPLS-VPN traffic over the shared link and forward the traffic as per an IP lookup in the VRF routing table. However, the traffic is MPLS encapsulated, like it is in Option B.

Route Distribution and Packet Forwarding in Non-CSC Networks

The following sections describe MPLS VPN--Inter-AS Option AB operation:

Note All imported routes are accomplished by configuring the appropriate route targets (RTs).

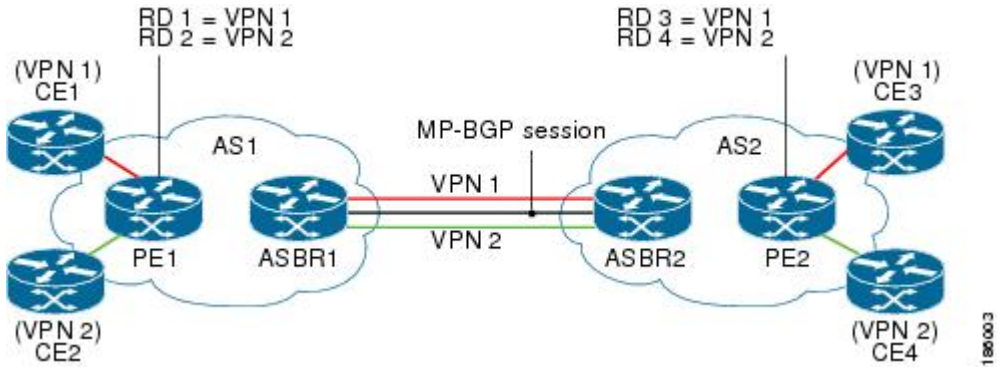
The following attributes describe the topology of the sample MPLS VPN--Inter-AS Option AB network shown in the figure below:

- Customer edge 1 (CE1) and CE3 belong to VPN 1.

- CE2 and CE 4 belong to VPN 2.
- Provider edge 1 (PE1) uses route distinguisher 1 (RD 1) for VPN 1 (VRF 1) and RD 2 for VPN 2 (VRF 2).
- PE2 uses RD 3 for VPN 1 (VRF 1) and RD 4 for VPN 2 (VRF 2).
- ASBR1 has VRF 1 provisioned with RD 5 and VRF 2 provisioned with RD 6.
- ASBR2 has VRF 1 provisioned with RD 7 and VRF 2 provisioned and RD 8.
- ASBR1 and ASBR2 have three links between them:
  - VRF 1
  - VRF 2
  - MP-BGP session

Note The VRFs configured on the ASBRs are called Option AB VRFs. The eBGP peers on the ASBRs are called Option AB Peers.

Figure 12: MPLS VPN Inter-AS Option AB Topology



Route Distribution for VPN 1

A route distinguisher (RD) is an identifier attached to a route that identifies which VPN belongs to each route. Each routing instance must have a unique RD autonomous system associated with it. The RD is used to place a boundary around a VPN so that the same IP address prefixes can be used in different VPNs without having these IP address prefixes overlap.

Note An RD statement is required if the instance type is a VRF.

The following process describes the route distribution process for VPN 1 in the figure above. Prefix “N” is used in this process to indicate the IP address of a VPN.

1 CE1 advertises the prefix N to PE1.

2. PE1 advertises a VPN prefix RD 1:N to ASBR1 through MP internal BGP (iBGP).
3. ASBR1 imports the prefix into VPN 1 and creates a prefix RD 5:N.
4. ASBR1 advertises the imported prefix RD 5:N to ASBR2. ASBR1 sets itself as the next hop for prefix RD 5:N and allocates a local label that is signaled with this prefix.
5. ASBR1 advertises the route with the export RT configured on the VRF rather than the originally received RTs. By default, ASBR1 does not advertise the source prefix RD 1:N to ASBR2. This advertisement is suppressed because the prefix is being imported into an Option AB VRF.

Note In an Option 10B connection, the source prefix can be advertised to another ASBR on which ASBR1 has an Option 10B connection. An ASBR with an Option 10B connection maintains all VPNv4 routes in its BGP table.

1. ASBR2 receives the prefix RD 5:N and imports it into VPN 1 as RD 7:N.
2. ASBR2 advertises the route with the export RT configured on the VRF rather than the originally received RTs.
3. While importing the prefix, ASBR2 sets the next hop of RD 7:N to the ASBR1 interface IP address in VRF 1. The next hop table ID is also set to VRF 1. When installing the MPLS forwarding entry for RD 7:N, by default ASBR2 does not install the outgoing label in the forwarding process. This enables the traffic between the ASBRs to be IP.
4. ASBR2 advertises the imported prefix RD 7:N to PE2. It sets itself as the next hop for this prefix and also allocates a local label that is signaled with the prefix. By default, ASBR2 does not advertise the source prefix RD 5:N to PE2. This advertisement is suppressed because the prefix is being imported into an Option AB VRF.
5. PE2 imports the RD 7:N into VRF 1 as RD 3:N.

Packet Forwarding for VPN 1

The following packet forwarding process works the same as it does in an Option A scenario. The ASBR acts like the PE by terminating the VPN and then forwards its traffic as standard IP packets with no VPN label to the next PE, which in turn repeats the VPN process. Each PE router, therefore, treats the adjacent PE router as a CE router, and the standard Layer 3 MPLS VPN mechanisms are used for route redistribution with each autonomous system; that is, the PEs use external BGP (eBGP) to distribute unlabeled IPv4 addresses to each other.

Note Prefix “N” is used in this process to indicate the IP address of a VPN.

1. CE3 sends a packet destined for N to PE2.
2. PE2 encapsulates the packet with the VPN label allocated by ASBR2 and the Interior Gateway Protocol (IGP) label needed to tunnel the packet to ASBR2.
3. The packet arrives on ASBR2 with the VPN label. ASBR2 removes the VPN label and sends the packet as IP to ASBR1 on the VRF 1 interface.

4 The IP packet arrives at ASBR1 on the VRF 1 interface. ASBR1 then encapsulates the packet with the VPN label allocated by PE1 and the IGP label needed to tunnel the packet to PE1.

5 The packet arrives on PE1 with the VPN label. PE1 disposes the VPN label and forwards the IP packet to CE1.

Route Distribution for VPN 2

The following information describes the route distribution process for VPN 2 in the figure above:

1 CE2 advertises prefix N to PE1, where N is the VPN IP address.

2 PE1 advertises a VPN prefix RD 2:N to ASBR1 through MP-iBGP.

3 ASBR1 imports the prefix into VPN 2 and creates a prefix RD 6:N.

4 ASBR1 advertises the imported prefix RD 6:N to ASBR2. It sets itself as the next hop for this prefix and also allocates a local label that is signaled with the prefix. By default, ASBR1 does not advertise the source prefix RD 2:N to ASBR2. This advertisement is suppressed as the prefix is being imported into an Option AB VRF.

Note In the case of an Option 10B connection, the source prefix can be advertised to another ASBR on which ASBR1 has an Option 10B connection. An ASBR with an Option 10B connection maintains all VPNv4 routes in its BGP table.

1 ASBR2 receives the prefix RD 6:N and imports it into VPN 2 as RD 8:N.

2 While importing the prefix, ASBR2 sets the next hop of RD 8:N to ASBR1s interface address in VRF 2. The next hop table ID is also set to that of VRF 2. While installing the MPLS forwarding entry for RD 8:N, by default ASBR2 does not install the outgoing label in the forwarding process. This enables traffic between the ASBRs to be IP.

3 ASBR2 advertises the imported prefix RD 8:N to PE2. It sets itself as the next hop for this prefix and also allocates a local label that is signaled with the prefix. By default, ASBR2 does not advertise the source prefix RD 6:N to PE2. This advertisement is suppressed because the prefix is being imported into an Option AB VRF.

4 PE2 imports the RD 8:N into VRF 2 as RD 4:N.

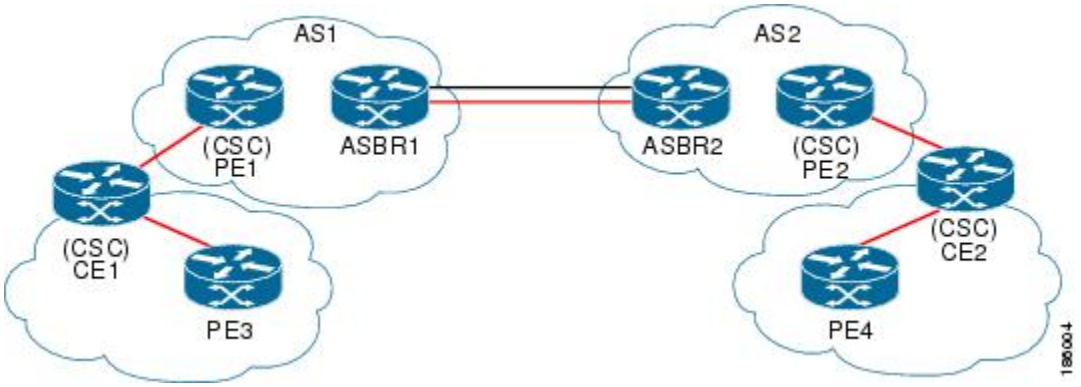
Route Distribution and Packet Forwarding for CSC

The following sections describe MPLS VPN--Inter-AS Option AB operation for a CSC scenario for VPN 1. These sections are similar to those found in Route Distribution and Packet Forwarding in Non-CSC Networks for VPN 1, except for the method in which MPLS labels are handled between the two ASBRs.

Note VPN 2 is not shown or discussed in this section.

The figure below shows how VPN 1 provides VPN service to a small customer carrier that in turn provides a VPN service to its customer. This configuration implies that VPN 1 is used to provide a label switched path (LSP) between the PE (PE 3 and PE 4) loopback interfaces of the small customer carrier.

Figure 13: MPLS VPN Inter-AS Option AB CSC Topology



Note The RD, RT, VRF, and Link provisioning in this section is the same as in the Route Distribution and Packet Forwarding in Non-CSC Networks example for VPN 1.

Route Distribution for VPN 1

The following information describe the route distribution process for VPN 1 in Figure 1 . Prefix “N” is used in these steps to indicate the IP address of a VPN.

1. CE1 advertises PE 3 loopback N to PE1.
2. PE1 advertises a VPN prefix RD 1:N to ASBR1 through MP-iBGP.
3. ASBR1 imports the prefix into VPN 1 and creates a prefix RD 5:N.
4. ASBR1 advertises the imported prefix RD 5:N to ASBR2. It sets itself as the next hop for this prefix and also allocates a local label that is signaled with the prefix.
5. ASBR1 advertises the route with the export RT configured on the VRF rather than the originally received RTs. By default, ASBR1 does not advertise the source prefix RD 1:N to ASBR2. This advertisement is suppressed as the prefix is being imported into an Option AB VRF.

Note In an Option 10B connection, the source prefix can be advertised to another ASBR on which ASBR1 has an Option 10B connection. An ASBR with an Option 10B connection maintains all VPNv4 routes in its BGP table.

1. ASBR2 receives the prefix RD 5:N and imports it into VPN 1 as RD 7:N.
2. ASBR2 advertises the route with the export RT configured on the VRF rather than the originally received RTs.

3 While importing the prefix, ASBR2 sets the next hop of RD 7:N to the ASBR1 interface address in VRF 1. The next hop table ID is also set to that of VRF 1.

Note In a CSC scenario, an outgoing MPLS label can be installed in forwarding by making a configuration change. See the How to Configure Inter-AS Option AB, on page 98.

1 While installing the MPLS forwarding entry for RD 7:N, ASBR2 installs the outgoing label during the forwarding process, which enables the traffic between the ASBRs to be MPLS traffic.

2 ASBR2 advertises the imported prefix RD 7:N to PE2. It sets itself as the next hop for this prefix and also allocates a local label that is signaled with the prefix. By default, ASBR2 does not advertise the source prefix RD 5:N to PE2. This advertisement is suppressed as the prefix is being imported into an Option AB VRF.

3 PE2 imports the RD 7:N into VRF 1 as RD 3:N.

Packet Forwarding for VPN 1

The packet forwarding process shown below works the same as it does in an Option A scenario. See the Route Distribution and Packet Forwarding in Non-CSC Networks section for more information about Option A.

1 PE 4 sends an MPLS packet destined for N to CE2.

2 CE2 swaps the MPLS label and sends a packet destined for N to PE2.

3 PE2 encapsulates the packet with the VPN label allocated by ASBR2 and the IGP label needed to tunnel the packet to ASBR2.

4 The packet arrives on ASBR2 with the VPN label. ASBR2 swaps the received VPN label with the outgoing label received from ASBR1 and sends the MPLS packet on to the VRF 1 interface.

5 The MPLS packet arrives at ASBR1 on the VRF 1 interface. ASBR1 then swaps the received MPLS label with a label stack consisting of the VPN label allocated by PE1 and the IGP label needed to tunnel the packet to PE1.

6 The packet arrives on PE1 with the VPN label. PE1 disposes the VPN label and forwards the MPLS packet to CE1. CE1 in turn swaps the label and forwards the labeled packet to PE 3.

Shared Link Forwarding in Non-CSC Networks

Note All imported routes are accomplished by configuring the appropriate route targets (RTs).

The following attributes describe the sample network topology shown in the "Route Distribution and Packet Forwarding in Non-CSC Networks" section:

- Customer edge 1 (CE1) and CE3 belong to VPN 1.
- CE2 and CE 4 belong to VPN 2.

- Provider edge 1 (PE1) uses route distinguisher 1 (RD 1) for VPN 1 (VRF 1) and RD 2 for VPN 2 (VRF 2).
- PE2 uses RD 3 for VPN 1 (VRF 1) and RD 4 for VPN 2 (VRF 2).
- ASBR1 has VRF 1 provisioned with RD 5 and VRF 2 provisioned with RD 6.
- ASBR2 has VRF 1 provisioned with RD 7 and VRF 2 provisioned and RD 8.
- ASBR1 and ASBR2 have three links between them:
  - VRF 1
  - VRF 2
  - MP-BGP session

Note The VRFs configured on the ASBRs are called Option AB+ VRFs. The eBGP peers on the ASBRs are called Option AB+ Peers.

The following sections describe MPLS VPN—Inter-AS Option AB+ shared link forwarding in a non-CSC network:

Route Distribution for VPN 1

The following process describe the route distribution process for VPN 1 shown in the figure in the "Route Distribution and Packet Forwarding in Non-CSC Networks" section. Prefix “N” is used in these steps to indicate the IP address of a VPN.

1. CE1 advertises PE 3 loopback N to PE1.
2. PE1 advertises a VPN prefix RD 1:N to ASBR1 through MP-iBGP.
3. ASBR1 imports the prefix into VPN 1 and creates a prefix RD 5:N.
4. ASBR1 advertises the imported prefix RD 5:N to ASBR2. ASBR1 sets itself as the next hop for prefix RD 5:N and also allocates a local label that is signaled with this prefix.
5. By default, ASBR1 does not advertise the source prefix RD 1:N to ASBR2. This advertisement is suppressed because the prefix is being imported into an Option AB+ VRF.

Note In an Option 10B connection, the source prefix can be advertised to another ASBR on which ASBR1 has an Option 10B connection. An ASBR with an Option 10B connection maintains all VPNv4 routes in its BGP table.

1. ASBR2 receives the prefix RD 5:N and imports it into VPN 1 as RD 7:N.
2. While importing the prefix, ASBR2 retains the next hop of RD7:N as received in the BGP update from ASBR2. This is the address of ASBR1 shared interface address in the global table. The next hop tableid is also left unchanged and corresponds to that of the global table

3. When installing the MPLS forwarding entry for RD 7:N, ASBR2 installs the outgoing label in the forwarding process. This enables the traffic between the ASBRs to be IP.
4. ASBR2 advertises the imported prefix RD 7:N to PE2. It sets itself as the next hop for this prefix and also allocates a local label that is signaled with the prefix.
5. By default, ASBR2 does not advertise the source prefix RD 5:N to PE2. This advertisement is suppressed because the prefix is being imported into an Option AB+ VRF.
6. PE2 imports the RD 7:N into VRF 1 as RD 3:N.

Packet Forwarding for VPN1

The following packet forwarding process works the same as it does in an Option B scenario.

1. CE3 sends a packet destined for N to PE2.
2. PE2 encapsulates the packet with the VPN label allocated by ASBR2 and the IGP label needed to tunnel the packet to ASBR2.
3. The packet arrives on ASBR2 with the VPN label. ASBR2 swaps the received VPN label with the outgoing label received from ASBR1 and sends the MPLS packet on the global shared link interface.
4. The MPLS packet arrives at ASBR1 on the global shared link interface. ASBR1 then swaps the received MPLS label with a label stack consisting of the VPN label allocated by PE1 and the IGP label needed to tunnel the packet to PE1.
5. The packet arrives on PE1 with the VPN label. PE1 removes the VPN label and forwards the IP packet to CE1.

How to Configure Inter-AS Option AB

The following sections describe how to configure the Inter-AS Option AB feature on an ASBR for either an MPLS VPN or an MPLS VPN that supports CSC:

Note If Inter-AS Option AB is already deployed in your network and you want to do Option B style peering for some prefixes (that is, implement Inter-AS Option AB+), configure the **inter-as-hybrid global** command as described in the "Configuring the Routing Policy for VPNs that Need Inter-AS Connections" section.

Configuring an Inter-AS Option AB Connection

The following sections are required and describe how to configure an Inter-AS Option AB connection on an ASBR:

Note See the Configuring MPLS Layer 3 VPNs feature module for more information on configuring PE and CE routers in an MPLS VPN.

Configuring the VRFs on the ASBR Interface for Each VPN Customer

Use the following steps to configure the VRFs on the ASBR interface for each VPN customer so that these VPNs have connectivity over the MPLS VPN--Inter-AS Option AB network.

Note The **mpls bgp forwarding** command is used only on the ASBR interface for VRFs that support CSC.

Use all of the steps in the following procedure to configure additional VRFs that need to be configured on the ASBR interface and the VRFs that need to be configured on the peer ASBR interface.

SUMMARY STEPS

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

	Command or Action	Purpose
Step 5	**mpls bgp forwarding** **Example:** `Router(config-if)# mpls bgp forwarding`	(Optional) Configures BGP to enable MPLS forwarding on connecting interfaces for VRFs that must support MPLS traffic. • This step applies to a CSC network only.
Step 6	**end** **Example:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.

Configuring the MP-BGP Session Between ASBR Peers

BGP propagates reachability information for VPN-IPv4 prefixes among PE routers by means of the BGP multiprotocol extensions (see 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.

Follow the steps in this section to configure the MP-BGP session on the ASBR.

Use all of the steps in the following procedure to configure the MP BGP session on the peer ASBR.

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*} **inter-as-hybrid**
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.

	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 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 192.168.0.1 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 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 **unicast** keyword specifies IPv4 unicast address prefixes.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 192.168.0.1 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 7	**neighbor** {*ip-address* | *peer-group-name*} **inter-as-hybrid** **Example:** `Router(config-router-af)# neighbor 192.168.0.1 inter-as-hybrid`	Configures eBGP peer router (ASBR) as an Inter-AS Option AB peer. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • If any prefixes are imported into Option AB VRFs, then the imported paths are advertised to this peer. • If any prefixes are received from this peer and are imported into Option AB VRFs, then the imported paths are advertised to iBGP peers.

	Command or Action	Purpose
		Note Advertised routes have RTs that are configured on the VRF. Advertised routes do not have their original RTs.
Step 8	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits from address family configuration mode.
Step 9	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Configuring the Routing Policy for VPNs that Need Inter-AS Connections

Use the steps in this section to configure VRFs for the VPNs that need Inter-AS connections between ASBR peers, by configuring the appropriate routing policy and Option AB configuration.

Use all of the steps in the following procedure to configure additional VPNs that need Inter-AS Option AB connectivity on this ASBR and the peer ASBR.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **address-family ipv4**
6. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
7. For Inter-AS Option AB+, go to Step 10; otherwise, go to Step 8.
8. **inter-as-hybrid** [**csc**]
9. **inter-as-hybrid** [**csc**] [**next-hop** *ip-address*]
10. **inter-as-hybrid next-hop global**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition 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 autonomous system number: 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** **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.
Step 6	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf-af)# 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 list of import, export, or both (import and export) route-target extended communities.

	Command or Action	Purpose
Step 7	For Inter-AS Option AB+, go to Step 10; otherwise, go to Step 8.	—
Step 8	**inter-as-hybrid** [**csc**] **Example:** `Router(config-vrf-af)# inter-as-hybrid`	Specifies the VRF as an Option AB VRF, which has the following effects: • Routes imported to this VRF can be advertised to Option AB peers and VPNv4 iBGP peers. • When routes received from Option AB peers and are imported into the VRF, the next hop table ID of the route is set to the table ID of the VRF. • If the **csc** keyword is not used, a per-VRF label is allocated for imported routes. • When routes are received from Option AB peers and are imported next into the VRF, the learned out label can be installed only in forwarding when the **csc** keyword is used. The **csc** keyword implies the following: • A per-prefix label is allocated for imported routes. • For routes received from Option AB peers that are imported into the VRF, the learned out label is installed in forwarding.

Step 9	**inter-as-hybrid** [**csc**] [**next-hop** *ip-address*]
	
	Example:
	
	`Router(config-vrf-af)# inter-as-hybrid next-hop 192.168.1.0`
		(Optional) Specifies the next hop IP address to be set on paths that are imported into the VRF and that are received from an Option AB peer.
		• The next hop context is also set to the VRF, which imports these paths.
		• The **csc** keyword implies the following:
		  • A per-prefix label is allocated for imported routes.
		  • For routes received from Option AB peers that are imported into the VRF, the learned out label is installed in forwarding.

Step 10	**inter-as-hybrid next-hop global** **Example:** `Router(config-vrf-af)# inter-as-hybrid next-hop global`	(For Option AB+) Enables Inter-AS Option AB+. • Specifies that the next-hop address for BGP updates to be set on paths that are imported to the VRF and that are received from an Option AB+ peer are placed in the global routing table. • The address used is the address of the interface that is at the remote end of the external BGP (eBGP) global shared link. The next-hop context is retained as global and not modified to that of the importing VRF.
Step 11	**end** **Example:** `Router(config-vrf-af)# end`	(Optional) Exits to privileged EXEC mode.

Changing an Inter-AS Option A Deployment to an Option AB Deployment

In an Option A deployment, the VRF instances are back-to-back between the ASBR routers and there is direct connectivity between PE routers of different autonomous systems. The PE routers are attached by multiple physical or logical interfaces, each of which is associated with a given VPN (through a VRF instance).

In the Option AB deployment, the different autonomous systems interconnect by using a single MP-BGP session in the global routing table to carry control plane traffic.

Use the following steps to change an MPLS VPN Inter-AS Option A deployment to an Option AB deployment.

1. Configure the MP-BGP session on the ASBR. BGP multiprotocol extensions are used to define support for address families other than IPv4 so 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. See the Configuring the MP-BGP Session Between ASBR Peers, on page 100 for detailed configuration information.
2. Identify the VRFs that need an upgrade from Option A and configure them for Option AB by using the **inter-as-hybrid** command. See the Configuring the Routing Policy for VPNs that Need Inter-AS Connections, on page 102 for detailed configuration information.
3. Use the following steps in this section to remove the configuration for the eBGP (peer ASBR) neighbor.
4. Repeat all the steps in the following procedure to remove the configuration for additional eBGP (peer ASBR) neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4 vrf** *vrf-name*
5. **no neighbor** {*ip-address* | *peer-group-name*}
6. **exit-address-family**
7. **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	**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	**address-family ipv4 vrf** *vrf-name* **Example:** `Router(config-router)# address-family ipv4 vrf vpn4`	Configures each VRF that is identified in the MP-BGP session on the ASBR so 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. • Enters address family configuration mode to specify an address family for a VRF.
Step 5	**no neighbor** {*ip-address* | *peer-group-name*} **Example:** `Router(config-router-af)# no neighbor 192.168.0.1`	Removes the configuration for the exchange of information with the neighboring eBGP (ASBR) 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 6	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits from address family configuration mode.
Step 7	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Configuration Examples for MPLS VPN--Inter-AS Option AB

The following sections describe standard and CSC MPLS VPN configurations between two ASBR peers that use the Inter-AS AB feature:

Examples Inter-AS AB Network Configuration

The following examples show the configuration of an Inter-AS Option AB network that uses nonoverlapping IP addresses:

Example CE1

```
!
ip cef distributed
!
interface lo0
 ip address 192.168.13.13 255.255.255.255
 no shutdown
!
interface et4/0
 ip address 192.168.36.1 255.255.255.0
 no shutdown
!
router ospf 300
 nsf enforce global
 redistribute connected subnets
 auto-cost reference-bandwidth 1000
 passive-interface et4/0
 network 192.168.13.13 0.0.0.0 area 300
!
router bgp 300
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no synchronization
 neighbor 192.168.36.2 remote-as 100
 neighbor 192.168.36.2 advertisement-interval 5
 address-family ipv4 no auto-summary
 redistribute connected
 neighbor 192.168.36.2 activate
```

Example CE2

```
!
ip cef distributed
!
interface lo0
 ip address 192.168.14.14 255.255.255.255
 no shutdown
!
interface et1/6
 ip address 192.168.37.1 255.255.255.0
 no ipv6 address
 no shutdown
!
router ospf 400
 nsf enforce global
 redistribute connected subnets
 auto-cost reference-bandwidth 1000
```

```
 passive-interface et1/6
 network 192.168.14.14 0.0.0.0 area 400
!
router bgp 400
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no synchronization
 neighbor 192.168.0.2 remote-as 100
 neighbor 192.168.0.2 advertisement-interval 5
 address-family ipv4  no auto-summary
 redistribute connected
 neighbor 192.168.0.2 activate
!
```

Example PE1

```
!
ip cef distributed
!
ip vrf vpn1
   rd 100:1
   route-target import 100:1
   route-target import 200:1
   route-target export 100:1
!
ip vrf vpn2
   rd 100:2
   route-target import 100:2
   route-target import 200:2
   route-target export 100:2
!
mpls ldp router-id lo0 force
mpls ldp graceful-restart
mpls ip
mpls ip propagate-ttl
mpls ldp advertise-labels
mpls label protocol ldp
!
interface lo0
 ip address 192.168.17.17 255.255.255.255
 no shutdown
!
interface gi3/1
 ip vrf forwarding vpn1
 ip address 192.168.36.2 255.255.255.0
 no shutdown
!
interface gi3/8
 mpls ip
 mpls label protocol ldp
 ip address 192.168.31.2 255.255.255.0
!
interface gi3/10
 mpls ip
 mpls label protocol ldp
 ip address 192.168.40.1 255.255.255.0
 no shutdown
!
interface gi3/13
 ip vrf forwarding vpn2
 ip address 192.168.0.2 255.0.0.0
 no shutdown
!
router ospf 100
 nsf enforce global
 redistribute connected subnets
 auto-cost reference-bandwidth 1000
 passive-interface gi3/1
 passive-interface gi3/13
```

```
 network 192.168.0.0 0.0.255.255 area 10
 network 192.168.17.17 0.0.0.0 area 100
 network 192.168.0.0 0.0.255.255 area 100
!
router bgp 100
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no bgp default ipv4-unicast
 no synchronization
 neighbor 192.168.19.19 remote-as 100
 neighbor 192.168.19.19 update-source Loopback0
 address-family ipv4 vrf vpn1
no auto-summary
 redistribute connected
 neighbor 192.168.36.1 remote-as 300
 neighbor 192.168.36.1 activate
 neighbor 192.168.36.1 advertisement-interval 5
 address-family ipv4 vrf vpn2  no auto-summary
 redistribute connected
 neighbor 192.168.37.1 remote-as 400
 neighbor 192.168.37.1 activate
 neighbor 192.168.37.1 advertisement-interval 5
 address-family vpnv4
 bgp scan-time import 5
 neighbor 192.168.19.19 activate
 neighbor 192.168.19.19 send-community extended
!
```

Example Route Reflector 1

```
!
ip cef distributed
mpls ldp router-id lo0 force
mpls ldp graceful-restart
mpls ip
mpls ip propagate-ttl
mpls ldp advertise-labels
mpls ip
mpls label protocol ldp
!
interface lo0
 ip address 192.168.19.19 255.255.255.255
 no shutdown
!
interface gi3/3
 mpls ip
 mpls label protocol ldp
 ip address 192.168.40.2 255.255.255.0
 no shutdown
!
router ospf 100
 nsf enforce global
 redistribute connected subnets
 auto-cost reference-bandwidth 1000
 network 192.168.19.19 0.0.0.0 area 100
 network 192.168.0.0 0.0.255.255 area 100 !
router bgp 100
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.11.11 remote-as 100
 neighbor 192.168.11.11 update-source Loopback0
 neighbor 192.168.17.17 remote-as 100
 neighbor 192.168.17.17 update-source Loopback0
 neighbor 192.168.11.11 route-reflector-client
 address-family ipv4
 no neighbor 192.168.17.17 activate
 neighbor 192.168.11.11 route-reflector-client
 address-family vpnv4
```

```
 bgp scan-time import 5
 neighbor 192.168.11.11 activate
 neighbor 192.168.11.11 send-community extended
 neighbor 192.168.17.17 activate
 neighbor 192.168.17.17 send-community extended
 neighbor 192.168.11.11 route-reflector-client
 neighbor 192.168.17.17 route-reflector-client
!
```

Example ASBR1

```
!
ip cef distributed
!
ip vrf vpn1
   rd 100:1
   route-target import 100:1
   route-target import 200:1
   route-target export 100:1
   inter-as-hybrid next-hop 192.168.32.2
exit
ip vrf vpn2
   rd 100:2
   route-target import 100:2
   route-target import 200:2
   route-target export 100:2
   inter-as-hybrid next-hop 192.168.33.2
exit
mpls ldp router-id lo0 force
mpls ldp graceful-restart
mpls ip
mpls ip propagate-ttl
mpls ldp advertise-labels
 mpls ip
 mpls label protocol ldp
interface lo0
 ip address 192.168.11.11 255.255.255.255
 no ipv6 address
ip route-cache distributed
ip route-cache cef distributed
no shutdown
interface gi3/8
mpls ip
 mpls label protocol ldp
 ip address 192.168.13.1 255.255.255.0
 no ipv6 address
ip route-cache distributed
ip route-cache cef distributed
no shutdown
interface gi3/10
   ip vrf forwarding vpn1
 ip address 192.168.32.1 255.255.255.0
 no ipv6 address
ip route-cache distributed
ip route-cache cef distributed
no shutdown
interface gi3/11
   ip vrf forwarding vpn2
 ip address 192.168.33.1 255.255.255.0
 no ipv6 address
ip route-cache distributed
ip route-cache cef distributed
no shutdown
interface gi3/46
 ip address 192.168.34.1 255.255.255.0
 no ipv6 address
ip route-cache distributed
ip route-cache cef distributed
no shutdown
router ospf 100
```

```
nsf enforce global
 redistribute connected subnets
auto-cost reference-bandwidth 1000
passive-interface gi3/10
passive-interface gi3/11
passive-interface gi3/46
network 192.168.0.0 0.0.255.255 area 100
network 192.168.11.11 0.0.0.0 area 100

router bgp 100
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no synchronization
 no bgp default route-target filter
 bgp router-id 192.168.11.11
 neighbor 192.168.34.2 remote-as 200
 neighbor 192.168.34.2 advertisement-interval 5
 neighbor 192.168.19.19 remote-as 100
 neighbor 192.168.19.19 update-source Loopback0
 address-family ipv4
  no auto-summary
 address-family ipv4 vrf vpn1
  no auto-summary
 address-family ipv4 vrf vpn2
  no auto-summary
 address-family vpnv4
  bgp scan-time import 5
  neighbor 192.168.34.2 activate
  neighbor 192.168.34.2 send-community both
  neighbor 192.168.34.2 inter-as-hybrid
  neighbor 192.168.19.19 activate
  neighbor 192.168.19.19 send-community extended !
ip route vrf vpn1 192.168.12.12 255.255.255.255 gi3/10 192.168.32.2
ip route vrf vpn2 192.168.12.12 255.255.255.255 gi3/11 192.168.33.2
!
```

Example ASBR 3

```
!
ip cef distributed
!
ip vrf vpn1
   rd 200:1
   route-target import 100:1
   route-target import 200:1
   route-target export 200:1
   inter-as-hybrid next-hop 192.168.32.1
!
ip vrf vpn2
   rd 200:2
   route-target import 100:2
   route-target import 200:2
   route-target export 200:2
   inter-as-hybrid next-hop 192.168.33.1
!
mpls ldp router-id lo0 force
mpls ldp graceful-restart
mpls ip
mpls ip propagate-ttl
mpls ldp advertise-labels
mpls label protocol ldp
!
interface lo0
 ip address 192.168.12.12 255.255.255.255
 no shutdown
!
interface po2/1/0
 mpls ip
 mpls label protocol ldp
```

```
 ip address 192.168.35.1 255.255.255.0
 crc 16
 clock source internal
 no shutdown
!
interface gi3/10
 ip vrf forwarding vpn1
 ip address 192.168.32.2 255.255.255.0
 no shutdown
!
interface gi3/11
 ip vrf forwarding vpn2
 ip address 192.168.33.2 255.255.255.0
 no shutdown
!
interface gi3/45
 ip address 192.168.34.2 255.255.255.0
 no shutdown
!
router ospf 200
nsf enforce global
redistribute connected subnets
auto-cost reference-bandwidth 1000
passive-interface gi3/10
passive-interface gi3/11
passive-interface gi3/45
network 192.168.0.0 0.0.255.255 area 200 network 192.168.12.12 0.0.0.0 area 200

router bgp 200
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no synchronization
 no bgp default route-target filter
 bgp router-id 192.168.12.12
 neighbor 192.168.34.1 remote-as 100
 neighbor 192.168.34.1 advertisement-interval 5
 neighbor 192.168.20.20 remote-as 200
 neighbor 192.168.20.20 update-source Loopback0
 address-family ipv4
 no auto-summary
 address-family ipv4 vrf vpn1
  no auto-summary
 address-family ipv4 vrf vpn2
  no auto-summary
 address-family vpnv4
  bgp scan-time import 5
  neighbor 192.168.34.1 activate
  neighbor 192.168.34.1 send-community both
  neighbor 192.168.34.1 inter-as-hybrid
  neighbor 192.168.20.20 activate
  neighbor 192.168.20.20 send-community extended !
ip route vrf vpn1 192.168.11.11 255.255.255.255 gi3/10 192.168.32.1
ip route vrf vpn2 192.168.11.11 255.255.255.255 gi3/11 192.168.33.1
!
```

Example PE2

```
!
ip cef distributed
!
ip vrf vpn1
   rd 200:1
   route-target import 100:1
   route-target import 200:1
   route-target export 200:1
!
ip vrf vpn2
   rd 200:2
   route-target import 100:2
```

```
   route-target import 200:2
   route-target export 200:2
!
mpls ldp router-id lo0 force
mpls ldp graceful-restart
mpls ip
mpls ip propagate-ttl
mpls ldp advertise-labels
mpls label protocol ldp
!
interface lo0
 ip address 192.168.18.18 255.255.255.255
 no shutdown
!
interface po1/0/0
 mpls ip
 mpls label protocol ldp
 ip address 192.168.35.2 255.255.255.0
 crc 16
 clock source internal
 no shutdown
!
interface gi3/2
 ip vrf forwarding vpn1
 ip address 192.168.38.2 255.255.255.0
 no shutdown
!
interface gi3/8
 mpls ip
 mpls label protocol ldp
 ip address 192.168.4.1 255.255.255.0
 no shutdown
!
interface gi3/10
 ip vrf forwarding vpn2
 ip address 192.168.39.2 255.255.255.0
 no shutdown
!
router ospf 200
 nsf enforce global
 redistribute connected subnets
 auto-cost reference-bandwidth 1000
 passive-interface gi3/10
 passive-interface gi3/2
 network 192.168.0.0 0.0.255.255 area 200
 network 192.168.18.18 0.0.0.0 area 200
 network 192.168.0.0 0.0.255.255 area 200 !
 router bgp 200
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no bgp default ipv4-unicast
 no synchronization
 neighbor 192.168.20.20 remote-as 200
 neighbor 192.168.20.20 update-source Loopback0
 address-family ipv4 vrf vpn1
    no auto-summary
    redistribute connected
    neighbor 192.168.38.1 remote-as 500
    neighbor 192.168.38.1 activate
    neighbor 192.168.38.1 advertisement-interval 5
 address-family ipv4 vrf vpn2
    no auto-summary
    redistribute connected
    neighbor 192.168.9.1 remote-as 600
    neighbor 192.168.9.1 activate
    neighbor 192.168.9.1 advertisement-interval 5
 address-family vpnv4
    bgp scan-time import 5
    neighbor 192.168.20.20 activate
    neighbor 192.168.20.20 send-community extended
!
```

Example CE3

```
!
ip cef distributed
!
interface lo0
 ip address 192.168.15.15 255.255.255.255
 no shutdown
!
interface gi0/2
 ip address 192.168.38.1 255.255.255.0
 no shutdown
!
router ospf 500
 nsf enforce global
 redistribute connected subnets
 auto-cost reference-bandwidth 1000
 passive-interface gi0/2
 network 192.168.15.15 0.0.0.0 area 500
!
router bgp 500
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no synchronization
 neighbor 192.168.38.2 remote-as 200
 neighbor 192.168.38.2 advertisement-interval 5
 address-family ipv4
 no auto-summary
 redistribute connected
 neighbor 192.168.38.2 activate
!
```

Example CE4

```
!
ip cef distributed
!
interface lo0
 ip address 192.168.16.16 255.255.255.255
 no shutdown
!
interface et6/2
 ip address 192.168.9.1 255.255.255.0
 no shutdown
!
router ospf 600
 nsf enforce global
 redistribute connected subnets
 auto-cost reference-bandwidth 1000
 passive-interface et6/2
 network 192.168.16.16 0.0.0.0 area 600
!
router bgp 600
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 no synchronization
 neighbor 192.168.39.2 remote-as 200
 neighbor 192.168.39.2 advertisement-interval 5
 address-family ipv4  no auto-summary
 redistribute connected
 neighbor 192.168.39.2 activate
!
```

Examples Inter-AS AB CSC Configuration

The following examples show the configuration of an Inter-AS Option AB network with CSC:

Example CE1

```
!
ip cef distributed
!
interface Loopback0
 ip address 192.168.20.20 255.255.255.255
!
interface Ethernet3/3
 ip address 192.168.41.2 255.255.255.0
!
!
router bgp 500
 bgp router-id 192.168.20.20
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.4.1 remote-as 300
 !
 address-family ipv4
  redistribute connected
  neighbor 192.168.4.1 activate
  neighbor 192.168.4.1 advertisement-interval 5
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example CE2

```
!
ip cef distributed
!
interface Loopback0
 ip address 192.168.21.21 255.255.255.255
!
interface Ethernet0/0/7
 ip address 192.168.42.2 255.255.255.0
!
router bgp 600
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart  neighbor 192.168.42.1 remote-as 400
 !
 address-family ipv4
  redistribute connected
  neighbor 192.168.42.1 activate
  neighbor 192.168.42.1 advertisement-interval 5
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example CE3

```
!
ip cef distributed
!
interface Loopback0
 ip address 192.168.22.22 255.255.255.255
!
interface Ethernet6/2
 ip address 192.168.43.2 255.255.255.0
!
router bgp 500
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart neighbor 192.168.43.1 remote-as 300
 !
 address-family ipv4
  redistribute connected
  neighbor 192.168.43.1 activate
  neighbor 192.168.43.1 advertisement-interval 5
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example CE4

```
!
ip cef distributed
!
interface Loopback0
 ip address 192.168.23.23 255.255.255.255
!
!
interface Ethernet0/0/7
 ip address 192.168.44.2 255.255.255.0
!
router bgp 600
 bgp router-id 192.168.23.23
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
neighbor 192.168.44.1 remote-as 400
 !
 address-family ipv4
  redistribute connected
  neighbor 192.168.44.1 activate
  neighbor 192.168.44.1 advertisement-interval 5
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example PE1

```
!
ip cef distributed
!
ip vrf vpn3
 rd 300:3
 route-target export 300:3
 route-target import 300:3
```

```
!
mpls ldp graceful-restart
!
mpls label protocol ldp
!
mpls ip
!
interface Loopback0
 ip address 192.168.192.10 255.255.255.255
!
interface Ethernet3/1
 ip vrf forwarding vpn3
 ip address 192.168.4.1 255.255.255.0
!
interface Ethernet5/3
 ip address 192.168.3.1 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
!
router ospf 300
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 network 192.168.192.10 0.0.0.0 area 300
 network 192.168.0.0 0.0.255.255 area 300
!
router bgp 300
 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 192.168.19.19 remote-as 300
 neighbor 192.168.19.19 update-source Loopback0
 !
 address-family vpnv4
  neighbor 192.168.19.19 activate
  neighbor 192.168.19.19 send-community extended
  bgp scan-time import 5
 exit-address-family
 !
 address-family ipv4 vrf vpn3
  redistribute connected
  neighbor 192.168.41.2 remote-as 500
  neighbor 192.168.41.2 activate
  neighbor 192.168.41.2 as-override
  neighbor 192.168.41.2 advertisement-interval 5
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example CSC-CE1

```
!
ip cef distributed
!
mpls ldp graceful-restart
mpls label protocol ldp
!
mpls ip
!
interface Loopback0
 ip address 192.168.11.11 255.255.255.255
!
!
interface Ethernet3/4
 ip address 192.168.30.2 255.255.255.0
 mpls label protocol ldp
```

```
 mpls ip
!
router ospf 300
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 redistribute bgp 300 metric 3 subnets
 passive-interface FastEthernet1/0
 network 192.168.11.11 0.0.0.0 area 300
 network 192.168.0.0 0.0.255.255 area 300
 distance ospf intra-area 19 inter-area 19
!
router bgp 300
 bgp router-id 192.168.11.11
 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 192.168.13.1 remote-as 100
 !
 address-family ipv4
  redistribute ospf 300 metric 4 match internal external 1 external 2
  neighbor 192.168.13.1 activate
  neighbor 192.168.13.1 send-label
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example CSC-PE1

```
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
 route-target import 100:5
 route-target import 200:1
!
ip vrf vpn2
 rd 100:2
 route-target export 100:2
 route-target import 100:2
 route-target import 100:6
 route-target import 200:2
!
mpls ldp graceful-restart
mpls label protocol ldp
!
mpls ip
!
interface Loopback0
 ip address 192.168.12.12 255.255.255.255
!
!
interface FastEthernet4/0/0
 ip address 192.168.34.1 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
interface FastEthernet4/0/1
 ip vrf forwarding vpn1
 ip address 192.168.13.1 255.255.255.0
 mpls bgp forwarding
!
!
interface FastEthernet4/1/0
 ip vrf forwarding vpn2
 ip address 192.168.33.1 255.255.255.0
```

```
 mpls bgp forwarding
!
router ospf 100
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 network 192.168.12.12 0.0.0.0 area 100
 network 192.168.0.0 0.0.255.255 area 100
!
router bgp 100
 bgp router-id 192.168.12.12
 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 192.168.15.15 remote-as 100
 neighbor 192.168.15.15 update-source Loopback0
 !
 address-family vpnv4
  neighbor 192.168.15.15 activate
  neighbor 192.168.15.15 send-community extended
  bgp scan-time import 5
 exit-address-family
 !
 address-family ipv4 vrf vpn2
  neighbor 192.168.33.2 remote-as 400
  neighbor 192.168.33.2 update-source FastEthernet4/1/0
  neighbor 192.168.33.2 activate
  neighbor 192.168.33.2 as-override
  neighbor 192.168.33.2 advertisement-interval 5
  neighbor 192.168.33.2 send-label
  no auto-summary
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf vpn1
  neighbor 192.168.31.2 remote-as 300
  neighbor 192.168.31.2 update-source FastEthernet4/0/1
  neighbor 192.168.31.2 activate
  neighbor 192.168.31.2 as-override
  neighbor 192.168.31.2 advertisement-interval 5
  neighbor 192.168.31.2 send-label
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example PE 2

```
ip cef distributed
!
ip vrf vpn4
 rd 400:4
 route-target export 400:4
 route-target import 400:4
!
!
mpls ldp graceful-restart
mpls label protocol ldp
!
mpls ip
!
interface Loopback0
 ip address 192.168.13.13 255.255.255.255
!
!
interface Ethernet4/1/2
 ip vrf forwarding vpn4
```

```
 ip address 192.168.42.1 255.255.255.0
!
!
interface Ethernet4/1/6
 ip address 192.168.32.1 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
!
router ospf 400
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 network 192.168.13.13 0.0.0.0 area 400
 network 192.168.0.0 0.0.255.255 area 400
!
router bgp 400
 bgp router-id 192.168.13.13
 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 192.168.25.25 remote-as 400
 neighbor 192.168.25.25 update-source Loopback0
 !
 address-family vpnv4
  neighbor 192.168.25.25 activate
  neighbor 192.168.25.25 send-community extended
  bgp scan-time import 5
 exit-address-family
 !
 address-family ipv4 vrf vpn4
  redistribute connected
  neighbor 192.168.42.2 remote-as 600
  neighbor 192.168.42.2 activate
  neighbor 192.168.42.2 as-override
  neighbor 192.168.42.2 advertisement-interval 5
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example CSC-CE2

```
!
ip cef distributed
!
mpls ldp graceful-restart
mpls label protocol ldp
!
mpls ip
interface Loopback0
 ip address 192.168.14.14 255.255.255.255
!
!
interface GigabitEthernet8/16
 ip address 192.168.33.2 255.255.255.0
 mpls bgp forwarding
!
!
interface GigabitEthernet8/24
 ip address 192.168.32.2 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
!
router ospf 400
 log-adjacency-changes
```

```
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 redistribute bgp 400 metric 3 subnets
 passive-interface GigabitEthernet8/16
 network 192.168.14.14 0.0.0.0 area 400
 network 192.168.0.0 0.0.255.255 area 400
 distance ospf intra-area 19 inter-area 19
!
router bgp 400
 bgp router-id 192.168.14.14
 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 192.168.33.1 remote-as 100
 !
 address-family ipv4
  no synchronization
  redistribute connected
  redistribute ospf 400 metric 4 match internal external 1 external 2
  neighbor 192.168.33.1 activate
  neighbor 192.168.33.1 advertisement-interval 5
  neighbor 192.168.33.1 send-label
  no auto-summary
 exit-address-family
!
```

Example ASBR1

```
!
ip vrf vpn5
 rd 100:5
 route-target export 100:5
 route-target import 100:5
 route-target import 100:1
 route-target import 200:5
 inter-as-hybrid csc next-hop 192.168.35.2
!
ip vrf vpn6
 rd 100:6
 route-target export 100:6
 route-target import 100:6
 route-target import 100:2
 route-target import 200:6
 inter-as-hybrid csc next-hop 192.168.36.2
!
mpls ldp graceful-restart
mpls label protocol ldp
!
!
interface Loopback0
 ip address 192.168.15.15 255.255.255.255
!
interface GigabitEthernet2/3
 ip vrf forwarding vpn5
 ip address 192.168.35.1 255.255.255.0
 mpls bgp forwarding
!
interface GigabitEthernet2/4
 ip vrf forwarding vpn6
 ip address 192.168.36.1 255.255.255.0
 mpls bgp forwarding
!
!
interface GigabitEthernet2/5
 ip address 192.168.34.2 255.255.255.0
 mpls label protocol ldp
 mpls ip
```

```
!
!
interface GigabitEthernet2/16
 ip address 192.168.37.1 255.255.255.0
 mpls bgp forwarding
!
!
router ospf 100
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 network 192.168.15.15 0.0.0.0 area 100
 network 192.168.0.0 0.0.255.255 area 100
!
router bgp 100
 bgp router-id 192.168.15.15
 no bgp default ipv4-unicast
 no bgp default route-target filter
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.12.12 remote-as 100
 neighbor 192.168.12.12 update-source Loopback0
 neighbor 192.168.0.2 remote-as 200
 neighbor 192.168.0.2 disable-connected-check
 !
 address-family ipv4
  no synchronization
  no auto-summary
 exit-address-family
 !
 address-family vpnv4
  neighbor 192.168.12.12 activate
  neighbor 192.168.12.12 send-community extended
  neighbor 192.168.0.2 activate
  neighbor 192.168.0.2 send-community extended
  neighbor 192.168.0.2 inter-as-hybrid
 exit-address-family
 !
 address-family ipv4 vrf vpn5
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf vpn6
  no synchronization
 exit-address-family
!
ip route 192.168.16.16 255.255.255.255 GigabitEthernet2/16 192.168.0.2
ip route vrf vpn5 192.168.16.16 255.255.255.255 GigabitEthernet2/3 192.168.35.2
ip route vrf vpn6 192.168.16.16 255.255.255.255 GigabitEthernet2/4 192.168.36.2
!
ip vrf vpn5
 rd 200:5
 route-target export 200:5
 route-target import 200:5
 route-target import 200:1
 route-target import 100:1
 route-target import 100:5
 inter-as-hybrid csc next-hop 192.168.35.1
!
ip vrf vpn6
 rd 200:6
 route-target export 200:6
 route-target import 200:6
 route-target import 200:2
 route-target import 100:2
 route-target import 100:6
 inter-as-hybrid csc next-hop 192.168.36.1
!
mpls ldp graceful-restart
mpls label protocol ldp
```

```
!
!
interface Loopback0
 ip address 192.168.16.16 255.255.255.255
!
!
interface GigabitEthernet3/1
 ip vrf forwarding vpn5
 ip address 192.168.35.2 255.255.255.0
 mpls bgp forwarding
!
interface GigabitEthernet3/2
 ip vrf forwarding vpn6
 ip address 192.168.36.2 255.255.255.0
 mpls bgp forwarding
!       !
interface GigabitEthernet3/14
 ip address 192.168.0.2 255.0.0.0
 mpls bgp forwarding
!
interface GigabitEthernet3/15
 ip address 192.168.38.2 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
router ospf 200
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 network 192.168.16.16 0.0.0.0 area 200
 network 192.168.0.0 0.0.255.255 area 200
!
router bgp 200
 bgp router-id 192.168.16.16
 no bgp default ipv4-unicast
 no bgp default route-target filter
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.17.17 remote-as 200
 neighbor 192.168.17.17 update-source Loopback0
 neighbor 192.168.37.1 remote-as 100
 neighbor 192.168.37.1 disable-connected-check
 !
 address-family ipv4
  no synchronization
  no auto-summary
 exit-address-family
 !
 address-family vpnv4
  neighbor 192.168.17.17 activate
  neighbor 192.168.17.17 send-community extended
  neighbor 192.168.37.1 activate
  neighbor 192.168.37.1 send-community extended
  neighbor 192.168.37.1 inter-as-hybrid
 exit-address-family
 !
 address-family ipv4 vrf vpn5
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf vpn6
  no synchronization
 exit-address-family
!
ip route 192.168.15.15 255.255.255.255 GigabitEthernet3/14 192.168.37.1
ip route vrf vpn5 192.168.15.15 255.255.255.255 GigabitEthernet3/1 192.168.35.1
ip route vrf vpn6 192.168.15.15 255.255.255.255 GigabitEthernet3/2 192.168.36.1
!
```

Example CSC-PE 3

```
ip vrf vpn1
 rd 200:1
 route-target export 200:1
 route-target import 200:1
 route-target import 200:5
 route-target import 100:1
!
ip vrf vpn2
 rd 200:2
 route-target export 200:2
 route-target import 200:2
 route-target import 200:6
 route-target import 100:2
!
mpls ldp graceful-restart
mpls label protocol ldp
!
mpls ip
!
interface Loopback0
 ip address 192.168.17.17 255.255.255.255
!
interface FastEthernet4/0/2
 ip vrf forwarding vpn2
 ip address 192.168.5.1 255.255.255.0
 mpls bgp forwarding
!
!
interface FastEthernet4/0/4
 ip vrf forwarding vpn1
 ip address 192.168.9.1 255.255.255.0
 mpls bgp forwarding
!
!
interface FastEthernet4/0/7
 ip address 192.168.38.1 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
router ospf 200
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 network 192.168.17.17 0.0.0.0 area 200
 network 192.168.0.0 0.0.255.255 area 200
!
router bgp 200
 bgp router-id 192.168.17.17
 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 192.168.16.16 remote-as 200
 neighbor 192.168.16.16 update-source Loopback0
 !
 address-family vpnv4
  neighbor 192.168.16.16 activate
  neighbor 192.168.16.16 send-community extended
  bgp scan-time import 5
 exit-address-family
 !
 address-family ipv4 vrf vpn2
  neighbor 192.168.55.0 remote-as 400
  neighbor 192.168.55.0 update-source FastEthernet4/0/2
  neighbor 192.168.55.0 activate
  neighbor 192.168.55.0 as-override
```

```
  neighbor 192.168.55.0 advertisement-interval 5
  neighbor 192.168.55.0 send-label
  no auto-summary
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf vpn1
  neighbor 192.168.39.2 remote-as 300
  neighbor 192.168.39.2 update-source FastEthernet4/0/4
  neighbor 192.168.39.2 activate
  neighbor 192.168.39.2 as-override
  neighbor 192.168.39.2 advertisement-interval 5
  neighbor 192.168.39.2 send-label
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example CSC-CE3

```
!
interface Loopback0
 ip address 192.168.18.18 255.255.255.255
!
!
interface Ethernet3/3
 ip address 192.168.40.2 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
!
interface FastEthernet5/0
 ip address 192.168.39.2 255.255.255.0
 mpls bgp forwarding
!
!
router ospf 300
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 redistribute bgp 300 metric 3 subnets
 network 192.168.18.18 0.0.0.0 area 300
 network 192.168.0.0 0.0.255.255 area 300
 distance ospf intra-area 19 inter-area 19
!
router bgp 300
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.9.1 remote-as 200
 !
 address-family ipv4
  redistribute connected
  redistribute ospf 300 metric 4 match internal external 1 external 2
  neighbor 192.168.9.1 activate
  neighbor 192.168.9.1 advertisement-interval 5
  neighbor 192.168.9.1 send-label
  no auto-summary
  no synchronization
 exit-address-family
!
```

Example CSC-CE 4

```
!
ip cef distributed
```

```
!
mpls ldp graceful-restart
mpls label protocol ldp
!
mpls ip
!
interface Loopback0
 ip address 192.168.24.24 255.255.255.255
!
!
interface FastEthernet1/1
 ip address 192.168.55.0 255.255.255.0
 mpls bgp forwarding
!
!
interface Ethernet3/5
 ip address 192.168.56.2 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
!
router ospf 400
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 redistribute bgp 400 metric 3 subnets
 network 192.168.24.24 0.0.0.0 area 400
 network 192.168.0.0 0.0.255.255 area 400
!
router bgp 400
 bgp log-neighbor-changes
 neighbor 192.168.5.1 remote-as 200
 !
 address-family ipv4
  redistribute connected
  redistribute ospf 400 metric 4 match internal external 1 external 2
  neighbor 192.168.5.1 activate
  neighbor 192.168.5.1 advertisement-interval 5
  neighbor 192.168.5.1 send-label
  no auto-summary
  no synchronization
 exit-address-family
```

Example PE 3

```
!
ip cef distributed
!
ip vrf vpn3
 rd 300:3
 route-target export 300:3
 route-target import 300:3
 mpls ip
!
!
mpls ldp graceful-restart
mpls label protocol ldp
!
!
interface Loopback0
 ip address 192.168.19.19 255.255.255.255
!
!
interface Ethernet5/1/1
 ip vrf forwarding vpn3
 ip address 192.168.43.1 255.255.255.0
!
!
interface Ethernet5/1/4
 ip address 192.168.40.1 255.255.255.0
```

```
 mpls label protocol ldp
 mpls ip
!
!
router ospf 300
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 network 192.168.19.19 0.0.0.0 area 300
 network 192.168.0.0 0.0.255.255 area 300
 network 192.168.0.0 0.0.255.255 area 300
!
router bgp 300
 bgp router-id 192.168.19.19
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.192.10 remote-as 300
 neighbor 192.168.192.10 update-source Loopback0
 !
 address-family ipv4
  no neighbor 192.168.192.10 activate
  no auto-summary
  no synchronization
 exit-address-family
 !
 address-family vpnv4
  neighbor 192.168.192.10 activate
  neighbor 192.168.192.10 send-community extended
  bgp scan-time import 5
 exit-address-family
 !
 address-family ipv4 vrf vpn3
  neighbor 192.168.43.2 remote-as 500
  neighbor 192.168.43.2 activate
  neighbor 192.168.43.2 as-override
  neighbor 192.168.43.2 advertisement-interval 5
  no auto-summary
  no synchronization
 exit-address-family
```

Example PE 4

```
!
ip cef distributed
!
ip vrf vpn4
 rd 400:4
 route-target export 400:4
 route-target import 400:4
!
mpls ldp graceful-restart
mpls ldp protocol ldp
!
mpls ip
!
interface Loopback0
 ip address 192.168.25.25 255.255.255.255
!
!
interface Ethernet5/0/4
 ip address 192.168.56.1 255.255.255.0
 mpls label protocol ldp
 mpls ip
!
!
interface Ethernet5/0/7
 ip vrf forwarding vpn4
```

```
 ip address 192.168.44.1 255.255.255.0
!
!
router ospf 400
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 network 192.168.25.25 0.0.0.0 area 400
 network 192.168.0.0 0.0.255.255 area 400
!
router bgp 400
 bgp router-id 192.168.25.25
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 192.168.13.13 remote-as 400
 neighbor 192.168.13.13 ebgp-multihop 7
 neighbor 192.168.13.13 update-source Loopback0
 !
 address-family ipv4
  no neighbor 192.168.13.13 activate
  no auto-summary
  no synchronization
 exit-address-family
 !
 address-family vpnv4
  neighbor 192.168.13.13 activate
  neighbor 192.168.13.13 send-community extended
  bgp scan-time import 5
 exit-address-family
 !
 address-family ipv4 vrf vpn4
  neighbor 192.168.44.2 remote-as 600
  neighbor 192.168.44.2 activate
  neighbor 192.168.44.2 as-override
  neighbor 192.168.44.2 advertisement-interval 5
  no auto-summary
  no synchronization
 exit-address-family
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
MPLS commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
MPLS VPNs	Configuring MPLS Layer 3 VPNs
MPLS VPN interautonomous systems	• MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses • MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

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 2283	*Multiprotocol Extensions for BGP-4*
RFC 4364	*BGP/MPLS IP Virtual Private Networks*

Technical Assistance

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

Feature Information for MPLS VPN--Inter-AS Option AB

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 MPLS VPN--Inter-AS Option AB

Feature Name	Release	Feature Information
MPLS VPN--Inter-AS Option AB	12.2(33)SRC 15.0(1)M 15.0(1)S 15.0(1)SY Cisco IOS XE Release 2.4	This feature combines the best functionality of an Inter-AS Option 10 A and Inter-AS Option 10 B network to allow an MPLS VPN service provider to interconnect different autonomous systems to provide VPN services. In Cisco IOS Release 12.2(33)SRC, this feature was introduced. In Cisco IOS Release 15.0(1)M, this feature was implemented on Cisco 1900, 2900, 3800, and 3900 series routers. In Cisco IOS XE Release 2.4, this feature was implemented on the Cisco ASR 1000 Series Routers. These commands were introduced or modified: **neighbor inter-as-hybrid**, **inter-as-hybrid**.

Feature Name	Release	Feature Information
MPLS VPN--Inter-AS Option AB+	15.0(1)SY	The MPLS VPN—Inter-AS Option AB+ feature addresses the scalability concerns of MPLS VPN—Inter-AS Option A by using a single BGP session to signal VPN prefixes (as described in Inter-AS Option B). In an Inter-AS AB+ deployment, the forwarding connections between the ASBRs are maintained on a per-VRF basis while the control plane information is exchanged by a single Multiprotocol BGP session. In Cisco IOS Release 15.0(1)SY, this feature was introduced. These commands were introduced or modified: **inter-as-hybrid**.

Glossary

ASBR -- Autonomous System Boundary router. A router that connects one autonomous system to another.

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).

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.

CSC --Carrier Supporting Carrier. A hierarchical VPN model that allows small service providers, or customer carriers, to interconnect their IP or MPLS networks over an MPLS backbone. This eliminates the need for customer carriers to build and maintain their own MPLS backbone.

eBGP --external 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.

edge router--A router that is at the edge of the network. It defines the boundary of the MPLS network. It receives and transmits packets. Also referred to as edge label switch router and label edge router.

iBGP --internal 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.

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.

LDP --Label Distribution Protocol. A standard protocol between MPLS-enabled routers to negotiate the labels (addresses) used to forward packets.

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.

MP-BGP --Multiprotocol BGP.

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.

NSF --Nonstop forwarding enables routers to continuously forward IP packets following a Route Processor takeover or switchover to another Route Processor. NSF maintains and updates Layer 3 routing and forwarding information in the backup Route Processor to ensure that IP packets and routing protocol information are forwarded continuously during the switchover and route convergence process.

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. All MPLS VPN processing occurs in the PE router.

QoS --quality of service. Measure of performance for a transmission system that indicates its transmission quality and service availability.

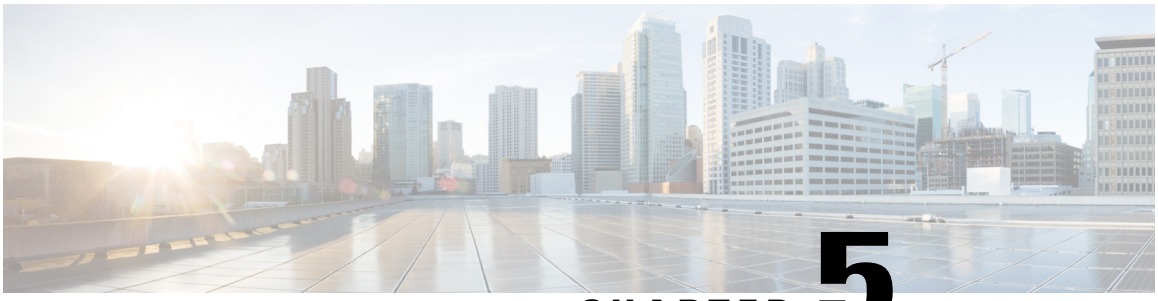
RD --route distinguisher. An 8-byte value that is concatenated with an IPv4 prefix to create a unique VPN-IPv4 prefix.

RT --route target. Extended community attribute used to identify the VRF routing table into which a prefix is imported.

SLA --Service Level Agreement given to VPN subscribers.

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 VPN site that is attached to a 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.



CHAPTER 5

MPLS VPN Carrier Supporting Carrier Using LDP and an IGP

Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) Carrier Supporting Carrier (CSC) enables one MPLS VPN-based service provider to allow other service providers to use a segment of its backbone network. This module explains how to configure the MPLS VPN CSC network using MPLS Label Distribution Protocol (LDP) to distribute MPLS labels and an Interior Gateway Protocol (IGP) to distribute routes.

- Finding Feature Information, page 133
- Prerequisites for MPLS VPN CSC with LDP and IGP, page 134
- Restrictions for MPLS VPN CSC with LDP and IGP, page 134
- Information About MPLS VPN CSC with LDP and IGP, page 135
- How to Configure MPLS VPN CSC with LDP and IGP, page 141
- Configuration Examples for MPLS VPN CSC with LDP and IGP, page 152
- Additional References for MPLS VPN Carrier Supporting Carrier Using LDP and an IGP, page 193
- Feature Information for MPLS VPN CSC with LDP and IGP, page 194
- Glossary, page 194

Finding Feature Information

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

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 CSC with LDP and IGP

- The provider edge (PE) routers of the backbone carrier require 128 MB of memory.
- The backbone carrier must enable the PE router to check that the packets it receives from the customer edge (CE) router contain only the labels that the PE router advertised to the CE router. This prevents data spoofing, which occurs when a packet from an unrecognized IP address is sent to a router.

Restrictions for MPLS VPN CSC with LDP and IGP

The following features are not supported with this feature:

- ATM MPLS
- Carrier supporting carrier traffic engineering
- Carrier supporting carrier quality of service (QoS)
- RSVP aggregation
- VPN Multicast between the customer carrier and the backbone carrier network

The following router platforms are supported on the edge of the MPLS VPN:

- Cisco 7200 series
- Cisco 7500 series
- Cisco 12000 series

See the table below for Cisco 12000 series line card support added for Cisco IOS releases.

Table 6: Cisco12000 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 1-Port OC-12 POS 8-Port OC-3 POS 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 x OC-3 POS ISE 4 Port OC-12 POS ISE 1-Port OC-48 POS ISE	12.0(16)ST 12.0(21)ST 12.0(22)S

Type	Line Cards	Cisco IOS Release Added
Electrical Interface	6- Port DS3 12- Port DS3 6-Port E3	12.0(16)ST 12.0(21)ST
ATM	4-Port OC-3 ATM 1-Port OC12 ATM 4-Port OC-12 ATM	12.0(22)S
Channelized Interface	2-Port CHOC-3 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	12.0(22)S

Information About MPLS VPN CSC with LDP and IGP

MPLS VPN CSC Introduction

Carrier supporting carrier is where one service provider allows another service provider to use a segment of its backbone network. The service provider that provides the segment of the backbone network to the other provider is called the backbone carrier. The service provider that uses the segment of the backbone network is called the customer carrier.

A backbone carrier offers Border Gateway Protocol and Multiprotocol Label Switching (BGP/MPLS) VPN services. The customer carrier can be either:

- An Internet service provider (ISP)
- A BGP/MPLS VPN service provider

Benefits of Implementing MPLS VPN CSC

The MPLS VPN CSC network provides the following benefits to service providers who are backbone carriers and to customer carriers.

Benefits to the Backbone Carrier

- The backbone carrier can accommodate many customer carriers and give them access to its backbone. The backbone carrier does not need to create and maintain separate backbones for its customer carriers. Using one backbone network to support multiple customer carriers simplifies the backbone carrier's

VPN operations. The backbone carrier uses a consistent method for managing and maintaining the backbone network. This is also cheaper and more efficient than maintaining separate backbones.

- The MPLS VPN carrier supporting carrier feature is scalable. Carrier supporting carrier can change the VPN to meet changing bandwidth and connectivity needs. The feature can accommodate unplanned growth and changes. The carrier supporting carrier feature enables tens of thousands of VPNs to be set up over the same network, and it allows a service provider to offer both VPN and Internet services.
- The MPLS VPN carrier supporting carrier feature is a flexible solution. The backbone carrier can accommodate many types of customer carriers. The backbone carrier can accept customer carriers who are ISPs or VPN service providers or both. The backbone carrier can accommodate customer carriers that require security and various bandwidths.

Benefits to the Customer Carriers

- The MPLS VPN carrier supporting carrier feature removes from the customer carrier the burden of configuring, operating, and maintaining its own backbone. The customer carrier uses the backbone network of a backbone carrier, but the backbone carrier is responsible for network maintenance and operation.
- Customer carriers who use the VPN services provided by the backbone carrier receive the same level of security that Frame Relay or ATM-based VPNs provide. Customer carriers can also use IPSec in their VPNs for a higher level of security; it is completely transparent to the backbone carrier.
- Customer carriers can use any link layer technology (SONET, DSL, Frame Relay, and so on) to connect the CE routers to the PE routers and the PE routers to the P routers. The MPLS VPN carrier supporting carrier feature is link layer independent. The CE routers and PE routers use IP to communicate, and the backbone carrier uses MPLS.
- The customer carrier can use any addressing scheme and still be supported by a backbone carrier. The customer address space and routing information are independent of the address space and routing information of other customer carriers or the backbone provider.

Configuration Options for MPLS VPN CSC with LDP and IGP

The backbone carrier offers BGP and MPLS VPN services. The customer carrier can be one of the two types of service providers described in the following sections, which explain how the backbone and customer carriers distribute IPv4 routes and MPLS labels.

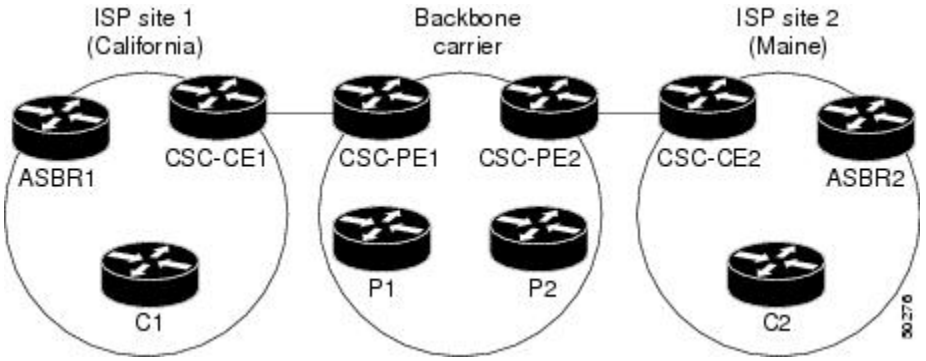
Customer Carrier Is an ISP

This section explains how a BGP/MPLS VPN service provider (backbone carrier) can provide a segment of its backbone network to a customer who is an ISP.

Consider the following example:

An ISP has two sites: one in California, the other in Maine. Each site is a point of presence (POP). The ISP wants to connect these sites using a VPN service provided by a backbone carrier. The figure below illustrates this situation.

Figure 14: Sample BGP/MPLS Backbone Carrier Supporting an ISP



Note The CE routers in the figures are CE routers to the backbone carrier. However, they are PE routers to the customer carrier.

In this example, only the backbone carrier uses MPLS. The customer carrier (ISP) uses only IP. As a result, the backbone carrier must carry all the Internet routes of the customer carrier, which could be as many as 100,000 routes. This poses a scalability problem for the backbone carrier. To solve the scalability problem, the backbone carrier is configured as follows:

- The backbone carrier allows only internal routes of the customer carrier (IGP routes) to be exchanged between the CE routers of the customer carrier and the PE routers of the backbone carrier.
- MPLS is enabled on the interface between the CE router of the customer carrier and the PE router of the backbone carrier.

Internal and external routes are differentiated this way:

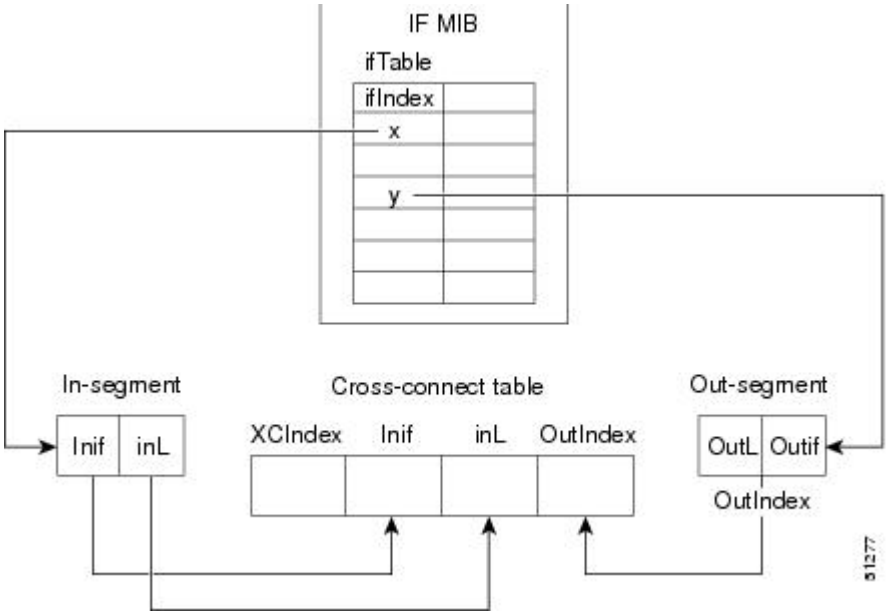
- Internal routes go to any of the routers within the ISP.
- External routes go to the Internet.

The number of internal routes is much lower than the number of external routes. Restricting the routes between the CE routers of the customer carrier and the PE routers of the backbone carrier significantly reduces the number of routes that the PE router needs to maintain.

Because the PE routers do not have to carry external routes in the VRF routing table, they can use the incoming label in the packet to forward the customer carrier Internet traffic. Adding MPLS to the routers provides a consistent method of transporting packets from the customer carrier to the backbone carrier. MPLS allows the exchange of an MPLS label between the PE and the CE routers for every internal customer carrier route. The routers in the customer carrier have all the external routes either through internal Border Gateway Protocol

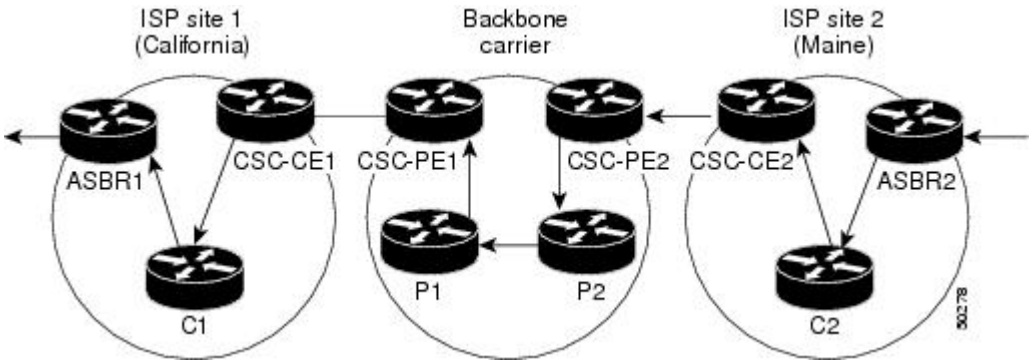
(iBGP) or route redistribution to provide Internet connectivity. The figure below shows how information is exchanged when the network is configured in this manner.

Figure 15: Backbone Carrier Exchanging Routing Information with a Customer Carrier Who Is an ISP



In the figure below, routes are created between the backbone carrier and the customer carrier sites. ASBR2 receives an Internet route that originated outside the network. All routers in the ISP sites have all the external routes through IBGP connections among them.

Figure 16: Establishing a Route Between a Backbone Carrier and a Customer Carrier Who Is an ISP



The table below describes the process of establishing the route, which can be divided into two distinct steps:

- The backbone carrier propagates the IGP information of the customer carrier, which enables the customer carrier routers to reach all the customer carrier routers in the remote sites.
- Once the routers of the customer carriers in different sites are reachable, external routes can be propagated in the customer carrier sites, using IBGP without using the backbone carrier routers.

Table 7: Establishing a Route Between the Backbone Carrier and the Customer Carrier ISP

Step	Description
1	CSC-CE2 sends the internal routes within site 2 to CSC-PE2. The routes include the route to ASBR2.
2	CSC-PE2 sends the routing information for site 2 to CSC-PE1, using MPLS VPN processes. CSC-PE1 gets one label (called L3), which is associated with the route to the VPN-IP address for ASBR2. CSC-PE1 gets another label (called L2), which is associated with the route to CSC-PE2.
3	CSC-PE1 sends the routing information associated with internal routes from site 2 to CSC-CE1. CSC-PE1 also sends the label binding information. As a result, CSC-CE1 gets the route to ASBR2 with CSC-PE1 as the next hop. The label associated with that route is called L1.
4	CSC-CE1 distributes the routing information through site 1. Every router in site 1 gets a route for every internal destination in site 2. Therefore, every router in site 1 can reach routers in site 2 and learn external routes through IBGP.
5	ASBR2 receives an Internet route.
6	The IBGP sessions exchange the external routing information of the ISP, including a route to the Internet. Every router in site 1 knows a route to the Internet, with ASBR2 as the next hop of that route.

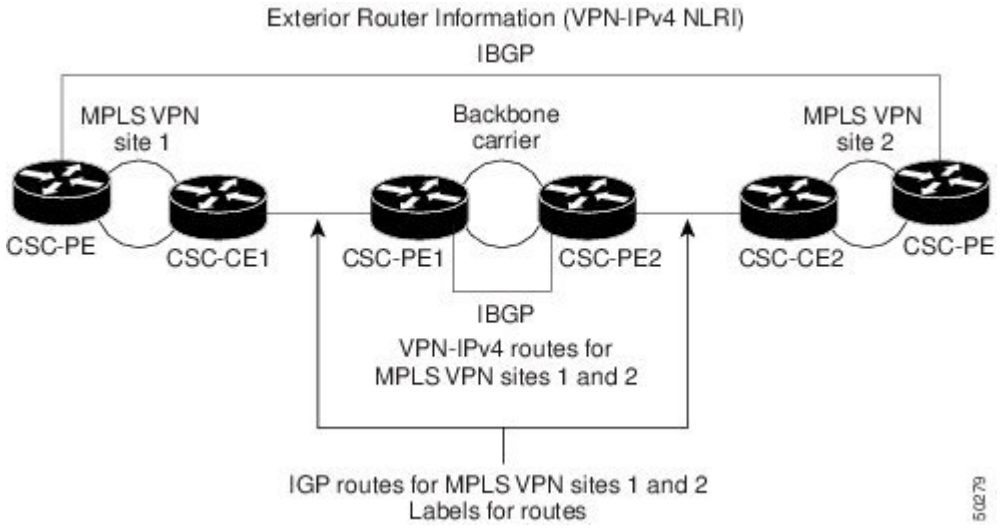
Customer Carrier Is a BGP MPLS VPN Service Provider

When a backbone carrier and the customer carrier both provide BGP/MPLS VPN services, the method of transporting data is different from when a customer carrier provides only ISP services. The following list highlights those differences:

- When a customer carrier provides BGP/MPLS VPN services, its external routes are VPN-IPv4 routes. When a customer carrier is an ISP, its external routes are IP routes.
- When a customer carrier provides BGP/MPLS VPN services, every site within the customer carrier must use MPLS. When a customer carrier is an ISP, the sites do not need to use MPLS.

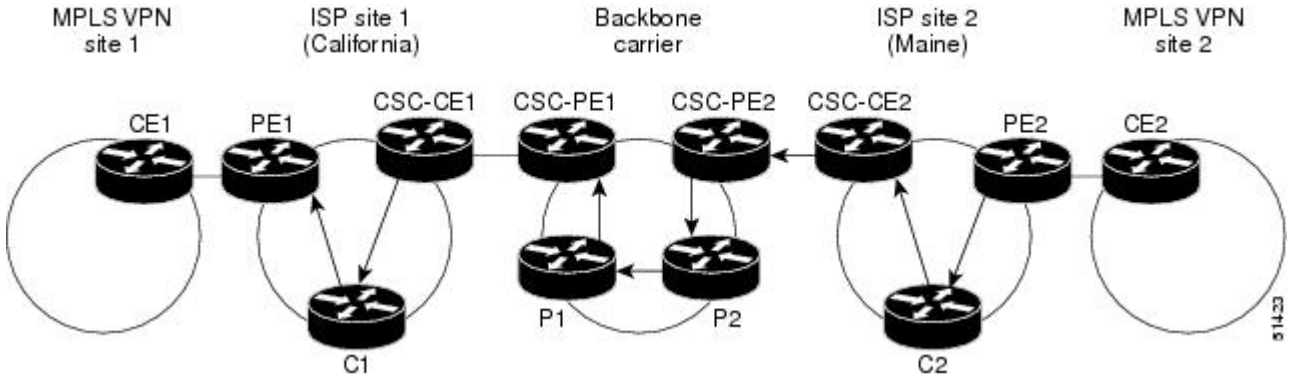
The figure below shows how information is exchanged when MPLS VPN services reside on all customer carrier sites and on the backbone carrier.

Figure 17: Backbone Carrier Exchanging Information with a Customer Carrier Who Is an MPLS VPN Service Provider



In the example shown in the figure below, routes are created between the backbone carrier and the customer carrier sites.

Figure 18: Establishing a Route Between a Backbone Carrier and a Customer Carrier Who Is an MPLS VPN Service Provider



The table below describes the process of establishing the route.

Table 8: Establishing a Route Between the Backbone Carrier and Customer Carrier Site

Step	Description
1	CE2 sends all the internal routes within site 2 to CSC-PE2.

Step	Description
2	CSC-PE2 sends the routing information for site 2 to CSC-PE1, using MPLS VPN processes. CSC-PE1 gets one label (called L3), which is associated with the route to the VPN-IP address for PE2. CSC-PE1 gets another label (called L2), which is associated with the route to CSC-PE2.
3	CSC-PE1 sends the routing information associated with internal routes from site 2 to CSC-CE1. CSC-PE1 also sends the label binding information. As a result, CSC-CE1 gets the route to PE2 with CSC-PE1 as the next hop. The label associated with that route is called L1.
4	CE1 distributes the routing and labeling information through site 1. Every router in site 1 gets a route for every internal destination in site 2. Therefore, PE1 can establish an MP-IBGP session with PE2.
5	CE2 advertises the internal routes of MPLS VPN site 2 to PE2.
6	PE2 allocates labels for all the VPN routes (regular MPLS VPN functionality) and advertises the labels to PE1, using MP-IBGP.
7	PE1 can forward traffic from VPN site 1 that is destined for VPN site 2.

How to Configure MPLS VPN CSC with LDP and IGP

Configuring the Backbone Carrier Core

Configuring the backbone carrier core requires configuring connectivity and routing functions for the CSC core and the CSC-PE routers.

Configuring and verifying the CSC core (backbone carrier) involves the following tasks:

Prerequisites

Before you configure a backbone carrier core, configure the following on the CSC core routers:

- An IGP routing protocol--BGP, OSPF, IS-IS, EIGRP, static, and so on. For information, see Configuring a Basic BGP Network, Configuring OSPF, Configuring a Basic IS-IS Network, and Configuring EIGRP.

- Label Distribution Protocol (LDP). For information, see MPLS Label Distribution Protocol.

Verifying IP Connectivity and LDP Configuration in the CSC Core

Perform this task to verify IP connectivity and LDP configuration in the CSC core. For a configuration example for this task, see the Verifying IP Connectivity and LDP Configuration in the CSC Core, on page 142.

SUMMARY STEPS

1. **enable**
2. **ping** [*protocol*] {*host-name* | *system-address*}
3. **trace** [*protocol*] [*destination*]
4. **show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [*-label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]
5. **show mpls ldp discovery** [**vrf** *vrf-name* | **all**]
6. **show mpls ldp neighbor** [[**vrf** *vrf-name*] [*address* | *interface*] [**detail**] | **all**]
7. **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]
8. **show mpls interfaces** [[**vrf** *vrf-name*] [*interface*] [**detail**] |**all**]
9. **show ip route**
10. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping** [*protocol*] {*host-name*	*system-address*} **Example:** `Router# ping ip 10.0.0.1`	(Optional) Diagnoses basic network connectivity on AppleTalk, Connectionless Network Service (CLNS), IP, Novell, Apollo, VINES, DECnet, or Xerox Network System (XNS) networks. • Use the **ping ip** command to verify the connectivity from one CSC core router to another.
Step 3	**trace** [*protocol*] [*destination*] **Example:** `Router# trace ip 10.0.0.1`	(Optional) Discovers the routes that packets will actually 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 mpls forwarding-table** [*network* {*mask*	*length*}	**labels** *label* [*-label*]	**interface** *interface*	(Optional) Displays the contents of the MPLS label forwarding information base (LFIB).

	Command or Action	Purpose
	| **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**] **Example:** `Router# show mpls forwarding-table`	• Use the **show mpls forwarding-table** command to verify that MPLS packets are being forwarded.
Step 5	**show mpls ldp discovery** [**vrf** *vrf-name* | **all**] **Example:** `Router# show mpls ldp discovery`	(Optional) Displays the status of the LDP discovery process. • Use the **show mpls ldp discovery** command to verify that LDP is operational in the CSC core.
Step 6	**show mpls ldp neighbor** [[**vrf** *vrf-name*] [*address* | *interface*] [**detail**] | **all**] **Example:** `Router# show mpls ldp neighbor`	(Optional) Displays the status of LDP sessions. • Use the**show mpls ldp neighbor** command to verify LDP configuration in the CSC core.
Step 7	**show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**] **Example:** `Router# show ip cef`	(Optional) Displays entries in the forwarding Information Base (FIB). • Use the **show ip cef** command to check the forwarding table (prefixes, next hops, and interfaces).
Step 8	**show mpls interfaces** [[**vrf** *vrf-name*] [*interface*] [**detail**] |**all**] **Example:** `Router# show mpls interfaces`	(Optional) Displays information about one or more or all interfaces that are configured for label switching. • Use the**show mpls interfaces** command to verify that the interfaces are configured to use LDP.
Step 9	**show ip route** **Example:** `Router# show ip route`	(Optional) Displays IP routing table entries. • Use the **show ip route** command to display the entire routing table, including host IP address, next hop, and interface.
Step 10	**disable** **Example:** `Router# disable`	(Optional) Returns to privileged EXEC mode.

Troubleshooting Tips

You can use the **ping** and **trace** commands to verify complete MPLS connectivity in the core. You also get useful troubleshooting information from the additional **show** commands.

Configuring VRFs for CSC-PE Routers

Perform this task to configure VPN routing and forwarding (VRF) instances for the backbone carrier edge (CSC-PE) routers.

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**
8. **interface** *type number*
9. **ip vrf forwarding** *vrf-name*
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	**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*	Creates routing and forwarding tables.

	Command or Action	Purpose
	Example: `Router(config-vrf)# rd 100:1`	• 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, 192.168.122.15:1
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.
Step 8	**interface** *type number* **Example:** `Router(config)# interface Ethernet5/0`	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 9	**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.

	Command or Action	Purpose
Step 10	**end** **Example:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

Enter a **show ip vrf detail** command and make sure the MPLS VPN is up and associated with the right interfaces.

Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier

Perform this task to configure Multiprotocol BGP (MP-BGP) connectivity in the backbone carrier.

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*} **update-source** *interface-type*
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.

	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	**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-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 10.5.5.5 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*} **update-source** *interface-type* **Example:** `Router(config-router)# neighbor 10.2.0.0 update-source loopback0`	Allows BGP sessions to use a specific operational interface for TCP connections. • 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 *interface-type* argument specifies the interface to be used as the source.
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.

	Command or Action	Purpose
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.4.0.0 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

You can enter a **show ip bgp neighbor** command to verify that the neighbors are up and running. If this command generates an error message, enter a **debug ip bgp** *x.x.x.x* **events** command, where *x.x.x.x* is the IP address of the neighbor.

Configuring the CSC-PE and CSC-CE Routers

To enable the CSC-PE and CSC-CE routers to distribute routes and MPLS labels, perform the following tasks:

Prerequisites

Before you configure the CSC-PE and CSC-CE routers, you must configure an IGP on the CSC-PE and CSC-CE routers. A routing protocol is required between the PE and CE routers that connect the backbone carrier to the customer carrier. The routing protocol enables the customer carrier to exchange IGP routing information with the backbone carrier. Use the same routing protocol that the customer carrier uses. You can choose RIP, OSPF, or static routing as the routing protocol. BGP is not supported. For the configuration steps, see Configuring MPLS Layer 3 VPNs .

Configuring LDP on the CSC-PE and CSC-CE Routers

MPLS LDP is required between the PE and CE routers that connect the backbone carrier to the customer carrier. You can configure LDP as the default label distribution protocol for the entire router or just for the PE-to-CE interface for VRF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **interface** *type number*
5. **mpls label protocol ldp**
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	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Specifies MPLS LDP as the default label distribution protocol for the router.
Step 4	**interface** *type number* **Example:** `Router(config)# interface Ethernet5/0`	(Optional) 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 5	**mpls label protocol ldp** **Example:** `Router(config-if)# mpls label protocol ldp`	(Optional) Specifies MPLS LDP as the default label distribution protocol for the interface.
Step 6	**exit** **Example:** `Router(config-if)# exit`	(Optional) Exits to privileged EXEC mode.

Enabling MPLS Encapsulation on the CSC-PE and CSC-CE Routers

Every packet that crosses the backbone carrier must be encapsulated, so that the packet includes MPLS labels. You can enable MPLS encapsulation for the entire router or just on the interface of the PE or CE router. To enable the encapsulation of packets, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ip**
4. **interface** *type number*
5. **mpls ip**
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	**mpls ip** **Example:** `Router(config)# mpls ip`	Enables MPLS encapsulation for the router.
Step 4	**interface** *type number* **Example:** `Router(config)# interface Ethernet5/0`	(Optional) 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.

	Command or Action	Purpose
Step 5	**mpls ip** **Example:** `Router(config-if)# mpls ip`	(Optional) Enables MPLS encapsulation for the specified interface.
Step 6	**exit** **Example:** `Router(config-if)# exit`	(Optional) Exits to privileged EXEC mode.

Verifying the Carrier Supporting Carrier Configuration

The following commands verify the status of LDP sessions that were configured between the backbone carrier and customer carrier. Now the customer carrier ISP sites appear as a VPN customer to the backbone carrier.

SUMMARY STEPS

1. **show mpls ldp discovery vrf** *vrf-name*
2. **show mpls ldp discovery all**

DETAILED STEPS

Step 1 **show mpls ldp discovery vrf** *vrf-name*

Use this command to show that the LDP sessions are in VRF VPN1 of the PE router of the backbone carrier, for example:

Example:

```
Router# show mpls ldp discovery vrf vpn1
Local LDP Identifier:
    10.0.0.0:0
Discovery Sources:
    Interfaces:
        Ethernet1/0 (ldp): xmit/recv
            LDP Id: 10.0.0.1:0
POS6/0 (ldp): xmit
```

Step 2 **show mpls ldp discovery all**

Use this command to list all LDP sessions in a router, for example:

Example:

```
Router# show mpls ldp discovery all
Local LDP Identifier:
    10.10.10.10:0
```

```
Discovery Sources:
    Interfaces:
        Ethernet1/5 (ldp): xmit/recv
            LDP Id: 10.5.5.5:0
VRF vpn1: Local LDP Identifier:
    10.0.0.1:0
Discovery Sources:
    Interfaces:
        Ethernet1/0 (ldp): xmit/recv
            LDP Id: 10.0.0.1:0
POS6/0 (ldp): xmit
```

The Local LDP Identifier field shows the LDP identifier for the local label switching router for this session. The Interfaces field displays the interfaces engaging in LDP discovery activity:

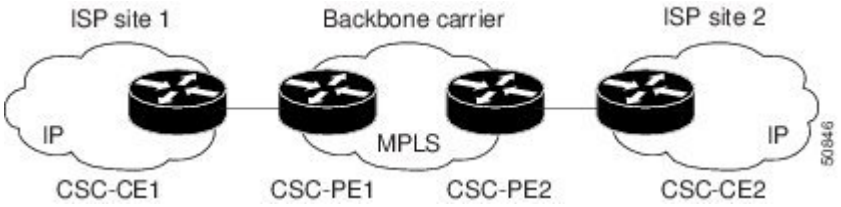
- xmit indicates that the interface is transmitting LDP discovery hello packets.
- recv indicates that the interface is receiving LDP discovery hello packets.

Configuration Examples for MPLS VPN CSC with LDP and IGP

MPLS VPN CSC Network with a Customer Who Is an ISP Example

The figure below shows a carrier supporting carrier network configuration where the customer carrier is an ISP. The customer carrier has two sites, each of which is a POP. The customer carrier connects these sites using a VPN service provided by the backbone carrier. The backbone carrier uses MPLS. The ISP sites use IP. To enable packet transfer between the ISP sites and the backbone carrier, the CE routers that connect the ISPs to the backbone carrier run MPLS.

Figure 19: Carrier Supporting Carrier Network with a Customer Carrier Who Is an ISP



The following examples show the configuration of each router in the carrier supporting carrier network. OSPF is used to connect the customer carrier to the backbone carrier.

CSC-CE1 Configuration

```
mpls label protocol ldp
!
interface Loopback0
 ip address 10.14.14.14 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
```

```
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 101 0 51 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM2/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM2/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 200
 log-adjacency-changes
 redistribute connected subnets
 network 10.14.14.14 0.0.0.0 area 200
 network 10.15.0.0 0.255.255.255 area 200
 network 10.16.0.0 0.255.255.255 area 200
```

CSC-PE1 Configuration

```
ip cef distributed
!
ip vrf vpn1
 rd 100:0
 route-target export 100:0
 route-target import 100:0
mpls label protocol ldp
no mpls aggregate-statistics
!
interface Loopback0
 ip address 10.11.11.11 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Loopback100
 ip vrf forwarding vpn1
 ip address 10.19.19.19 255.255.255.255
 no ip directed-broadcast
!
interface ATM1/1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
```

```
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/1/0.1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM3/0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM3/0/0.1 point-to-point
 ip vrf forwarding vpn1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 101 0 51 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 100
 log-adjacency-changes
 passive-interface ATM3/0/0.1
 passive-interface Loopback100
 network 10.11.11.11 0.0.0.0 area 100
 network 10.0.0.0 0.255.255.255 area 100
!
router ospf 200 vrf vpn1
 log-adjacency-changes
 redistribute bgp 100 metric-type 1 subnets
 network 10.19.19.19 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
!
router bgp 100
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor 10.12.12.12 remote-as 100
 neighbor 10.12.12.12 update-source Loopback0
 !
 address-family ipv4
 neighbor 10.12.12.12 activate
 neighbor 10.12.12.12 send-community extended
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.12.12.12 activate
 neighbor 10.12.12.12 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vpn1
 redistribute ospf 200 match internal external 1 external 2
 no auto-summary
 no synchronization
 exit-address-family
```

CSC-PE2 Configuration

```
ip cef distributed
!
ip vrf vpn1
```

```
 rd 100:0
 route-target export 100:0
 route-target import 100:0
mpls label protocol ldp
no mpls aggregate-statistics
!
interface Loopback0
 ip address 10.12.12.12 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Loopback100
 ip vrf forwarding vpn1
 ip address 10.20.20.20 255.255.255.255
 no ip directed-broadcast
!
interface ATM0/1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM0/1/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM3/0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM3/0/0.1 point-to-point
 ip vrf forwarding vpn1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 100
 log-adjacency-changes
 passive-interface ATM3/0/0.1
 passive-interface Loopback100
 network 10.12.12.12 0.0.0.0 area 100
 network 10.0.0.0 0.255.255.255 area 100
!
router ospf 200 vrf vpn1
 log-adjacency-changes
 redistribute bgp 100 metric-type 1 subnets
 network 10.20.20.20 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
!
router bgp 100
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor 10.11.11.11 remote-as 100
 neighbor 10.11.11.11 update-source Loopback0
 !
```

```
 address-family ipv4
 neighbor 10.11.11.11 activate
 neighbor 10.11.11.11 send-community extended
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.11.11.11 activate
 neighbor 10.11.11.11 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vpn1
 redistribute ospf 200 match internal external 1 external 2
 no auto-summary
 no synchronization
 exit-address-family
```

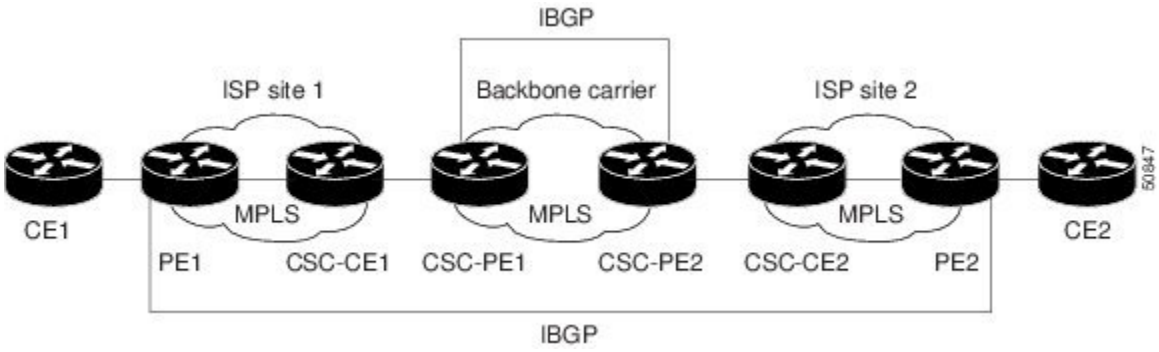
CSC-CE2 Configuration

```
ip cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address 10.16.16.16 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM5/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 200
 log-adjacency-changes
 redistribute connected subnets
 network 10.16.16.16 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
 network 10.0.0.0 0.255.255.255 area 200
```

MPLS VPN CSC Network with a Customer Who Is an MPLS VPN Provider Example

The figure below shows a carrier supporting carrier network configuration where the customer carrier is an MPLS VPN provider. The customer carrier has two sites. The backbone carrier and the customer carrier use MPLS. The IBGP sessions exchange the external routing information of the ISP.

Figure 20: Carrier Supporting Carrier Network with a Customer Carrier Who Is an MPLS VPN Provider



The following configuration examples show the configuration of each router in the carrier supporting carrier network. OSPF is the protocol used to connect the customer carrier to the backbone carrier.

CE1 Configuration

```
ip cef
!
interface Loopback0
 ip address 10.17.17.17 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0/1
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
!
router ospf 300
 log-adjacency-changes
 redistribute bgp 300 subnets
 passive-interface Ethernet0/1
 network 10.17.17.17 0.0.0.0 area 300
!
router bgp 300
 no synchronization
 bgp log-neighbor-changes
 timers bgp 10 30
 redistribute connected
 redistribute ospf 300 match internal external 1 external 2
 neighbor 10.0.0.1 remote-as 200
 neighbor 10.0.0.1 advertisement-interval 5
 no auto-summary
```

PE1 Configuration

```
ip cef
!
ip vrf vpn2
 rd 200:1
 route-target export 200:1
 route-target import 200:1
mpls label protocol ldp
!
interface Loopback0
 ip address 10.13.13.13 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface Ethernet3/0
 ip vrf forwarding vpn2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
router ospf 200
 log-adjacency-changes
 redistribute connected subnets
 passive-interface Ethernet3/0
 network 10.13.13.13 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
!
router bgp 200
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor 10.15.15.15 remote-as 200
 neighbor 10.15.15.15 update-source Loopback0
 !
 address-family ipv4
 neighbor 10.15.15.15 activate
 neighbor 10.15.15.15 send-community extended
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.15.15.15 activate
 neighbor 10.15.15.15 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vpn2
 neighbor 10.0.0.2 remote-as 300
 neighbor 10.0.0.2 activate
 neighbor 10.0.0.2 as-override
 neighbor 10.0.0.2 advertisement-interval 5
 no auto-summary
```

```
 no synchronization
 exit-address-family
```

CSC-CE1 Configuration

```
mpls label protocol ldp
!
interface Loopback0
 ip address 10.14.14.14 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 101 0 51 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM2/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM2/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 200
 log-adjacency-changes
 redistribute connected subnets
 network 10.14.14.14 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
 network 10.0.0.0 0.255.255.255 area 200
```

CSC-PE1 Configuration

```
ip cef distributed
!
ip vrf vpn1
 rd 100:0
 route-target export 100:0
 route-target import 100:0
mpls label protocol ldp
no mpls aggregate-statistics
!
interface Loopback0
 ip address 11.11.11.11 255.255.255.255
```

```
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Loopback100
 ip vrf forwarding vpn1
 ip address 10.19.19.19 255.255.255.255
 no ip directed-broadcast
!
interface ATM1/1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/1/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM3/0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM3/0/0.1 point-to-point
 ip vrf forwarding vpn1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 101 0 51 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 100
 log-adjacency-changes
 passive-interface ATM3/0/0.1
 passive-interface Loopback100
 network 10.11.11.11 0.0.0.0 area 100
 network 10.0.0.0 0.255.255.255 area 100
!
router ospf 200 vrf vpn1
 log-adjacency-changes
 redistribute bgp 100 metric-type 1 subnets
 network 10.19.19.19 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
!
router bgp 100
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor 10.12.12.12 remote-as 100
 neighbor 10.12.12.12 update-source Loopback0
 !
 address-family ipv4
 neighbor 10.12.12.12 activate
 neighbor 10.12.12.12 send-community extended
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.12.12.12 activate
 neighbor 10.12.12.12 send-community extended
 exit-address-family
 !
```

```
 address-family ipv4 vrf vpn1
 redistribute ospf 200 match internal external 1 external 2
 no auto-summary
 no synchronization
 exit-address-family
```

CSC-PE2 Configuration

```
ip cef distributed
!
ip vrf vpn1
 rd 100:0
 route-target export 100:0
 route-target import 100:0
mpls label protocol ldp
no mpls aggregate-statistics
!
interface Loopback0
 ip address 10.12.12.12 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Loopback100
 ip vrf forwarding vpn1
 ip address 10.20.20.20 255.255.255.255
 no ip directed-broadcast
!
interface ATM0/1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM0/1/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM3/0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM3/0/0.1 point-to-point
 ip vrf forwarding vpn1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 100
 log-adjacency-changes
 passive-interface ATM3/0/0.1
 passive-interface Loopback100
 network 10.12.12.12 0.0.0.0 area 100
```

```
 network 10.0.0.0 0.255.255.255 area 100
!
router ospf 200 vrf vpn1
 log-adjacency-changes
 redistribute bgp 100 metric-type 1 subnets
 network 10.20.20.20 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
!
router bgp 100
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor 10.11.11.11 remote-as 100
 neighbor 10.11.11.11 update-source Loopback0
 !
 address-family ipv4
 neighbor 10.11.11.11 activate
 neighbor 10.11.11.11 send-community extended
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.11.11.11 activate
 neighbor 10.11.11.11 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vpn1
 redistribute ospf 200 match internal external 1 external 2
 no auto-summary
 no synchronization
 exit-address-family
```

CSC-CE2 Configuration

```
ip cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address 10.16.16.16 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM5/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
```

```
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 200
 log-adjacency-changes
 redistribute connected subnets
 network 10.16.16.16 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
 network 10.0.0.0 0.255.255.255 area 200
```

PE2 Configuration

```
ip cef
ip cef accounting non-recursive
!
ip vrf vpn2
 rd 200:1
 route-target export 200:1
 route-target import 200:1
mpls label protocol ldp
!
interface Loopback0
 ip address 10.15.15.15 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet3/0
 ip vrf forwarding vpn2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
!
interface ATM5/0
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 200
 log-adjacency-changes
 redistribute connected subnets
 passive-interface Ethernet3/0
 network 10.15.15.15 0.0.0.0 area 200
 network 10.0.0.0 0.255.255.255 area 200
!
router bgp 200
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor 10.13.13.13 remote-as 200
 neighbor 10.13.13.13 update-source Loopback0
 !
 address-family ipv4
 neighbor 10.13.13.13 activate
 neighbor 10.13.13.13 send-community extended
 no synchronization
 exit-address-family
 !
```

```
 address-family vpnv4
 neighbor 10.13.13.13 activate
 neighbor 10.13.13.13 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vpn2
 neighbor 10.0.0.2 remote-as 300
 neighbor 10.0.0.2 activate
 neighbor 10.0.0.2 as-override
 neighbor 10.0.0.2 advertisement-interval 5
 no auto-summary
 no synchronization
 exit-address-family
```

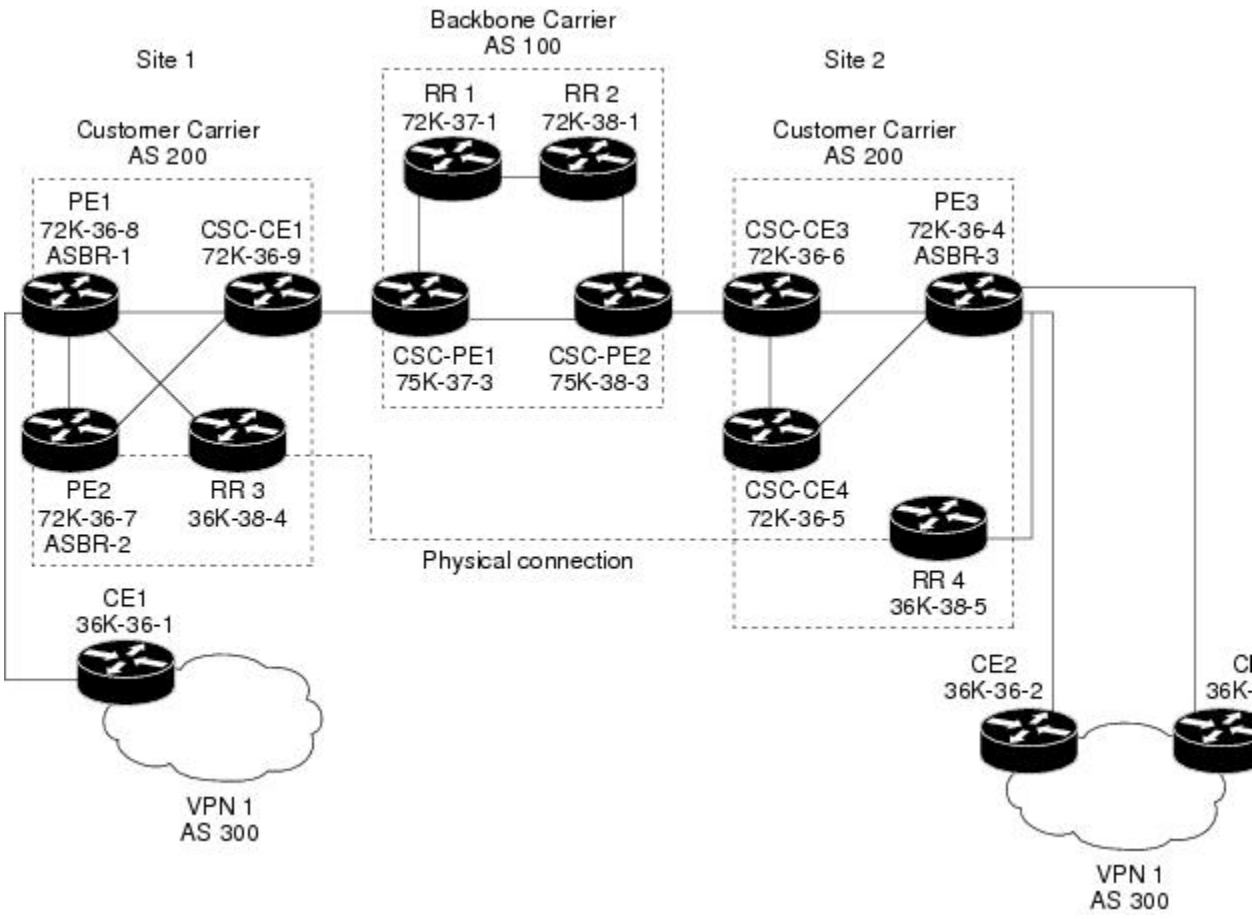
CE2 Configuration

```
ip cef
!
interface Loopback0
 ip address 10.18.18.18 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0/1
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
!
router ospf 300
 log-adjacency-changes
 redistribute bgp 300 subnets
 passive-interface Ethernet0/1
 network 10.18.18.18 0.0.0.0 area 300
!
router bgp 300
 no synchronization
 bgp log-neighbor-changes
 timers bgp 10 30
 redistribute connected
 redistribute ospf 300 match internal external 1 external 2
 neighbor 10.0.0.1 remote-as 200
 neighbor 10.0.0.1 advertisement-interval 5
 no auto-summary
```

MPLS VPN CSC Network That Contains Route Reflectors Example

The figure below shows a carrier supporting carrier network configuration that contains route reflectors. The customer carrier has two sites.

Figure 21: Carrier Supporting Carrier Network that Contains Route Reflectors



Note A connection between route reflectors (RRs) is not necessary.

The following configuration examples show the configuration of each router in the carrier supporting carrier network. Note the following:

- The router IP addresses are abbreviated for ease of reading. For example, the loopback address for PE 1 is 25, which is equivalent to 10.25.25.25.
- The following list shows the loopback addresses for the CSC-PE routers:
  - CSC-PE1 (75K-37-3): loopback 0 = 10.15.15.15, loopback 1 = 10.18.18.18
  - CSC-PE2 (75K-38-3): loopback 0 = 10.16.16.16, loopback 1 = 10.20.20.20

Backbone Carrier Configuration

Route Reflector 1 (72K-37-1) Configuration

```
interface Loopback0
 ip address 10.13.13.13 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/0.1 mpls
ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
interface ATM1/1
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/1.1 mpls
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
router ospf 100
 auto-cost reference-bandwidth 10000
 network 10.0.0.0 0.255.255.255 area 100
 network 10.1.0.0 0.255.255.255 area 100
 network 10.2.0.0 0.255.255.255 area 100
!
router bgp 100
 no synchronization
 no bgp default ipv4-unicast
 bgp cluster-id 1
 redistribute static
 neighbor 10.15.15.15 remote-as 100
 neighbor 10.15.15.15 update-source Loopback0
 neighbor 10.16.16.16 remote-as 100
 neighbor 10.16.16.16 update-source Loopback0
 !
 address-family ipv4 vrf vpn1
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.15.15.15 activate
 neighbor 10.15.15.15 route-reflector-client
 neighbor 10.15.15.15 send-community extended
 neighbor 10.16.16.16 activate
 neighbor 10.16.16.16 route-reflector-client
 neighbor 10.16.16.16 send-community extended
```

```
 bgp scan-time import 5
 exit-address-family
```

Route Reflector 2 (72K-38-1) Configuration

```
interface Loopback0
 ip address 10.14.14.14 255.255.255.255
 no ip directed-broadcast
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/0.1 mpls
 ip address 10.0.0.1 255.0.0.0
no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
interface ATM1/1
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/1.1 mpls
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
router ospf 100
 auto-cost reference-bandwidth 10000
 network 10.0.0.0 0.255.255.255 area 100
 network 10.1.0 0.255.255.255 area 100
 network 10.2.0.0 0.255.255.255 area 100
!
router bgp 100
 no synchronization
 no bgp default ipv4-unicast
 bgp cluster-id 1
 redistribute static
 neighbor 10.15.15.15 remote-as 100
 neighbor 10.15.15.15 update-source Loopback0
 neighbor 10.16.16.16 remote-as 100
 neighbor 10.16.16.16 update-source Loopback0
 !
 address-family ipv4 vrf vpn1
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.15.15.15 activate
 neighbor 10.15.15.15 route-reflector-client
 neighbor 10.15.15.15 send-community extended
 neighbor 10.16.16.16 activate
 neighbor 10.16.16.16 route-reflector-client
 neighbor 10.16.16.16 send-community extended
 bgp scan-time import 5
 exit-address-family
```

CSC-PE1 (75K-37-3) Configuration

```
ip cef distributed
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
interface Loopback0
 ip address 10.15.15.15 255.255.255.255
 no ip directed-broadcast
!
interface Loopback1
 ip vrf forwarding vpn1
 ip address 10.18.18.18 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0/0/1
 ip vrf forwarding vpn1
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip route-cache distributed
 mpls label protocol ldp
 mpls ip
!
interface ATM1/1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM1/1/0.1 mpls
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
interface ATM3/0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM3/0/0.1 point-to-point
 ip vrf forwarding vpn1
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 6 32 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM3/1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
```

```
interface ATM3/1/0.1 mpls
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
router ospf 100
 auto-cost reference-bandwidth 10000
 network 10.0.0.0 0.255.255.255 area 100
 network 10.1.0.0 0.255.255.255 area 100
 network 10.2.0.0 0.255.255.255 area 100
 network 10.3.0.0 0.255.255.255 area 100
 network 10.4.0.0 0.255.255.255 area 100
!
router ospf 1 vrf vpn1
 redistribute bgp 100 metric-type 1 subnets
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 10.13.13.13 remote-as 100
 neighbor 10.13.13.13 update-source Loopback0
 neighbor 10.14.14.14 remote-as 100
 neighbor 10.14.14.14 update-source Loopback0
 !
 address-family ipv4
 redistribute static
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.13.13.13 activate
 neighbor 10.13.13.13 send-community extended
 neighbor 10.14.14.14 activate
 neighbor 10.14.14.14 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vpn1
 redistribute ospf 1 match internal external 1 external 2
 no auto-summary
 no synchronization
 exit-address-family
```

CSC-PE2 (75K-38-3) Configuration

```
ip cef distributed
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
interface Loopback0
 ip address 10.16.16.16 255.255.255.255
 no ip directed-broadcast
!
interface Loopback1
 ip vrf forwarding vpn1
 ip address 10.20.20.20 255.255.255.255
 no ip directed-broadcast
!
interface ATM0/1/0
 no ip address
 no ip directed-broadcast
```

```
 no ip route-cache distributed
 atm clock INTERNAL
atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM0/1/0.1 mpls
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
interface ATM2/1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM2/1/0.1 mpls
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
interface ATM3/0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM3/0/0.1 point-to-point
 ip vrf forwarding vpn1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 6 32 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
interface ATM3/1/0
 no ip address
 no ip directed-broadcast
no ip route-cache distributed
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM3/1/0.1 point-to-point
 ip vrf forwarding vpn1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 101 6 33 aal5snap
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls ip
!
router ospf 100
 auto-cost reference-bandwidth 10000
 network 10.0.0.0 0.255.255.255 area 100
 network 10.0.0.0 0.255.255.255 area 100
 network 10.0.0.0 0.255.255.255 area 100
 network 10.0.0.0 0.255.255.255 area 100
```

```
 network 10.0.0.0 0.255.255.255 area 100
!
router ospf 1 vrf vpn1
 redistribute bgp 100 metric-type 1 subnets
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 10.13.13.13 remote-as 100
 neighbor 10.13.13.13 update-source Loopback0
 neighbor 10.14.14.14 remote-as 100
 neighbor 10.14.14.14 update-source Loopback0
 !
 address-family ipv4
 redistribute static
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.13.13.13 activate
 neighbor 10.13.13.13 send-community extended
 neighbor 10.14.14.14 activate
 neighbor 10.14.14.14 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vpn1
 redistribute ospf 1 match internal external 1 external 2
 no auto-summary
 no synchronization
 exit-address-family
```

Customer Carrier Site 1 Configuration

PE1 (72K-36-8) Configuration

```
ip cef
!
ip vrf vpn2
 rd 200:1
 route-target export 200:1
 route-target import 200:1
no mpls ip propagate-ttl
!
interface Loopback0
 ip address 10.25.25.25 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
 mpls label protocol ldp
 mpls ip
!
interface Ethernet3/0
```

```
 ip vrf forwarding vpn2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
interface Ethernet3/1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 mpls label protocol ldp
 mpls ip
!
interface Ethernet3/2
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
mpls label protocol ldp
mpls ip
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
!
router bgp 200
 neighbor 10.22.22.22 remote-as 200
 neighbor 10.22.22.22 update-source Loopback0
 neighbor 10.23.23.23 remote-as 200
 neighbor 10.23.23.23 update-source Loopback0
 !
 address-family ipv4 vrf vpn2
 redistribute connected
 neighbor 10.0.0.2 remote-as 300
 neighbor 10.0.0.2 activate
 neighbor 10.0.0.2 as-override
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.22.22.22 activate
 neighbor 10.22.22.22 send-community extended
 neighbor 10.23.23.23 activate
 neighbor 10.23.23.23 send-community extended
 exit-address-family
```

CSC-CE1 (72K-36-9) Configuration

```
ip cef
no ip domain-lookup
!
interface Loopback0
 ip address 10.11.11.11 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 6 32 aal5snap
mpls label protocol ldp
```

```
mpls ip
!
interface ATM2/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM2/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 50 aal5snap
mpls label protocol ldp
mpls ip
!
interface Ethernet3/0
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
mpls label protocol ldp
mpls ip
!
interface Ethernet3/1
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
mpls label protocol ldp
mpls ip
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
```

PE2 (72K-36-7) Configuration

```
ip cef
!
ip vrf vpn2
 rd 200:1
 route-target export 200:1
 route-target import 200:1
no mpls ip propagate-ttl
!
interface Loopback0
 ip address 10.24.24.24 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Ethernet3/0
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
mpls label protocol ldp
mpls ip
!
interface Ethernet3/1
 ip vrf forwarding vpn2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
interface Ethernet3/2
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
```

```
mpls label protocol ldp
mpls ip
!
interface Ethernet3/3
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
mpls label protocol ldp
mpls ip
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
!
router bgp 200
 neighbor 10.22.22.22 remote-as 200
 neighbor 10.22.22.22 update-source Loopback0
 neighbor 10.23.23.23 remote-as 200
 neighbor 10.23.23.23 update-source Loopback0
 !
 address-family ipv4 vrf vpn2
 neighbor 10.0.0.2 remote-as 300
 neighbor 10.0.0.2 activate
 neighbor 10.0.0.2 as-override
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.22.22.22 activate
 neighbor 10.22.22.22 send-community extended
 neighbor 10.23.23.23 activate
 neighbor 10.23.23.23 send-community extended
 exit-address-family
```

Route Reflector 3 (36K-38-4) Configuration

```
ip cef
!
interface Loopback0
 ip address 10.23.23.23 255.255.255.255
!
interface Ethernet1/1
 ip address 10.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
!
interface Ethernet1/2
 ip address 10.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
!
interface ATM3/0
 no ip address
 no ip mroute-cache
 atm clock INTERNAL
 no atm scrambling cell-payload
 no atm ilmi-keepalive
!
interface ATM3/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 atm pvc 100 0 55 aal5snap
 mpls label protocol ldp
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 101
```

```
 network 10.1.0.0 0.255.255.255 area 101
 network 10.2.0.0 0.255.255.255 area 101
 network 10.3.0.0 0.255.255.255 area 101
!
router bgp 200
 no synchronization
 no bgp default ipv4-unicast
 bgp cluster-id 2
 redistribute static
 neighbor 10.21.21.21 remote-as 200
 neighbor 10.21.21.21 update-source Loopback0
 neighbor 10.24.24.24 remote-as 200
 neighbor 10.24.24.24 update-source Loopback0
 neighbor 10.25.25.25 remote-as 200
 neighbor 10.25.25.25 update-source Loopback0
 !
 address-family ipv4 vrf vpn2
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.21.21.21 activate
 neighbor 10.21.21.21 route-reflector-client
 neighbor 10.21.21.21 send-community extended
 neighbor 10.24.24.24 activate
 neighbor 10.24.24.24 route-reflector-client
 neighbor 10.24.24.24 send-community extended
 neighbor 10.25.25.25 activate
 neighbor 10.25.25.25 route-reflector-client
 neighbor 10.25.25.25 send-community extended
 exit-address-family
```

CE1 (36K-36-1) Configuration

```
ip cef
!
interface Loopback0
 ip address 10.28.28.28 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0/1
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
!
interface Ethernet0/2
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
!
router bgp 300
 network 10.0.0.0
 network 10.0.0.0
 network 10.0.0.0
 neighbor 10.0.0.1 remote-as 200
 neighbor 10.0.0.1 remote-as 200
```

Customer Carrier Site 2 Configuration

CSC-CE3 (72K-36-6) Configuration

```
ip cef
!
interface Loopback0
 ip address 10.12.12.12 255.255.255.255
 no ip directed-broadcast
```

```
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 6 32 aal5snap
mpls label protocol ldp
 mpls ip
!
interface POS2/0
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 encapsulation ppp
mpls label protocol ldp
mpls ip
!
interface ATM5/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 40 aal5snap
 mpls ip
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 101
 network 10.1.0.0 0.255.255.255 area 101
 network 10.2.0.0 0.255.255.255 area 101
 network 10.3.0.0 0.255.255.255 area 101
```

PE3 (72K-36-4) Configuration

```
ip cef
!
ip vrf vpn2
 rd 200:1
 route-target export 200:1
 route-target import 200:1
!
!
interface Loopback0
 ip address 10.21.21.21 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet3/0
 ip vrf forwarding vpn2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
!
interface Ethernet3/1
 ip vrf forwarding vpn2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
!
interface Ethernet3/2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
```

```
mpls label protocol ldp
mpls ip
!
interface ATM5/0
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 40 aal5snap
mpls label protocol ldp
mpls ip
!
interface ATM6/0
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM6/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 20 aal5snap
mpls label protocol ldp
mpls ip
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 101
 network 10.1.0.0 0.255.255.255 area 101
 network 10.2.0.0 0.255.255.255 area 101
 network 10.3.0.0 0.255.255.255 area 101
!
router bgp 200
 neighbor 10.22.22.22 remote-as 200
 neighbor 10.22.22.22 update-source Loopback0
 neighbor 10.23.23.23 remote-as 200
 neighbor 10.23.23.23 update-source Loopback0
 !
 address-family ipv4 vrf vpn2
 redistribute connected
 neighbor 10.0.0.2 remote-as 300
 neighbor 10.0.0.2 activate
 neighbor 10.0.0.2 as-override
 neighbor 10.0.0.2 remote-as 300
 neighbor 10.0.0.2 activate
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.22.22.22 activate
 neighbor 10.22.22.22 send-community extended
 neighbor 10.23.23.23 activate
 neighbor 10.23.23.23 send-community extended
 exit-address-family
```

CSC-CE4 (72K-36-5) Configuration

```
ip cef
!
interface Loopback0
 ip address 10.10.10.10 255.255.255.255
 no ip directed-broadcast
!
interface POS4/0
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
```

```
 encapsulation ppp
mpls label protocol ldp
mpls ip
 clock source internal
!
interface ATM5/0
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 0 20 aal5snap
mpls label protocol ldp
mpls ip
!
interface ATM6/0
 no ip address
 no ip directed-broadcast
 atm clock INTERNAL
 no atm ilmi-keepalive
!
interface ATM6/0.1 point-to-point
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 atm pvc 100 6 33 aal5snap
mpls label protocol ldp
mpls ip
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 101
 network 10.1.0.0 0.255.255.255 area 101
 network 10.2.0.0 0.255.255.255 area 101
 network 10.3.0.0 0.255.255.255 area 101
```

Route Reflector 4 (36K-38-5) Configuration

```
ip cef
!
interface Loopback0
 ip address 10.22.22.22 255.255.255.255
!
interface Ethernet0/1
 ip address 10.0.0.2 255.0.0.0
 mpls label protocol ldp
 mpls ip
!
interface ATM2/0
 no ip address
 no ip mroute-cache
 atm clock INTERNAL
 no atm scrambling cell-payload
 no atm ilmi-keepalive
!
interface ATM2/0.1 point-to-point
 ip address 10.0.0.1 255.0.0.0
 atm pvc 100 0 55 aal5snap
 mpls label protocol ldp
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 101
 network 10.1.0.0 0.255.255.255 area 101
 network 10.2.0.0 0.255.255.255 area 101
!
router bgp 200
 no synchronization
```

```
 no bgp default ipv4-unicast
 bgp cluster-id 2
 redistribute static
 neighbor 10.21.21.21 remote-as 200
 neighbor 10.21.21.21 update-source Loopback0
 neighbor 10.24.24.24 remote-as 200
 neighbor 10.24.24.24 update-source Loopback0
 neighbor 10.25.25.25 remote-as 200
 neighbor 10.25.25.25 update-source Loopback0
 !
 address-family ipv4 vrf vpn2
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.21.21.21 activate
 neighbor 10.21.21.21 route-reflector-client
 neighbor 10.21.21.21 send-community extended
 neighbor 10.24.24.24 activate
 neighbor 10.24.24.24 route-reflector-client
 neighbor 10.24.24.24 send-community extended
 neighbor 10.25.25.25 activate
 neighbor 10.25.25.25 route-reflector-client
 neighbor 10.25.25.25 send-community extended
 exit-address-family
```

CE2 (36K-36-2) Configuration

```
ip cef
!
interface Loopback0
 ip address 10.26.26.26 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0/1
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
!
interface Ethernet0/2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
!
router ospf 300
 redistribute bgp 300
 network 10.0.0.0 0.255.255.255 area 300
 network 10.0.0.0 0.255.255.255 area 300
!
router bgp 300
 network 10.0.0.0
 network 10.1.0.0
 network 10.2.0.0
 neighbor 10.0.0.1 remote-as 200
```

CE3 (36K-36-3) Configuration

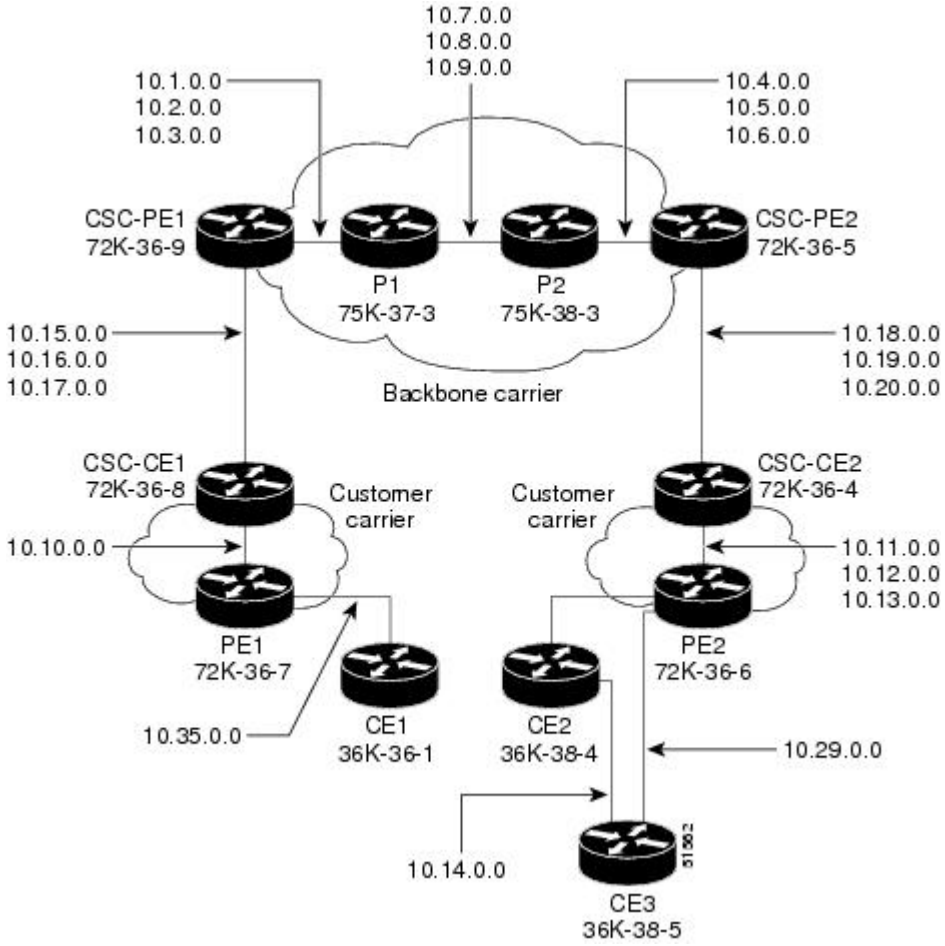
```
ip cef
!
interface Loopback0
 ip address 10.27.27.27 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet1/1
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
!
interface Ethernet1/2
 ip address 10.0.0.2 255.0.0.0
```

```
 no ip directed-broadcast
!
router ospf 300
 redistribute bgp 300
 network 10.0.0.0 0.255.255.255 area 300
 network 10.0.0.0 0.255.255.255 area 300
!
router bgp 300
 network 10.0.0.0
 network 10.1.0.0
 network 10.2.0.0
 neighbor 10.0.0.1 remote-as 200
```

MPLS VPN CSC Network with a Customer Who Has VPNs at the Network Edge Example

The figure below shows a carrier supporting carrier network configuration where the customer carrier has VPNs at the network edge.

Figure 22: Carrier Supporting Carrier Network



Backbone Carrier Configuration

CSC-PE1 (72K-36-9) Configuration

```
ip cef
!
ip vrf vpn1
rd 100:0
route-target export 100:0
route-target import 100:0
mpls label protocol ldp
!
!
interface Loopback0
ip address 10.14.14.14 255.255.255.255
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
!
interface Loopback100
ip vrf forwarding vpn1
ip address 10.22.22.22 255.255.255.255
no ip directed-broadcast
!
interface ATM1/0
no ip address
no ip directed-broadcast
no ip mroute-cache
atm clock INTERNAL
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
ip address 10.1.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM1/0.2 point-to-point
ip address 10.2.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM1/0.3 point-to-point
ip address 10.3.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM2/0
no ip address
no ip directed-broadcast
no ip mroute-cache
atm clock INTERNAL
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM2/0.1 point-to-point
ip vrf forwarding vpn1
ip address 10.15.0.2 255.255.0.0
no ip directed-broadcast
```

```
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM2/0.2 point-to-point
ip vrf forwarding vpn1
ip address 10.16.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM2/0.3 point-to-point
ip vrf forwarding vpn1
ip address 10.17.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
router ospf 100
log-adjacency-changes
redistribute connected subnets
passive-interface ATM2/0.1
passive-interface ATM2/0.2
passive-interface ATM2/0.3
passive-interface Loopback100
network 10.14.14.14 0.0.0.0 area 100
network 10.1.0.0 0.0.255.255 area 100
network 10.2.0.0 0.0.255.255 area 100
network 10.3.0.0 0.0.255.255 area 100
!
router ospf 200 vrf vpn1
log-adjacency-changes
redistribute connected subnets
redistribute bgp 100 metric-type 1 subnets
network 10.22.22.22 0.0.0.0 area 200
network 10.15.0.0 0.0.255.255 area 200
network 10.16.0.0 0.0.255.255 area 200
network 10.17.0.0 0.0.255.255 area 200
!
router bgp 100
bgp log-neighbor-changes
timers bgp 10 30
neighbor 10.11.11.11 remote-as 100
neighbor 10.11.11.11 update-source Loopback0
!
address-family ipv4
neighbor 10.11.11.11 activate
neighbor 10.11.11.11 send-community extended
no synchronization
exit-address-family
!
address-family vpnv4
neighbor 10.11.11.11 activate
neighbor 10.11.11.11 send-community extended
exit-address-family
!
address-family ipv4 vrf vpn1
redistribute ospf 200 match internal external 1 external 2
no auto-summary
no synchronization
exit-address-family
```

P1 (75K-37-3) Configuration

```
ip cef distributed
```

```
!
mpls label protocol ldp
!
interface Loopback0
ip address 10.12.12.12 255.255.255.255
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
!
interface ATM1/1/0
no ip address
no ip directed-broadcast
ip route-cache distributed
atm clock INTERNAL
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM1/1/0.1 point-to-point
ip address 10.7.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 103 0 53 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM1/1/0.2 point-to-point
ip address 10.8.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 104 0 54 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM1/1/0.3 point-to-point
ip address 10.9.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 105 0 55 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM3/0/0
no ip address
no ip directed-broadcast
ip route-cache distributed
atm clock INTERNAL
atm sonet stm-1
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM3/0/0.1 point-to-point
ip address 10.1.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
mpls accounting experimental input
tag-switching ip
!
interface ATM3/0/0.2 point-to-point
ip address 10.2.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM3/0/0.3 point-to-point
ip address 10.3.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
```

```
tag-switching ip
!
router ospf 100
log-adjacency-changes
redistribute connected subnets
network 10.12.12.12 0.0.0.0 area 100
network 10.1.0.0 0.0.255.255 area 100
network 10.2.0.0 0.0.255.255 area 100
network 10.3.0.0 0.0.255.255 area 100
network 10.7.0.0 0.0.255.255 area 100
network 10.8.0.0 0.0.255.255 area 100
network 10.9.0.0 0.0.255.255 area 100
```

P2 (75K-38-3) Configuration

```
ip cef distributed
!
mpls label protocol ldp
!
interface Loopback0
ip address 10.13.13.13 255.255.255.255
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
!
interface ATM0/1/0
no ip address
no ip directed-broadcast
ip route-cache distributed
atm clock INTERNAL
atm sonet stm-1
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM0/1/0.1 point-to-point
ip address 10.7.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 103 0 53 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM0/1/0.2 point-to-point
ip address 10.8.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 104 0 54 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM0/1/0.3 point-to-point
ip address 10.9.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 105 0 55 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM3/1/0
no ip address
no ip directed-broadcast
ip route-cache distributed
atm clock INTERNAL
atm sonet stm-1
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM3/1/0.1 point-to-point
ip address 10.4.0.2 255.255.0.0
no ip directed-broadcast
```

```
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM3/1/0.2 point-to-point
ip address 10.5.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM3/1/0.3 point-to-point
ip address 10.6.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
router ospf 100
log-adjacency-changes
redistribute connected subnets
network 10.13.13.13 0.0.0.0 area 100
network 10.4.0.0 0.0.255.255 area 100
network 10.5.0.0 0.0.255.255 area 100
network 10.6.0.0 0.0.255.255 area 100
network 10.7.0.0 0.0.255.255 area 100
network 10.8.0.0 0.0.255.255 area 100
network 10.9.0.0 0.0.255.255 area 100
!
```

CSC-PE2 (72K-36-5) Configuration

```
ip cef
!
ip vrf vpn1
rd 100:0
route-target export 100:0
route-target import 100:0
mpls label protocol ldp
!
interface Loopback0
ip address 10.11.11.11 255.255.255.255
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
!
interface Loopback100
ip vrf forwarding vpn1
ip address 10.23.23.23 255.255.255.255
no ip directed-broadcast
!
interface ATM5/0
no ip address
no ip directed-broadcast
no ip mroute-cache
atm clock INTERNAL
atm sonet stm-1
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
ip vrf forwarding vpn1
ip address 10.18.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
```

```
tag-switching ip
!
interface ATM5/0.2 point-to-point
ip vrf forwarding vpn1
ip address 10.19.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM5/0.3 point-to-point
ip vrf forwarding vpn1
ip address 10.20.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM6/0
no ip address
no ip directed-broadcast
no ip mroute-cache
atm clock INTERNAL
atm sonet stm-1
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM6/0.1 point-to-point
ip address 10.4.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM6/0.2 point-to-point
ip address 10.5.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM6/0.3 point-to-point
ip address 10.6.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
router ospf 100
log-adjacency-changes
redistribute connected subnets
passive-interface ATM5/0.1
passive-interface ATM5/0.2
passive-interface ATM5/0.3
passive-interface Loopback100
network 10.11.11.11 0.0.0.0 area 100
network 10.4.0.0 0.0.255.255 area 100
network 10.5.0.0 0.0.255.255 area 100
network 10.6.0.0 0.0.255.255 area 100
!
router ospf 200 vrf vpn1
log-adjacency-changes
redistribute connected subnets
redistribute bgp 100 metric-type 1 subnets
network 10.23.23.23 0.0.0.0 area 200
network 10.18.0.0 0.0.255.255 area 200
network 10.19.0.0 0.0.255.255 area 200
network 10.20.0.0 0.0.255.255 area 200
```

```
!
router bgp 100
bgp log-neighbor-changes
timers bgp 10 30
neighbor 10.14.14.14 remote-as 100
neighbor 10.14.14.14 update-source Loopback0
!
address-family ipv4
neighbor 10.14.14.14 activate
neighbor 10.14.14.14 send-community extended
no synchronization
exit-address-family
!
address-family vpnv4
neighbor 10.14.14.14 activate
neighbor 10.14.14.14 send-community extended
exit-address-family
!
address-family ipv4 vrf vpn1
redistribute ospf 200 match internal external 1 external 2
no auto-summary
no synchronization
exit-address-family
```

Customer Carrier Site 1 Configuration

CSC-CE1 (72K-36-8) Configuration

```
ip cef
!
mpls label protocol ldp
!
interface Loopback0
ip address 10.15.15.15 255.255.255.255
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
!
interface ATM1/0
no ip address
no ip directed-broadcast
no ip mroute-cache
atm clock INTERNAL
atm sonet stm-1
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
ip address 10.15.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM1/0.2 point-to-point
ip address 10.16.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM1/0.3 point-to-point
ip address 10.17.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
```

```
mpls label protocol ldp
tag-switching ip
!
interface Ethernet3/1
ip address 10.10.0.2 255.255.0.0
no ip directed-broadcast
no ip mroute-cache
mpls label protocol ldp
tag-switching ip
!
router ospf 200
log-adjacency-changes
redistribute connected subnets
network 10.15.15.15 0.0.0.0 area 200
network 10.10.0.0 0.0.255.255 area 200
network 10.15.0.0 0.0.255.255 area 200
network 10.16.0.0 0.0.255.255 area 200
network 10.17.0.0 0.0.255.255 area 200
```

PE2 (72K-36-7) Configuration

```
ip cef
!
ip vrf vpn2
 rd 200:1
 route-target export 200:1
 route-target import 200:1
no mpls ip propagate-ttl
!
interface Loopback0
 ip address 10.24.24.24 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Ethernet3/0
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
mpls label protocol ldp
mpls ip
!
interface Ethernet3/1
 ip vrf forwarding vpn2
 ip address 10.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
interface Ethernet3/2
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
mpls label protocol ldp
mpls ip
!
interface Ethernet3/3
 ip address 10.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
mpls label protocol ldp
mpls ip
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
 network 10.0.0.0 0.255.255.255 area 101
!
router bgp 200
 neighbor 10.22.22.22 remote-as 200
```

```
 neighbor 10.22.22.22 update-source Loopback0
 neighbor 10.23.23.23 remote-as 200
 neighbor 10.23.23.23 update-source Loopback0
 !
 address-family ipv4 vrf vpn2
 neighbor 10.0.0.2 remote-as 300
 neighbor 10.0.0.2 activate
 neighbor 10.0.0.2 as-override
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 10.22.22.22 activate
 neighbor 10.22.22.22 send-community extended
 neighbor 10.23.23.23 activate
 neighbor 10.23.23.23 send-community extended
 exit-address-family
```

CE1 (36K-36-1) Configuration

```
ip cef
!
interface Loopback0
ip address 10.19.19.19 255.255.255.255
no ip directed-broadcast
!
interface Ethernet0/2
ip address 30.35.0.1 255.255.0.0
no ip directed-broadcast
!
router ospf 300
log-adjacency-changes
redistribute connected subnets
redistribute bgp 300 subnets
passive-interface Ethernet0/2
network 10.19.19.19 0.0.0.0 area 300
!
router bgp 300
no synchronization
bgp log-neighbor-changes
timers bgp 10 30
redistribute connected
redistribute ospf 300 match internal external 1 external 2
neighbor 10.35.0.2 remote-as 200
neighbor 10.35.0.2 advertisement-interval 5
no auto-summary
```

Customer Carrier Site 2 Configuration

CSC-CE2 (72K-36-4) Configuration

```
ip cef
!
mpls label protocol ldp
!
interface Loopback0
ip address 10.17.17.17 255.255.255.255
no ip directed-broadcast
!
interface ATM5/0
no ip address
no ip directed-broadcast
atm clock INTERNAL
atm sonet stm-1
```

```
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
ip address 10.11.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM5/0.2 point-to-point
ip address 10.12.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM5/0.3 point-to-point
ip address 10.13.0.2 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM6/0
no ip address
no ip directed-broadcast
atm clock INTERNAL
atm sonet stm-1
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM6/0.1 point-to-point
ip address 10.18.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM6/0.2 point-to-point
ip address 10.19.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM6/0.3 point-to-point
ip address 10.20.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
router ospf 200
log-adjacency-changes
redistribute connected subnets
network 10.17.17.17 0.0.0.0 area 200
network 10.11.0.0 0.0.255.255 area 200
network 10.12.0.0 0.0.255.255 area 200
network 10.13.0.0 0.0.255.255 area 200
network 10.18.0.0 0.0.255.255 area 200
network 10.19.0.0 0.0.255.255 area 200
network 10.20.0.0 0.0.255.255 area 200
```

PE2 (72K-36-6) Configuration

```
ip cef
!
ip vrf customersite
rd 200:1
route-target export 200:1
route-target import 200:1
mpls label protocol ldp
!
interface Loopback0
ip address 10.18.18.18 255.255.255.255
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
!
interface Ethernet3/0
ip vrf forwarding customersite
ip address 10.29.0.2 255.255.0.0
no ip directed-broadcast
!
interface Ethernet3/1
ip vrf forwarding customersite
ip address 10.30.0.2 255.255.0.0
no ip directed-broadcast
!
interface ATM5/0
no ip address
no ip directed-broadcast
no ip mroute-cache
atm clock INTERNAL
atm sonet stm-1
no atm enable-ilmi-trap
no atm ilmi-keepalive
!
interface ATM5/0.1 point-to-point
ip address 10.11.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 100 0 50 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM5/0.2 point-to-point
ip address 10.12.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 101 0 51 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
interface ATM5/0.3 point-to-point
ip address 10.13.0.1 255.255.0.0
no ip directed-broadcast
atm pvc 102 0 52 aal5snap
no atm enable-ilmi-trap
mpls label protocol ldp
tag-switching ip
!
router ospf 200
log-adjacency-changes
redistribute connected subnets
passive-interface Ethernet3/0
passive-interface Ethernet3/1
network 10.18.18.18 0.0.0.0 area 200
network 10.11.0.0 0.0.255.255 area 200
network 10.12.0.0 0.0.255.255 area 200
network 10.13.0.0 0.0.255.255 area 200
!
router bgp 200
```

```
no bgp default ipv4-unicast
bgp log-neighbor-changes
timers bgp 10 30
neighbor 10.16.16.16 remote-as 200
neighbor 10.16.16.16 update-source Loopback0
!
address-family ipv4
neighbor 10.16.16.16 activate
neighbor 10.16.16.16 send-community extended
no synchronization
exit-address-family
!
address-family vpnv4
neighbor 10.16.16.16 activate
neighbor 10.16.16.16 send-community extended
exit-address-family
!
address-family ipv4 vrf customersite
neighbor 10.29.0.1 remote-as 300
neighbor 10.29.0.1 activate
neighbor 10.29.0.1 as-override
neighbor 10.29.0.1 advertisement-interval 5
neighbor 10.30.0.1 remote-as 300
neighbor 10.30.0.1 activate
neighbor 10.30.0.1 as-override
neighbor 10.30.0.1 advertisement-interval 5
no auto-summary
no synchronization
exit-address-family
```

CE2 (36K-38-4) Configuration

```
ip cef
!
interface Loopback0
ip address 10.21.21.21 255.255.255.255
!
interface Ethernet1/3
ip address 10.29.0.1 255.255.0.0
!
interface Ethernet5/0
ip address 10.14.0.1 255.255.0.0
!
router ospf 300
log-adjacency-changes
redistribute connected subnets
redistribute bgp 300 subnets
passive-interface Ethernet1/3
network 10.21.21.21 0.0.0.0 area 300
network 10.14.0.0 0.0.255.255 area 300
!
router bgp 300
no synchronization
timers bgp 10 30
redistribute connected
redistribute ospf 300 match internal external 1 external 2
neighbor 10.29.0.2 remote-as 200
neighbor 10.29.0.2 advertisement-interval 5
no auto-summary
```

CE3 (36K-38-5) Configuration

```
ip cef
!
interface Loopback0
ip address 10.20.20.20 255.255.255.255
no ip directed-broadcast
!
```

```
interface Ethernet0/2
ip address 10.30.0.1 255.255.0.0
no ip directed-broadcast
!
interface Ethernet0/3
ip address 10.14.0.2 255.255.0.0
no ip directed-broadcast
!
router ospf 300
log-adjacency-changes
redistribute connected subnets
redistribute bgp 300 subnets
passive-interface Ethernet0/2
network 10.20.20.20 0.0.0.0 area 300
network 10.14.0.0 0.0.255.255 area 300
!
router bgp 300
no synchronization
bgp log-neighbor-changes
timers bgp 10 30
redistribute connected
redistribute ospf 300 match internal external 1 external 2
neighbor 10.30.0.2 remote-as 200
neighbor 10.30.0.2 advertisement-interval 5
no auto-summary
```

Additional References for MPLS VPN Carrier Supporting Carrier Using LDP and an IGP

Related Documents

Related Topic	Document Title
MPLS	MPLS Product Literature

RFCs

RFC	Title
RFC 2547	BGP/MPLS 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 CSC with LDP and IGP

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 CSC with LDP and IGP

Feature Name	Releases	Feature Configuration Information
MPLS VPN Carrier Supporting Carrier	12.0(14)ST 12.0(16)ST 12.2(8)T 12.0(21)ST 12.0(22)S 12.0(23)S Cisco IOS XE Release 2.2	This feature enables you to set up and create an MPLS VPN CSC network that uses LDP to transport MPLS labels and an IGP to transport routes. In 12.0(14)ST, this feature was introduced. In 12.0(16)ST, this feature was integrated. In 12.2(8)T, this feature was integrated. In 12.0(21)ST, this feature was integrated. In 12.0(22)S, this feature was integrated. In 12.0(23)S, this feature was integrated. In Cisco IOS XE Release 2.2, this feature was implemented on the Cisco ASR 1000 Series Routers. This feature uses no new or modified commands.

Glossary

ASBR -- Autonomous System Boundary router. A router that connects one autonomous system to another.

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).

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.

CSC --Carrier Supporting Carrier. A hierarchical VPN model that allows small service providers, or customer carriers, to interconnect their IP or MPLS networks over an MPLS backbone. This eliminates the need for customer carriers to build and maintain their own MPLS backbone.

eBGP --external 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.

edge router--A router that is at the edge of the network. It defines the boundary of the MPLS network. It receives and transmits packets. Also referred to as edge label switch router and label edge router.

iBGP --internal 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.

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.

LDP --Label Distribution Protocol. A standard protocol between MPLS-enabled routers to negotiate the labels (addresses) used to forward packets.

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.

MP-BGP --Multiprotocol BGP.

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.

NSF --Nonstop forwarding enables routers to continuously forward IP packets following a Route Processor takeover or switchover to another Route Processor. NSF maintains and updates Layer 3 routing and forwarding information in the backup Route Processor to ensure that IP packets and routing protocol information are forwarded continuously during the switchover and route convergence process.

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. All MPLS VPN processing occurs in the PE router.

QoS --quality of service. Measure of performance for a transmission system that indicates its transmission quality and service availability.

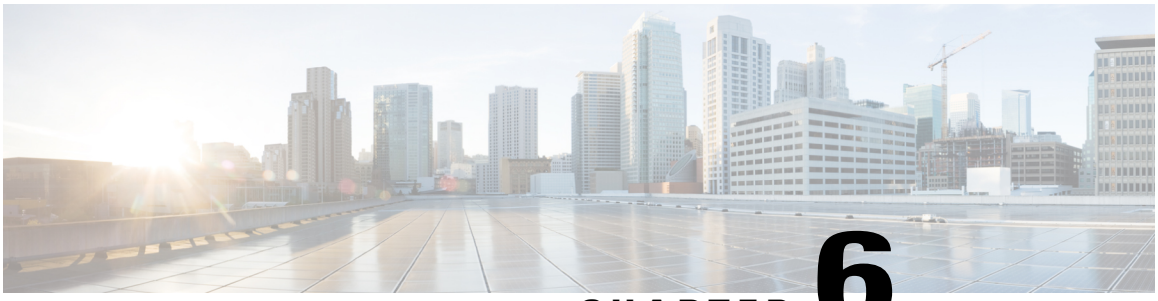
RD --route distinguisher. An 8-byte value that is concatenated with an IPv4 prefix to create a unique VPN-IPv4 prefix.

RT --route target. Extended community attribute used to identify the VRF routing table into which a prefix is imported.

SLA --Service Level Agreement given to VPN subscribers.

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 VPN site that is attached to a 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.



CHAPTER 6

MPLS VPN Carrier Supporting Carrier with BGP

Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) Carrier Supporting Carrier (CSC) enables one MPLS VPN-based service provider to allow other service providers to use a segment of its backbone network. This module explains how to configure an MPLS VPN CSC network that uses Border Gateway Protocol (BGP) to distribute routes and MPLS labels.

- Finding Feature Information, page 197
- Prerequisites for MPLS VPN CSC with BGP, page 198
- Restrictions for MPLS VPN CSC with BGP, page 198
- Information About MPLS VPN CSC with BGP, page 198
- How to Configure MPLS VPN CSC with BGP, page 201
- Configuration Examples for MPLS VPN CSC with BGP, page 230
- Additional References, page 243
- Feature Information for MPLS VPN CSC with BGP, page 244
- Glossary, page 245

Finding Feature Information

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

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 CSC with BGP

- You should be able to configure MPLS VPNs with end-to-end (CE-to-CE router) pings working. To accomplish this, you need to know how to configure Interior Gateway Protocols (IGPs), MPLS Label Distribution Protocol (LDP), and Multiprotocol Border Gateway Protocol (MP-BGP).
- Make sure that the CSC-PE routers and the CSC-CE routers run images that support BGP label distribution. Otherwise, you cannot run external BGP (EBGP) between them. Ensure that connectivity between the customer carrier and the backbone carrier. EBGP-based label distribution is configured on these links to enable MPLS between the customer and backbone carriers.

Restrictions for MPLS VPN CSC with BGP

On a provider edge (PE) router, you can configure an interface for either BGP with labels or LDP. You cannot enable both types of label distribution on the same interface. If you switch from one protocol to the other, then you must disable the existing protocol on all interfaces before enabling the other protocol.

This feature does not support the following:

- EBGP multihop between CSC-PE and CSC-CE routers
- EIBGP multipath load sharing

The physical interfaces that connect the BGP speakers must support Cisco Express Forwarding or distributed Cisco Express Forwarding and MPLS.

Information About MPLS VPN CSC with BGP

MPLS VPN CSC Introduction

Carrier supporting carrier is where one service provider allows another service provider to use a segment of its backbone network. The service provider that provides the segment of the backbone network to the other provider is called the backbone carrier. The service provider that uses the segment of the backbone network is called the customer carrier.

A backbone carrier offers Border Gateway Protocol and Multiprotocol Label Switching (BGP/MPLS) VPN services. The customer carrier can be either:

- An Internet service provider (ISP)
- A BGP/MPLS VPN service provider

Benefits of Implementing MPLS VPN CSC

The MPLS VPN CSC network provides the following benefits to service providers who are backbone carriers and to customer carriers.

Benefits to the Backbone Carrier

- The backbone carrier can accommodate many customer carriers and give them access to its backbone. The backbone carrier does not need to create and maintain separate backbones for its customer carriers. Using one backbone network to support multiple customer carriers simplifies the backbone carrier's VPN operations. The backbone carrier uses a consistent method for managing and maintaining the backbone network. This is also cheaper and more efficient than maintaining separate backbones.
- The MPLS VPN carrier supporting carrier feature is scalable. Carrier supporting carrier can change the VPN to meet changing bandwidth and connectivity needs. The feature can accommodate unplanned growth and changes. The carrier supporting carrier feature enables tens of thousands of VPNs to be set up over the same network, and it allows a service provider to offer both VPN and Internet services.
- The MPLS VPN carrier supporting carrier feature is a flexible solution. The backbone carrier can accommodate many types of customer carriers. The backbone carrier can accept customer carriers who are ISPs or VPN service providers or both. The backbone carrier can accommodate customer carriers that require security and various bandwidths.

Benefits to the Customer Carriers

- The MPLS VPN carrier supporting carrier feature removes from the customer carrier the burden of configuring, operating, and maintaining its own backbone. The customer carrier uses the backbone network of a backbone carrier, but the backbone carrier is responsible for network maintenance and operation.
- Customer carriers who use the VPN services provided by the backbone carrier receive the same level of security that Frame Relay or ATM-based VPNs provide. Customer carriers can also use IPSec in their VPNs for a higher level of security; it is completely transparent to the backbone carrier.
- Customer carriers can use any link layer technology (SONET, DSL, Frame Relay, and so on) to connect the CE routers to the PE routers and the PE routers to the P routers. The MPLS VPN carrier supporting carrier feature is link layer independent. The CE routers and PE routers use IP to communicate, and the backbone carrier uses MPLS.
- The customer carrier can use any addressing scheme and still be supported by a backbone carrier. The customer address space and routing information are independent of the address space and routing information of other customer carriers or the backbone provider.

Benefits of Implementing MPLS VPN CSC with BGP

You can configure your CSC network to enable BGP to transport routes and MPLS labels between the backbone carrier PE routers and the customer carrier CE routers using multiple paths. The benefits of using BGP to distribute IPv4 routes and MPLS label routes are:

- BGP takes the place of an IGP and LDP in a VPN forwarding/routing instance (VRF) table. You can use BGP to distribute routes and MPLS labels. Using a single protocol instead of two simplifies the configuration and troubleshooting.
- BGP is the preferred routing protocol for connecting two ISPs, mainly because of its routing policies and ability to scale. ISPs commonly use BGP between two providers. This feature enables those ISPs to use BGP.

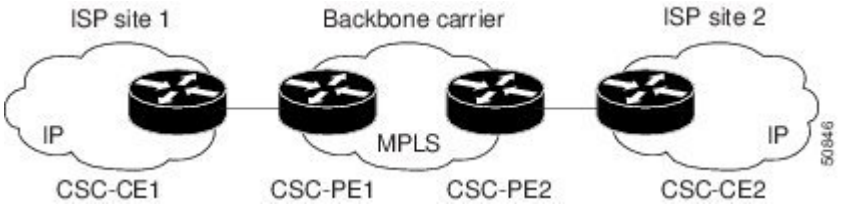
Configuration Options for MPLS VPN CSC with BGP

The following sections explain how the backbone and customer carriers distribute IPv4 routes and MPLS labels. The backbone carrier offers BGP and MPLS VPN services. The customer carrier can be either of the following:

Customer Carrier Is an ISP with an IP Core

The figure below shows a network configuration where the customer carrier is an ISP. The customer carrier has two sites, each of which is a point of presence (POP). The customer carrier connects these sites using a VPN service provided by the backbone carrier. The backbone carrier uses MPLS. The ISP sites use IP.

Figure 23: Network Where the Customer Carrier Is an ISP



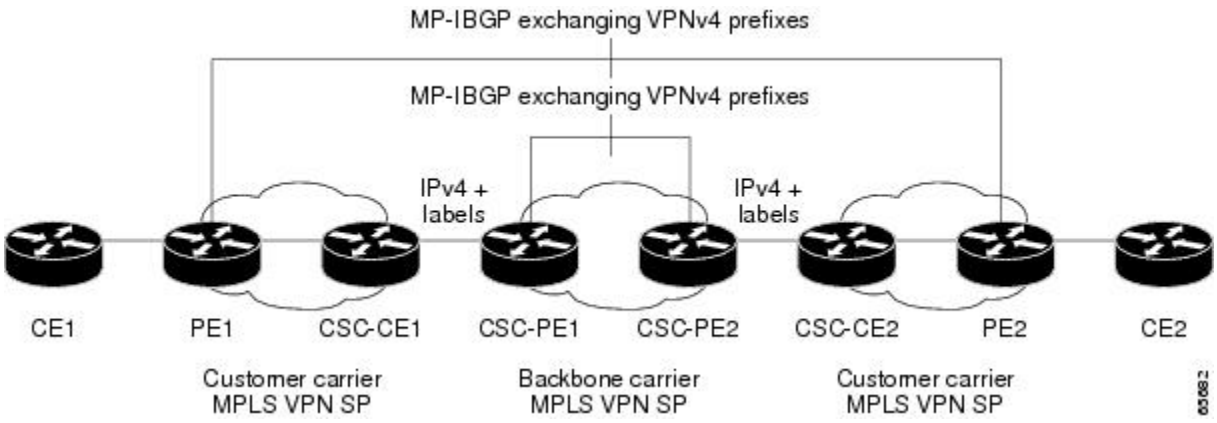
The links between the CE and PE routers use EBGP to distribute IPv4 routes and MPLS labels. Between the links, the PE routers use multiprotocol IBGP to distribute VPNv4 routes.

Note If a router other than a Cisco router is used as a CSC-PE or CSC-CE, that router must support IPv4 BGP label distribution (RFC 3107). Otherwise, you cannot run EBGP with labels between the routers.

Customer Carrier Is an MPLS Service Provider With or Without VPN Services

The figure below shows a network configuration where the backbone carrier and the customer carrier are BGP/MPLS VPN service providers. This is known as hierarchical VPNs. The customer carrier has two sites. Both the backbone carrier and the customer carrier use MPLS in their networks.

Figure 24: Network Where the Customer Carrier Is an MPLS VPN Service Provider



In this configuration, the customer carrier can configure its network in one of the following ways:

- The customer carrier can run IGP and LDP in its core network. In this case, the CSC-CE1 router in the customer carrier redistributes the EBGP routes it learns from the CSC-PE1 router of the backbone carrier to IGP.
- The CSC-CE1 router of the customer carrier system can run an IPv4 and labels IBGP session with the PE1 router.

How to Configure MPLS VPN CSC with BGP

Identifying the Carrier Supporting Carrier Topology

Before you configure the MPLS VPN CSC with BGP, you need to identify both the backbone and customer carrier topology.

For hierarchical VPNs, the customer carrier of the MPLS VPN network provides MPLS VPN services to its own customers. In this instance, you need to identify the type of customer carrier as well as the topology of the customer carriers. Hierarchical VPNs require extra configuration steps, which are noted in the configuration sections.

Note You can connect multiple CSC-CE routers to the same PE, or you can connect a single CSC-CE router to CSC-PEs using more than one interface to provide redundancy and multiple path support in CSC topology.

Perform this task to identify the carrier supporting carrier topology.

SUMMARY STEPS

1. Identify the type of customer carrier, ISP or MPLS VPN service provider.
2. (For hierarchical VPNs only) Identify the CE routers.
3. (For hierarchical VPNs only) Identify the customer carrier core router configuration.
4. Identify the customer carrier edge (CSC-CE) routers.
5. Identify the backbone carrier router configuration.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Identify the type of customer carrier, ISP or MPLS VPN service provider.	Sets up requirements for configuration of carrier supporting carrier network. • For an ISP, customer site configuration is not required. • For an MPLS VPN service provider, the customer site needs to be configured, as well as any task or step designated "for hierarchical VPNs only."
Step 2	(For hierarchical VPNs only) Identify the CE routers.	Sets up requirements for configuration of CE to PE connections.
Step 3	(For hierarchical VPNs only) Identify the customer carrier core router configuration.	Sets up requirements for connection configuration between core (P) routers and between P routers and edge routers (PE and CSC-CE routers).
Step 4	Identify the customer carrier edge (CSC-CE) routers.	Sets up requirements for configuration of CSC-CE to CSC-PE connections.
Step 5	Identify the backbone carrier router configuration.	Sets up requirements for connection configuration between CSC core routers and between CSC core routers and edge routers (CSC-CE and CSC-PE routers).

What to Do Next

Set up your carrier supporting carrier networks with the Configuring the Backbone Carrier Core, on page 202.

Configuring the Backbone Carrier Core

Configuring the backbone carrier core requires setting up connectivity and routing functions for the CSC core and the CSC-PE routers.

Configuring and verifying the CSC core (backbone carrier) involves the following tasks:

Prerequisites

Before you configure a backbone carrier core, configure the following on the CSC core routers:

- An IGP routing protocol--BGP, OSPF, IS-IS, EIGRP, static, and so on.
- Label Distribution Protocol (LDP). For information, see How to Configure MPLS LDP.

Verifying IP Connectivity and LDP Configuration in the CSC Core

Perform this task to verify IP connectivity and LDP configuration in the CSC core.

SUMMARY STEPS

1. **enable**
2. **ping** [*protocol*] {*host-name* | *system-address*}
3. **trace** [*protocol*] [*destination*]
4. **show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [- *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
5. **show mpls ldp discovery** [**vrf** *vrf-name* | **all**]
6. **show mpls ldp neighbor** [[**vrf** *vrf-name*] [*address* | *interface*] [**detail**] | **all**]
7. **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]
8. **show mpls interfaces** [[**vrf** *vrf-name*] [*interface*] [**detail**] | **all**]
9. **show ip route**
10. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping** [*protocol*] {*host-name*	*system-address*} **Example:** `Router# ping ip 10.1.0.0`	(Optional) Diagnoses basic network connectivity on AppleTalk, CLNS, IP, Novell, Apollo, VINES, DECnet, or XNS networks. • Use the **ping ip** command to verify the connectivity from one CSC core router to another.
Step 3	**trace** [*protocol*] [*destination*] **Example:** `Router# trace ip 10.2.0.0`	(Optional) Discovers the routes that packets will actually 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 or Action	Purpose
		command can help isolate a trouble spot if two routers cannot communicate.
Step 4	**show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [- *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:** `Router# show mpls forwarding-table`	(Optional) Displays the contents of the MPLS label forwarding information base (LFIB). • Use the **show mpls forwarding-table** command to verify that MPLS packets are being forwarded.
Step 5	**show mpls ldp discovery** [**vrf** *vrf-name* | **all**] **Example:** `Router# show mpls ldp discovery`	(Optional) Displays the status of the LDP discovery process. • Use the **show mpls ldp discovery** command to verify that LDP is operational in the CSC core.
Step 6	**show mpls ldp neighbor** [[**vrf** *vrf-name*] [*address* | *interface*] [**detail**] | **all**] **Example:** `Router# show mpls ldp neighbor`	(Optional) Displays the status of LDP sessions. • Use the **show mpls ldp neighbor** command to verify LDP configuration in the CSC core.
Step 7	**show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**] **Example:** `Router# show ip cef`	(Optional) Displays entries in the forwarding information base (FIB). • Use the **show ip cef** command to check the forwarding table (prefixes, next hops, and interfaces).
Step 8	**show mpls interfaces** [[**vrf** *vrf-name*] [*interface*] [**detail**] | **all**] **Example:** `Router# show mpls interfaces`	(Optional) Displays information about one or more or all interfaces that are configured for label switching. • Use the **show mpls interfaces** command to verify that the interfaces are configured to use LDP.
Step 9	**show ip route** **Example:** `Router# show ip route`	(Optional) Displays IP routing table entries. • Use the **show ip route** command to display the entire routing table, including host IP address, next hop, interface, and so forth.
Step 10	**disable** **Example:** `Router# disable`	(Optional) Returns to privileged EXEC mode.

Troubleshooting Tips

You can use the **ping** and **trace** commands to verify complete MPLS connectivity in the core. You also get useful troubleshooting information from the additional **show** commands.

Configuring VRFs for CSC-PE Routers

Perform this task to configure VPN forwarding/routing instances (VRFs) for the backbone carrier edge (CSC-PE) routers.

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**
8. **interface** *type number*
9. **ip vrf forwarding** *vrf-name*
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	**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.

	Command or Action	Purpose

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, 192.168.122.15:1

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.
Step 8	**interface** *type number* **Example:** `Router(config)# interface Ethernet5/0`	Specifies the interface to configure. • The *type* argument specifies the type of interface to be configured. • The *number* argument specifies the port, connector, or interface card number.
Step 9	**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.

	Command or Action	Purpose
Step 10	**end** **Example:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

Enter a **show ip vrf detail** command and make sure the MPLS VPN is up and associated with the right interfaces.

Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier

Perform this task to configure Multiprotocol BGP (MP-BGP) connectivity in the backbone carrier.

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*} **update-source** *interface-type*
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.

	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	**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-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 10.5.5.5 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*} **update-source** *interface-type* **Example:** `Router(config-router)# neighbor 10.2.0.0 update-source loopback0`	Allows BGP sessions to use a specific operational interface for TCP connections. • 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 *interface-type* argument specifies the interface to be used as the source.
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.

	Command or Action	Purpose
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.4.0.0 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

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.

Configuring the CSC-PE and CSC-CE Routers

Perform the following tasks to configure and verify links between a CSC-PE router and the carrier CSC-CE router for an MPLS VPN CSC network that uses BGP to distribute routes and MPLS labels.

The figure below shows the configuration for the peering with directly connected interfaces between CSC-PE and CSC-CE routers. This configuration is used as the example in the tasks that follow.

Figure 25: Configuration for Peering with Directly Connected Interfaces Between CSC-PE and CSC-CE Routers

Configuring CSC-PE Routers

Perform this task to configure the CSC-PE routers.

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. **neighbor** *ip-address* **as-override**
8. **neighbor** *ip-address* **send-label**
9. **exit-address-family**
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	**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* | *peer-group-name*} **remote-as** *as-number*	Adds an entry to the BGP or multiprotocol BGP neighbor table.

	Command or Action	Purpose
	Example: `Router(config-router-af)# neighbor 10.0.0.1 remote-as 200`	• 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 10.0.0.2 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* **as-override** **Example:** `Router(config-router-af)# neighbor 10.0.0.2 as-override`	Configures a PE router to override the autonomous system number (ASN) of a site with the ASN of a provider. • The *ip-address* argument specifies the IP address of the router that is to be overridden with the ASN provided.
Step 8	**neighbor** *ip-address* **send-label** **Example:** `Router(config-router-af)# neighbor 10.0.0.2 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 9	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 10	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

Enter a **show ip bgp neighbor** command to verify that the neighbors are up and running. Make sure you see the following line in the command output under Neighbor capabilities:

```
IPv4 MPLS Label capability:advertised and received
```

Configuring CSC-CE Routers

Perform this task to configure the CSC-CE routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **redistribute** *protocol*
6. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
7. **neighbor** {*ip-address* | *peer-group-name*} **activate**
8. **neighbor** *ip-address* **send-label**
9. **exit-address-family**
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 200`	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`	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.

	Command or Action	Purpose
		• 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	**redistribute** *protocol*
	
	Example:
	
	`Router(config-router-af)# redistribute static`
		Redistributes routes from one routing domain into another routing domain.
		• The *protocol* argument specifies the source protocol from which routes are being redistributed. It can be one of the following keywords: **bgp**, **egp**, **igrp**, **isis**, **ospf**, **mobile**, **static** [**ip**], **connected**, and **rip**.
		  • The **static** [**ip**] keyword redistributes IP static routes. The optional **ip** keyword is used when you redistribute static routes into IS-IS.
		  • The **connected** keyword refers to routes which are established automatically when IP is enabled on an interface. For routing protocols such as OSPF and IS-IS, these routes are redistributed as external to the autonomous system.

Step 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor 10.5.0.2 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 7	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.3.0.2 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 8	**neighbor** *ip-address* **send-label** **Example:** `Router(config-router-af)# neighbor 10.0.0.2 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 9	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits from the address family configuration mode.

	Command or Action	Purpose
Step 10	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Verifying Labels in the CSC-PE Routers

Perform this task to verify the labels in the CSC-PE routers.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**]
3. **show mpls interfaces** [**all**]
4. **show ip route vrf** *vrf-name* [*prefix*]
5. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**]
6. **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]
7. **show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
8. **traceroute vrf** [*vrf-name*] *ip-address*
9. **disable**

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 bgp vpnv4** {**all**	**rd** *route-distinguisher*	**vrf** *vrf-name*} [**summary**] [**labels**] **Example:** `Router# show ip bgp vpnv4 all summary`	(Optional) Displays VPN address information from the BGP table. • Use the **show ip bgp vpnv4 all summary** command to check that the BGP session is up and running between the CSC-PE routers and the CSC-CE routers. Check the data in the State/PfxRcd column to verify that prefixes are learned during each session.

	Command or Action	Purpose
Step 3	**show mpls interfaces** [**all**] **Example:** `Router# show mpls interfaces all`	(Optional) Displays information about one or more interfaces that have been configured for label switching. • Use the **show mpls interfaces all** command to check that MPLS interfaces are up and running, and that LDP-enabled interfaces show that LDP is up and running. Check that LDP is turned off on the VRF because EBGP distributes the labels.
Step 4	**show ip route vrf** *vrf-name* [*prefix*] **Example:** `Router# show ip route vrf vpn1 10.5.5.5`	(Optional) Displays the IP routing table associated with a VRF. • Use the **show ip route vrf** command to check that the prefixes for the PE routers are in the routing table of the CSC-PE routers. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes for the same destination learned from the CSC-CE are installed in the corresponding VRF routing table.
Step 5	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**] **Example:** `Router# show ip bgp vpnv4 vrf vpn1 labels`	(Optional) Displays VPN address information from the BGP table. • Use the **show ip bgp vpnv4 vrf** *vrf-name* **labels** command to check that the prefixes for the customer carrier MPLS service provider networks are in the BGP table and have the appropriate labels. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the labels for the same destination learned from the CSC-CE are installed in the corresponding VRF routing table.
Step 6	**show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**] **Example:** `Router# show ip cef vrf vpn1 10.1.0.0 detail`	(Optional) Displays entries in the forwarding information base (FIB) or displays a summary of the FIB. • Use the **show ip cef vrf** and the **show ip cef vrf detail** commands to check that the prefixes of the PE routers are in the CEF table.
Step 7	**show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:** `Router# show mpls forwarding-table vrf vpn1 10.1.0.0 detail`	(Optional) Displays the contents of the MPLS lable forwarding information base (LFIB). • Use the **show mpls forwarding-table** command with the **vrf** keyword and both the **vrf** and **detail** keywords to check that the prefixes for the PE routers in the local customer MPLS VPN service provider are in the LFIB. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the labels for the same destination learned from the CSC-CE are installed in the corresponding VRF table.
Step 8	**traceroute vrf** [*vrf-name*] *ip-address* **Example:** `Router# traceroute vrf vpn2 10.2.0.0`	Shows the routes that packets follow traveling through a network to their destination. • Use the **traceroute vrf** command to check the data path and transport labels from a PE to a destination CE router.

	Command or Action	Purpose
		Note This command works with MPLS-aware traceroute only if the backbone routers are configured to propagate and generate IP Time to Live (TTL) information. For more information, see the documentation on the **mpls ip propagate-ttl** command. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes for the same destination learned from the CSC-CE are installed in the corresponding VRF table.
Step 9	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Verifying Labels in the CSC-CE Routers

Perform this task to verify the labels in the CSC-CE routers.

SUMMARY STEPS

1. **enable**
2. **show ip bgp summary**
3. **show ip route** [*address*]
4. **show mpls ldp bindings** [*network* {*mask* | *length*}]
5. **show ip cef** [*network* [*mask*]] [**longer-prefixes**] [**detail**]
6. **show mpls forwarding table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [- *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
7. **show ip bgp labels**

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 bgp summary** **Example:** `Router# show ip bgp summary`	(Optional) Displays the status of all BGP connections. • Use the **show ip bgp summary** command to check that the BGP session is up and running on the CSC-CE routers.

	Command or Action	Purpose
Step 3	**show ip route** [*address*] **Example:** `Router# show ip route 10.1.0.0`	(Optional) Displays IP routing table entries. • Use the **show ip route** to check that the loopback address of the local and remote PE routers are in the routing table. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes for the same destination learned from the CSC-CE are installed in the corresponding VRF table.
Step 4	**show mpls ldp bindings** [*network* {*mask* | *length*}] **Example:** `Router# show mpls ldp bindings 10.2.0.0 255.255.255.255`	(Optional) Displays the contents of the label information base (LIB). • Use the **show mpls ldp bindings** command to check that the prefix of the local PE router is in the MPLS LDP bindings.
Step 5	**show ip cef** [*network* [*mask*]] [**longer-prefixes**] [**detail**] **Example:** `Router# show ip cef 10.5.0.0 detail`	(Optional) Displays entries in the forwarding information base (FIB) or a summary of the FIB. • Use the **show ip cef** and the **show ip cef detail** commands to check that the prefixes of the local and remote PE routers are in the Cisco Express Forwarding table. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes and the labels for the same destination learned from the CSC-CE are installed in the corresponding VRF table.
Step 6	**show mpls forwarding table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [- *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:** `Router# show mpls forwarding-table 10.2.0.0 detail`	(Optional) Displays the contents of the MPLS LFIB. • Use the **show mpls forwarding-table** and **show mpls forwarding-table detail** commands to check that the prefixes of the local and remote PE routers are in the MPLS forwarding table. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes and labels for the same destination learned from the CSC-CE are installed in the corresponding VRF routing table.
Step 7	**show ip bgp labels** **Example:** `Router# show ip bgp labels`	(Optional) Displays information about MPLS labels from the EBGP route table. • Use the **show ip bgp labels** command to check that the BGP routing table contains labels for prefixes in the customer carrier MPLS VPN service provider networks.

Configuring the Customer Carrier Network

Perform the following tasks to configure and verify the customer carrier network. This requires setting up connectivity and routing functions for the customer carrier core (P) routers and the customer carrier edge (PE) routers.

Prerequisites

Before you configure an MPLS VPN CSC network that uses BGP to distribute routes and MPLS labels, you must configure the following on your customer carrier routers:

- An IGP routing protocol--BGP, OSPF, IS-IS, EIGRP, static, and so on. For information, see Configuring a Basic BGP Network, Configuring OSPF, Configuring a Basic IS-IS Network, and Configuring EIGRP.
- MPLS VPN functionality on the PE routers (for hierarchical VPNs only).
- Label Distribution Protocol (LDP) on P and PE routers (for hierarchical VPNs only). For information, see How to Configure MPLS LDP.

Note You must configure the items in the preceding list before performing the tasks in this section.

Verifying IP Connectivity in the Customer Carrier

Perform this task to verify IP connectivity in the customer carrier.

SUMMARY STEPS

1. **enable**
2. **ping** [*protocol*] {*host-name* | *system-address*}
3. **trace** [*protocol*] [*destination*]
4. **show ip route**
5. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping** [*protocol*] {*host-name* | *system-address*}	Diagnoses basic network connectivity on AppleTalk, CLNS, IP, Novell, Apollo, VINES, DECnet, or XNS networks.

	Command or Action	Purpose
	Example: `Router# ping ip 10.2.0.0`	• Use the **ping** command to verify the connectivity from one customer carrier core router to another.
Step 3	**trace** [*protocol*] [*destination*] **Example:** `Router# trace ip 10.1.0.0`	Discovers the routes that packets will actually 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** **Example:** `Router# show ip route`	Displays IP routing table entries. • Use the **show ip route** command to display the entire routing table, including host IP address, next hop, interface, and so forth.
Step 5	**disable** **Example:** `Router# disable`	Returns to user mode.

Configuring a Customer Carrier Core Router as a Route Reflector

Perform this task to configure a customer carrier core (P) router as a route reflector of multiprotocol BGP prefixes.

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* **route-reflector-client**
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 200`	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 labels 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 10.1.1.1 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 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 10.1.1.1 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	**neighbor** *ip-address* **route-reflector-client** **Example:** `Router(config-router-af)# neighbor 10.1.1.1 route-reflector-client`	Configures the router as a BGP route reflector and configures the specified neighbor as its client. • The *ip-address* argument specifies the IP address of the BGP neighbor being identified as a client.
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.

Troubleshooting Tips

By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only unicast address prefixes. For neighbors to exchange other address prefix types, such as multicast and VPNv4, you must also activate neighbors using the **neighbor activate** command in address family configuration mode, as shown.

Route reflectors and clients (neighbors or internal BGP peer groups) that are defined in router configuration mode using the **neighbor route-reflector-client** command reflect unicast address prefixes to and from those clients by default. To cause them to reflect prefixes for other address families, such as multicast, define the reflectors and clients in address family configuration mode, using the **neighbor route-reflector-client** command, as shown.

Configuring the Customer Site for Hierarchical VPNs

Note This section applies only to customer carrier networks that use BGP to distribute routes and MPLS labels.

Perform the following tasks to configure and verify the customer site for hierarchical VPNs:

Note This section applies to hierarchical VPNs only.

Defining VPNs on PE Routers for Hierarchical VPNs

Perform this task to define VPNs on PE routers.

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. **ip vrf forwarding** *vrf-name*
8. **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 vpn2`	Creates a VRF routing table and a Cisco Express Forwarding table and enters VRF configuration mode. • The *vrf-name* argument is a name you assign to a VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 200:1`	Creates routing and forwarding tables for a VRF. • The *route-distinguisher* argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix.
Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf)# route-target export 200:1`	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.

	Command or Action	Purpose
		• The **both** keyword imports routing information from 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 6	**import map** *route-map* **Example:** `Router(config-vrf)# import map map23`	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	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-vrf)# ip vrf forwarding vpn2`	Associates a VPN VRF instance with an interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.
Step 8	**exit** **Example:** `Router(config-vrf)# exit`	Exits to global configuration mode.

Configuring BGP Routing Sessions on the PE Routers for Hierarchical VPNs

Perform this task to configure BGP routing sessions on the PE routers for PE-to-CE router communication.

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. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 200`	Configures the router to run a BGP 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 multicast`	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* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor 10.5.5.5 remote-as 300`	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 10.1.0.0 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.

	Command or Action	Purpose
Step 7	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Verifying Labels in Each PE Router for Hierarchical VPNs

Perform this task to verify labels in each PE router for hierarchical VPNs.

SUMMARY STEPS

1. **enable**
2. **show ip route vrf** *vrf-name* [*prefix*]
3. **show mpls forwarding-table** [**vrf** *vrf-name*] [*prefix*] [**detail**]
4. **show ip cef** [*network* [*mask* [**longer-prefix**]]] [**detail**]
5. **show ip cef vrf** *vrf-name* [*ip-prefix*]
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	**show ip route vrf** *vrf-name* [*prefix*] **Example:** `Router# show ip route vrf vpn2 10.5.5.5`	(Optional) Displays the IP routing table associated with a VRF. • Use the **show ip route vrf** command to check that the loopback addresses of the local and remote CE routers are in the routing table of the PE routers.
Step 3	**show mpls forwarding-table** [**vrf** *vrf-name*] [*prefix*] [**detail**] **Example:** `Router# show mpls forwarding-table vrf vpn2 10.1.0.0`	(Optional) Displays the contents of the LFIB. • Use the **show mpls forwarding-table** command to check that the prefixes for the local and remote CE routers are in the MPLS forwarding table, and that the specified prefix is untagged.
Step 4	**show ip cef** [*network* [*mask* [**longer-prefix**]]] [**detail**]	(Optional) Displays specific entries in the FIB based on IP address information.

	Command or Action	Purpose
	Example: `Router# show ip cef 10.2.0.0`	• Use the **show ip cef** command to check that the prefixes of the local and remote PE routers are in the Cisco Express Forwarding table.
Step 5	**show ip cef vrf** *vrf-name* [*ip-prefix*] **Example:** `Router# show ip cef vrf vpn2 10.3.0.0`	(Optional) Displays the Cisco Express Forwarding table associated with a VRF. • Use the **show ip cef vrf** command to check that the prefix of the remote CE router is in the Cisco Express Forwarding table.
Step 6	**exit** **Example:** `Router# exit`	(Optional) Exits to user EXEC mode.

Configuring CE Routers for Hierarchical VPNs

Perform this task to configure CE routers for hierarchical VPNs. This configuration is the same as that for an MPLS VPN that is not in a hierarchical topology.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef** [**distributed**]
4. **interface** *type number*
5. **ip addres** *ip-address mask* [**secondary**]
6. **exit**
7. **router bgp** *as-number*
8. **redistribute** *protocol*
9. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip cef** [**distributed**] **Example:** `Router(config)# ip cef distributed`	Enables Cisco Express Forwarding on the route processor card. • The **distributed** keyword enables distributed Cisco Express Forwarding operation. Cisco Express Forwarding information is distributed to the line cards. Line cards perform express forwarding. **Note** For the Cisco ASR 1000 Series Aggregation Services Router, the **distributed** keyword is required.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 0`	Configures an interface type and enters interface configuration mode. • The *type* argument specifies the type of interface to be configured. • A loopback interface indicates a software-only interface that emulates an interface that is always up. It is a virtual interface supported on all platforms. • The *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 you can create.
Step 5	**ip addres** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.8.0.0 255.255.255.255`	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 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 7	**router bgp** *as-number*	Configures a BGP routing process and enters router configuration mode.

	Command or Action	Purpose
	Example: `Router(config)# router bgp 100`	• 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 8	**redistribute** *protocol* **Example:** `Router(config-router)# redistribute connected`	Redistributes routes from one routing domain into another routing domain. • The *protocol* argument specifies the source protocol from which routes are being redistributed. It can be one of the following keywords: **bgp**, **connected**, **egp**, **igrp**, **isis**, **mobile**, **ospf**, **static** [**ip**], or **rip**. The **connected** keyword refers to routes that are established automatically when IP is enabled on an interface. For routing protocols such as Open Shortest Path First (OSPF) and IS-IS, these routes are redistributed as external to the autonomous system.
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.8.0.0 remote-as 100`	Adds the IP address of the neighbor in the remote autonomous system to the multiprotocol BGP neighbor table of the local 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. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 10	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Verifying IP Connectivity in the Customer Site

Perform this task to verify IP connectivity in the customer site.

SUMMARY STEPS

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

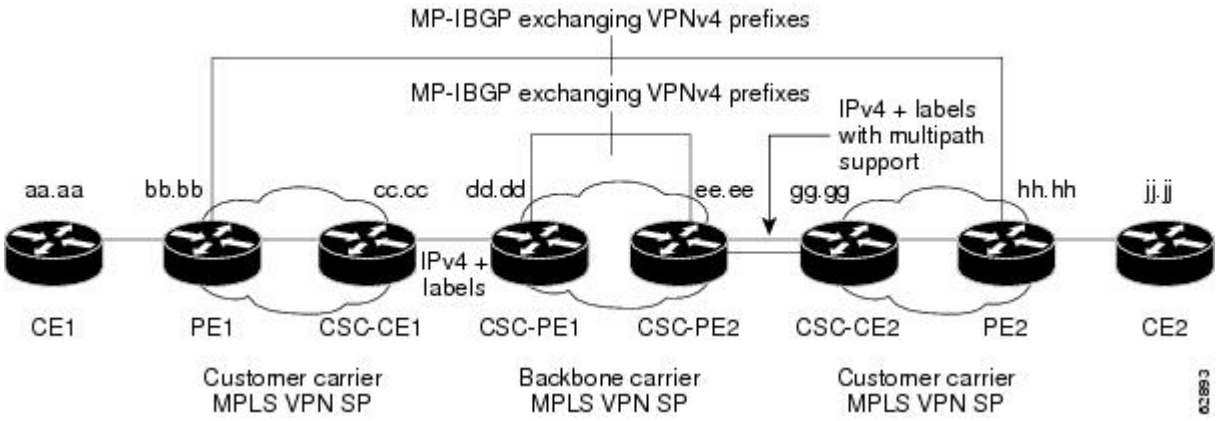
DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip route** [*ip-address* [*mask*]] [**longer-prefixes**] | *protocol* [*process-id*] | **list** [*access-list-number* | *access-list-name*] | **static download** **Example:** `Router# show ip route 10.5.5.5`	(Optional) Displays the current state of the routing table. • Use the **show ip route ip-address** command to check that the loopback addresses of the remote CE routers learned through the PE router are in the routing table of the local CE routers.
Step 3	**ping** [*protocol*] {*host-name* | *system-address*} **Example:** `Router# ping 10.5.5.5`	Diagnoses basic network connectivity on Apollo, AppleTalk, Connectionless Network Service (CLNS), DECnet, IP, Novell IPX, VINES, or XNS networks. • Use the **ping** command to check connectivity between customer site routers.
Step 4	**trace** [*protocol*] [*destination*] **Example:** `Router# trace ip 10.5.5.5`	Discovers the routes that packets will actually take when traveling to their destination. • Use the **trace** command to follow the path of the packets in the customer site. • To use nondefault parameters and invoke an extended **trace** test, enter the **trace** command without a destination argument. You will be stepped through a dialog to select the desired parameters.
Step 5	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Configuration Examples for MPLS VPN CSC with BGP

The figure below shows a sample CSC topology for exchanging IPv4 routes and MPLS labels. Use this figure as a reference for configuring and verifying carrier supporting carrier routers to exchange IPv4 routes and MPLS labels.

Figure 26: Sample CSC Topology for Exchanging IPv4 Routes and MPLS Labels



The table below describes the sample configuration shown in the figure above.

Table 10: Description of Sample Configuration Shown in figure 1

Routers	Description
CE1 and CE2	Belong to an end customer. CE1 and CE2 routers exchange routes learned from PE routers. The end customer is purchasing VPN services from a customer carrier.
PE1 and PE2	Part of a customer carrier network that is configured to provide MPLS VPN services. PE1 and PE2 are peering with a VPNv4 IBGP session to form an MPLS VPN network.
CSC-CE1 and CSC-CE2	Part of a customer carrier network. CSC-CE1 and CSC-CE2 routers exchange IPv4 BGP updates with MPLS labels and redistribute PE loopback addressees to and from the IGP (OSPF in this example). The customer carrier is purchasing carrier supporting carrier VPN services from a backbone carrier.

Routers	Description
CSC-PE1 and CSC-PE2	Part of the backbone carrier's network configured to provide carrier supporting carrier VPN services. CSC-PE1 and CSC-PE2 are peering with a VPNv4 IP BGP session to form the MPLS VPN network. In the VRF, CSC-PE1 and CSC-PE2 are peering with the CSC-CE routers, which are configured for carrying MPLS labels with the routes, with an IPv4 EBGP session.

Configuring the Backbone Carrier Core Examples

Configuration and verification examples for the backbone carrier core included in this section are as follows:

Verifying IP Connectivity and LDP Configuration in the CSC Core Example

Check that CSC-PE2 is reachable from CSC-PE1 by entering the following command on CSC-CE1:

```
Router# ping 10.5.5.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

Verify the path from CSC-PE1 to CSC-PE2 by entering the following command on CSC-CE1:

```
Router# trace 10.5.5.5
Type escape sequence to abort.
Tracing the route to 10.5.5.5
  1 10.5.5.5 0 msec 0 msec *
```

Check that CSC-PE router prefixes are in the MPLS forwarding table:

```
Router# show mpls forwarding-table
Local     Outgoing     Prefix or          Bytes tag    Outgoing      Next Hop
tag       tag or VC    Tunnel Id          switched     interface
16        2/nn         dd.dd.dd.dd/32       0          AT2/1/0.1     point2point
17        16           bb.bb.bb.bb/32[V]    30204      Et1/0         pp.0.0.1
21        Pop tag      cc.cc.cc.cc/32[V]    0          Et1/0         pp.0.0.1
22        Pop tag      nn.0.0.0/8[V]        570        Et1/0         pp.0.0.1
23        Aggregate    pp.0.0.0/8[V]        0
2         2/nn         gg.gg.gg.gg/32[V]    0          AT3/0.1       point2point
8         2/nn         hh.hh.hh.hh/32[V]    15452      AT3/0.1       point2point
29        2/nn         qq.0.0.0/8[V]        0          AT3/0.1       point2point
30        2/nn         ss.0.0.0/8[V]        0          AT3/0.1       point2point
```

Check the status of LDP discovery processes in the core:

```
Router# show mpls ldp discovery
Local LDP Identifier:
  ee.ee.ee.ee:0
  Discovery Sources:
  Interfaces:
    ATM2/1/0.1 (ldp): xmit/recv
      TDP Id: dd.dd.dd.dd:1
```

Check the status of LDP sessions in the core:

```
Router# show mpls ldp neighbor
```

```
Peer LDP Ident: dd.dd.dd.dd:1; Local LDP Ident ee.ee.ee.ee:1
       TCP connection: dd.dd.dd.dd.646 - ee.ee.ee.ee.11007
       State: Oper; Msgs sent/rcvd: 20/21; Downstream on demand
       Up time: 00:14:56
       LDP discovery sources:
         ATM2/1/0.1, Src IP addr: dd.dd.dd.dd
```

Check the forwarding table (prefixes, next-hops, and interfaces):

```
Router# show ip cef
Prefix              Next Hop          Interface
0.0.0.0/0           drop              Null0 (default route handler entry)
0.0.0.0/32          receive
dd.dd.dd.dd/32      dd.dd.dd.dd       ATM2/1/0.1
ee.ee.ee.ee/32      receive
224.0.0.0/4         drop
224.0.0.0/24        receive
255.255.255.255/32 receive
```

Note Also see the Verifying Labels in the CSC-CE Routers Examples, on page 237.

Verify that interfaces are configured to use LDP:

```
Router# show mpls interfaces
Interface              IP            Tunnel   Operational
Ethernet0/1            Yes (ldp)     No       Yes
```

Display the entire routing table, including host IP address, next hop, interface, and so forth:

```
Router# show ip route
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
     dd.0.0.0/32 is subnetted, 1 subnets
O       dd.dd.dd.dd [110/7] via dd.dd.dd.dd, 00:16:42, ATM2/1/0.1
     ee.0.0.0/32 is subnetted, 1 subnets
C       ee.ee.ee.ee is directly connected, Loopback0
```

Configuring VRFs for CSC-PE Routers Example

The following example shows how to configure a VPN routing and forwarding (VRF) instance for a CSC-PE router:

```
ip cef distributed
ip vrf vpn1
rd 100:1
route target both 100:1
!
```

Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier Example

The following example shows how to configure Multiprotocol BGP (MP-BGP) for VPN connectivity in the backbone carrier:

```
ip cef distributed
ip vrf vpn1
rd 100:1
```

```
route target both 100:1
hostname csc-pe1
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor ee.ee.ee.ee remote-as 100
 neighbor ee.ee.ee.ee update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee send-community extended
 bgp dampening 30
 exit-address-family
 !
router bgp 100
. . .
! (BGP IPv4 to CSC-CE router from CSC-PE router)
!
 address-family ipv4 vrf vpn1
 neighbor ss.0.0.2 remote-as 200
 neighbor ss.0.0.2 activate
 neighbor ss.0.0.2 as-override
 neighbor ss.0.0.2 advertisement-interval 5
 neighbor ss.0.0.2 send-label
 no auto-summary
 no synchronization
 bgp dampening 30
 exit-address-family
!
```

Configuring the Links Between CSC-PE and CSC-CE Routers Examples

This section contains the following examples:

Configuring the CSC-PE Routers Examples

The following example shows how to configure a CSC-PE router:

```
ip cef
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
mpls label protocol ldp
!
interface Loopback0
 ip address dd.dd.dd.dd 255.255.255.255
!
interface Ethernet3/1
 ip vrf forwarding vpn1
 ip address pp.0.0.2 255.0.0.0
!
interface ATM0/1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM0/1/0.1 mpls
 ip unnumbered Loopback0
 no ip directed-broadcast
```

```
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
router ospf 100
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Ethernet3/1
 network dd.dd.dd.dd 0.0.0.0 area 100
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor ee.ee.ee.ee remote-as 100
 neighbor ee.ee.ee.ee update-source Loopback0
!
address-family vpnv4                                !VPNv4 session with CSC-PE2
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee send-community extended
 bgp dampening 30
 exit-address-family
!
 address-family ipv4 vrf vpn1
 neighbor pp.0.0.1 remote-as 200
 neighbor pp.0.0.1 activate
 neighbor pp.0.0.1 as-override
 neighbor pp.0.0.1 advertisement-interval 5
 neighbor pp.0.0.1 send-label
 no auto-summary
 no synchronization
 bgp dampening 30
 exit-address-family
```

Configuring the CSC-CE Routers Examples

The following example shows how to configure a CSC-CE router:

```
ip cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address cc.cc.cc.cc 255.255.255.255
!
interface Ethernet3/0
 ip address pp.0.0.1 255.0.0.0
!
interface Ethernet4/0
 ip address nn.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 mpls label protocol ldp
 mpls ip
!
router ospf 200
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets                     !Exchange routes
 redistribute bgp 200 metric 3 subnets              !learned from PE1
 passive-interface ATM1/0
 passive-interface Ethernet3/0
 network cc.cc.cc.cc 0.0.0.0 area 200
 network nn.0.0.0 0.255.255.255 area 200
!
router bgp 200
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
```

```
 timers bgp 10 30
 neighbor pp.0.0.2 remote-as 100
 neighbor pp.0.0.2 update-source Ethernet3/0
 no auto-summary
!
address-family ipv4
 redistribute connected
 redistribute ospf 200 metric 4 match internal
 neighbor pp.0.0.2 activate
 neighbor pp.0.0.2 send-label
 no auto-summary
 no synchronization
 bgp dampening 30
 exit-address-family
```

Verifying Labels in the CSC-PE Routers Examples

The following examples show how to verify the configurations of the CSC-PE routers.

Verify that the BGP session is up and running between the CSC-PE router and the CSC-CE router. Check the data in the State/PfxRcd column to verify that prefixes are learned during each session.

```
Router# show ip bgp vpnv4 all summary
BBGP router identifier 10.5.5.5, local AS number 100
BGP table version is 52, main routing table version 52
12 network entries and 13 paths using 2232 bytes of memory
6 BGP path attribute entries using 336 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
1 BGP extended community entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Dampening enabled. 0 history paths, 0 dampened paths
BGP activity 16/4 prefixes, 27/14 paths, scan interval 5 secs
Neighbor        V   AS     MsgRcvd MsgSent  TblVer InQ  OutQ Up/Down  State/PfxRcd
10.5.5.5        4  100     7685    7686      52    0       0 21:17:04       6
10.0.0.2        4  200     7676    7678      52    0       0 21:16:43       7
```

Verify that the MPLS interfaces are up and running, and that LDP-enabled interfaces show that LDP is up and running. LDP is turned off on the VRF because EBGP distributes the labels.

```
Router# show mpls interfaces all
Interface              IP            Tunnel   Operational
GigabitEthernet6/0     Yes (ldp)     No       Yes
VRF vpn1:
Ethernet3/1            No            No       Yes
```

Verify that the prefix for the local PE router is in the routing table of the CSC-PE router:

```
Router# show ip route vrf vpn2 10.5.5.5
Routing entry for 10.5.5.5/32
  Known via "bgp 100", distance 20, metric 4
  Tag 200, type external
  Last update from pp.0.0.2 21:28:39 ago
  Routing Descriptor Blocks:
  * pp.0.0.2, from pp.0.0.2, 21:28:39 ago
      Route metric is 4, traffic share count is 1
      AS Hops 1, BGP network version 0
```

Verify that the prefix for the remote PE router is in the routing table of the CSC-PE router:

```
Router# show ip route vrf vpn2 10.5.5.5
Routing entry for 10.5.5.5/32
  Known via "bgp 100", distance 200, metric 4
  Tag 200, type internal
  Last update from 10.1.0.0 21:27:39 ago
  Routing Descriptor Blocks:
  * 10.1.0.0 (Default-IP-Routing-Table), from 10.1.0.0, 21:27:39 ago
      Route metric is 4, traffic share count is 1
      AS Hops 1, BGP network version 0
```

Verify that the prefixes for the customer carrier MPLS VPN service provider networks are in the BGP table, and have appropriate labels:

```
Router# show ip bgp vpnv4 vrf vpn2 labels

   Network          Next Hop      In label/Out label
Route Distinguisher: 100:1 (vpn1)
   cc.cc.cc.cc/32   pp.0.0.2        22/imp-null
   bb.bb.bb.bb/32   pp.0.0.2        27/20
   hh.hh.hh.hh/32   ee.ee.ee.ee     34/35
   gg.gg.gg.gg/32   ee.ee.ee.ee     30/30
   nn.0.0.0         pp.0.0.2        23/imp-null
   ss.0.0.0         ee.ee.ee.ee     33/34
   pp.0.0.0         pp.0.0.2        25/aggregate(vpn1)
```

Verify that the prefix of the PE router in the local customer carrier MPLS VPN service provider is in the Cisco Express Forwarding table:

```
Router# show ip cef vrf vpn2 10.1.0.0
10.1.0.0/32, version 19, cached adjacency pp.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: 27
    fast tag rewrite with Et3/1, pp.0.0.2, tags imposed {20}
  via pp.0.0.2, 0 dependencies, recursive
    next hop pp.0.0.2, Ethernet3/1 via pp.0.0.2/32
    valid cached adjacency
    tag rewrite with Et3/1, pp.0.0.2, tags imposed {20}

Router# show ip cef vrf vpn2 10.1.0.0 detail
10.1.0.0/32, version 19, cached adjacency pp.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: 27
    fast tag rewrite with Et3/1, pp.0.0.2, tags imposed {20}
  via pp.0.0.2, 0 dependencies, recursive
    next hop pp.0.0.2, Ethernet3/1 via pp.0.0.2/32
    valid cached adjacency
    tag rewrite with Et3/1, pp.0.0.2, tags imposed {20}
```

Verify that the prefix of the PE router in the local customer carrier MPLS VPN service provider is in the MPLS forwarding table:

```
Router# show mpls forwarding-table vrf vpn2 10.1.0.0
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
27     20          10.1.0.0/32[V]   958048     Et3/1       pp.0.0.2

Router# show mpls forwarding-table vrf vpn2 10.1.0.0 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
27     20 10.1.0.0/32[V]            958125     Et3/1      pp.0.0.2
        MAC/Encaps=14/18, MTU=1500, Tag Stack{20}
        00B04A74A05400B0C26E10558847 00014000
        VPN route: vpn1
        No output feature configured
    Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Verify that the prefix of the PE router in the remote customer carrier MPLS VPN service provider is in the Cisco Express Forwarding table:

```
Router# show ip cef vrf vpn2 10.3.0.0
10.3.0.0/32, version 25, cached adjacency rr.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: 34
    fast tag rewrite with Gi6/0, rr.0.0.2, tags imposed {35}
  via ee.ee.ee.ee, 0 dependencies, recursive
    next hop rr.0.0.2, GigabitEthernet6/0 via ee.ee.ee.ee/32
    valid cached adjacency
```

```
    tag rewrite with Gi6/0, rr.0.0.2, tags imposed {35}

Router# show ip cef vrf vpn2 10.3.0.0 detail
hh.hh.hh.hh/32, version 25, cached adjacency rr.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: 34
    fast tag rewrite with Gi6/0, rr.0.0.2, tags imposed {35}
  via ee.ee.ee.ee, 0 dependencies, recursive
    next hop rr.0.0.2, GigabitEthernet6/0 via ee.ee.ee.ee/32
    valid cached adjacency
    tag rewrite with Gi6/0, rr.0.0.2, tags imposed {35}
```

Verify that the prefix of the PE router in the remote customer carrier MPLS VPN service provider is in the MPLS forwarding table:

```
Router# show mpls forwarding-table vrf vpn2 10.3.0.0
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
34     35          hh.hh.hh.hh/32[V] 139034     Gi6/0      rr.0.0.2

Router# show mpls forwarding-table vrf vpn2 10.3.0.0 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
34     35          hh.hh.hh.hh/32[V] 139034     Gi6/0      rr.0.0.2
        MAC/Encaps=14/18, MTU=1500, Tag Stack{35}
        00B0C26E447000B0C26E10A88847 00023000
        VPN route: vpn1
        No output feature configured
    Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Verifying Labels in the CSC-CE Routers Examples

The following examples show how to verify the configurations of the CSC-CE routers.

Verify that the BGP session is up and running:

```
Router# show ip bgp summary
BGP router identifier cc.cc.cc.cc, local AS number 200
BGP table version is 35, main routing table version 35
14 network entries and 14 paths using 2030 bytes of memory
3 BGP path attribute entries using 168 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Dampening enabled. 1 history paths, 0 dampened paths
BGP activity 17/67 prefixes, 29/15 paths, scan interval 60 secs
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
pp.0.0.1        4   100    7615    7613       35    0    0 21:06:19        5
```

Verify that the loopback address of the local PE router is in the routing table:

```
Router# show ip route 10.1.0.0
Routing entry for 10.1.0.0/32
  Known via "ospf 200", distance 110, metric 101, type intra area
  Redistributing via bgp 200
  Advertised by bgp 200 metric 4 match internal
  Last update from nn.0.0.1 on Ethernet4/0, 00:34:08 ago
  Routing Descriptor Blocks:
  * nn.0.0.1, from bb.bb.bb.bb, 00:34:08 ago, via Ethernet4/0
      Route metric is 101, traffic share count is 1
```

Verify that the loopback address of the remote PE router is in the routing table:

```
Router# show ip route 10.5.5.5
Routing entry for 10.5.5.5/32
  Known via "bgp 200", distance 20, metric 0
  Tag 100, type external
  Redistributing via ospf 200
```

```
  Advertised by ospf 200 metric 3 subnets
  Last update from pp.0.0.1 00:45:16 ago
  Routing Descriptor Blocks:
  * pp.0.0.1, from pp.0.0.1, 00:45:16 ago
      Route metric is 0, traffic share count is 1
      AS Hops 2, BGP network version 0
```

Verify that the prefix of the local PE router is in the MPLS LDP bindings:

```
Router# show mpls ldp bindings 10.1.0.0 255.255.255.255
  tib entry: 10.1.0.0/32, rev 20
        local binding:  tag: 20
        remote binding: tsr: 10.1.0.0:0, tag: imp-null
```

Verify that the prefix of the local PE router is in the Cisco Express Forwarding table:

```
Router# show ip cef 10.1.0.0
10.1.0.0/32, version 46, cached adjacency nn.0.0.1
0 packets, 0 bytes
  tag information set
    local tag: 20
  via nn.0.0.1, Ethernet4/0, 0 dependencies
    next hop nn.0.0.1, Ethernet4/0
    unresolved
    valid cached adjacency
    tag rewrite with Et4/0, nn.0.0.1, tags imposed {}
```

Verify that the prefix of the local PE router is in the MPLS forwarding table:

```
Router# show mpls forwarding-table 10.1.0.0
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
20     Pop tag     bb.bb.bb.bb/32    893397     Et4/0      nn.0.0.1

Router# show mpls forwarding-table 10.1.0.0 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
20     Pop tag     bb.bb.bb.bb/32    893524     Et4/0      nn.0.0.1
        MAC/Encaps=14/14, MTU=1504, Tag Stack{}
        00074F83685400B04A74A0708847
        No output feature configured
    Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Verify that the BGP routing table contains labels for prefixes in the customer carrier MPLS VPN service provider networks:

```
Router# show ip bgp labels
Network          Next Hop         In Label/Out Label
cc.cc.cc.cc/32   0.0.0.0          imp-null/exp-null
bb.bb.bb.bb/32   nn.0.0.1         20/exp-null
hh.hh.hh.hh/32   pp.0.0.1         26/34
gg.gg.gg.gg/32   pp.0.0.1         23/30
nn.0.0.0         0.0.0.0          imp-null/exp-null
ss.0.0.0         pp.0.0.1         25/33
pp.0.0.0         0.0.0.0          imp-null/exp-null
pp.0.0.1/32      0.0.0.0          16/exp-null
```

Verify that the prefix of the remote PE router is in the Cisco Express Forwarding table:

```
Router# show ip cef 10.5.5.5
10.5.5.5/32, version 54, cached adjacency pp.0.0.1
0 packets, 0 bytes
  tag information set
    local tag: 26
    fast tag rewrite with Et3/0, pp.0.0.1, tags imposed {34}
  via pp.0.0.1, 0 dependencies, recursive
    next hop pp.0.0.1, Ethernet3/0 via pp.0.0.1/32
    valid cached adjacency
    tag rewrite with Et3/0, pp.0.0.1, tags imposed {34}
```

Verify that the prefix of the remote PE router is in the MPLS forwarding table:

```
Router# show mpls forwarding-table 10.5.5.5
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
26     34          hh.hh.hh.hh/32    81786      Et3/0      pp.0.0.1

Router# show mpls forwarding-table 10.5.5.5 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
26     34          hh.hh.hh.hh/32    81863      Et3/0      pp.0.0.1
       MAC/Encaps=14/18, MTU=1500, Tag Stack{34}
       00B0C26E105500B04A74A0548847 00022000
       No output feature configured
    Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Configuring the Customer Carrier Network Examples

Customer carrier configuration and verification examples in this section include:

Verifying IP Connectivity in the Customer Carrier Example

Verify the connectivity from one customer carrier core router to another (from CE1 to CE2) by entering the following command:

```
Router# ping 10.2.0.0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to jj.jj.jj.jj, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/12 ms
```

Verify the path that a packet goes through on its way to its final destination from CE1 to CE2:

```
Router# trace 10.2.0.0
Type escape sequence to abort.
Tracing the route to 10.2.0.0
  1 mm.0.0.2 0 msec 0 msec 4 msec
  2 nn.0.0.2 [MPLS: Labels 20/21 Exp 0] 8 msec 8 msec 12 msec
  3 pp.0.0.2 [MPLS: Labels 28/21 Exp 0] 8 msec 8 msec 12 msec
  4 ss.0.0.1 [MPLS: Labels 17/21 Exp 0] 8 msec 8 msec 12 msec
  5 ss.0.0.2 [MPLS: Labels 16/21 Exp 0] 8 msec 8 msec 12 msec
  6 tt.0.0.1 [AS 200] [MPLS: Label 21 Exp 0] 8 msec 8 msec 8 msec
  7 tt.0.0.2 [AS 200] 8 msec 4 msec *
```

Verify the path that a packet goes through on its way to its final destination from CE2 to CE1:

```
Router# trace 10.1.0.0
Type escape sequence to abort.
Tracing the route to 10.1.0.0
  1 tt.0.0.1 0 msec 0 msec 0 msec
  2 qq.0.0.2 [MPLS: Labels 18/21 Exp 0] 8 msec 12 msec 12 msec
  3 ss.0.0.1 [MPLS: Labels 28/21 Exp 0] 8 msec 8 msec 8 msec
  4 pp.0.0.2 [MPLS: Labels 17/21 Exp 0] 12 msec 8 msec 8 msec
  5 pp.0.0.1 [MPLS: Labels 16/21 Exp 0] 12 msec 12 msec 8 msec
  6 mm.0.0.2 [AS 200] [MPLS: Label 21 Exp 0] 12 msec 8 msec 12 msec
  7 mm.0.0.1 [AS 200] 4 msec 4 msec *
```

Configuring a Customer Carrier Core Router as a Route Reflector Example

The following example shows how to use an address family to configure internal BGP peer 10.1.1.1 as a route-reflector client for both unicast and multicast prefixes:

```
router bgp 200
 address-family vpnv4
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 route-reflector-client

router bgp 100
 address-family vpnv4
  neighbor xx.xx.xx.xx activate
  neighbor xx.xx.xx.xx route-reflector-client
   ! xx.xx.xx,xx is a PE router
  neighbor xx.xx.xx.xx send-community extended
 exit address-family
! You need to configure your peer BGP neighbor.
```

Configuring the Customer Site for Hierarchical VPNs Examples

This section contains the following configuration and verification examples for the customer site:

Configuring PE Routers for Hierarchical VPNs Examples

This example shows how to configure a PE router:

```
ip cef
!
ip vrf vpn2
 rd 200:1
 route-target export 200:1
 route-target import 200:1
mpls label protocol ldp
!
interface Loopback0
 ip address bb.bb.bb.bb 255.255.255.255
!
interface Ethernet3/0
 ip address nn.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 mpls label protocol ldp
 mpls ip
!
interface Ethernet3/3
 ip vrf forwarding vpn2
 ip address mm.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
router ospf 200
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Ethernet3/3
 network bb.bb.bb.bb 0.0.0.0 area 200
 network nn.0.0.0 0.255.255.255 area 200
!
router bgp 200
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
```

```
 neighbor hh.hh.hh.hh remote-as 200
 neighbor hh.hh.hh.hh update-source Loopback0
!
address-family vpnv4                                  !VPNv4 session with PE2
 neighbor hh.hh.hh.hh activate
 neighbor hh.hh.hh.hh send-community extended
 bgp dampening 30
 exit-address-family
!
 address-family ipv4 vrf vpn2
 neighbor mm.0.0.1 remote-as 300
 neighbor mm.0.0.1 activate
 neighbor mm.0.0.1 as-override
 neighbor mm.0.0.1 advertisement-interval 5
no auto-summary
 no synchronization
 bgp dampening 30
 exit-address-family
```

Verifying Labels in Each PE Router for Hierarchical VPNs Examples

The following examples show how to verify the configuration of PE router in hierarchical VPNs.

Verify that the loopback address of the local CE router is in the routing table of the PE1 router:

```
Router# show ip route vrf vpn2 10.2.2.2
Routing entry for 10.2.2.2/32
  Known via "bgp 200", distance 20, metric 0
  Tag 300, type external
  Last update from mm.0.0.2 20:36:59 ago
  Routing Descriptor Blocks:
  * mm.0.0.2, from mm.0.0.2, 20:36:59 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1, BGP network version 0
```

Verify that the prefix for the local CE router is in the MPLS forwarding table, and that the prefix is untagged:

```
Router# show mpls forwarding-table vrf vpn2 10.2.2.2
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
23     Untagged    aa.aa.aa.aa/32[V] 0          Et3/3      mm.0.0.2
```

Verify that the prefix of the remote PE router is in the Cisco Express Forwarding table:

```
Router# show ip cef 10.5.5.5

10.5.5.5/32, version 31, cached adjacency nn.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: 31
    fast tag rewrite with Et3/0, nn.0.0.2, tags imposed {26}
  via nn.0.0.2, Ethernet3/0, 2 dependencies
    next hop nn.0.0.2, Ethernet3/0
    unresolved
    valid cached adjacency
    tag rewrite with Et3/0, nn.0.0.2, tags imposed {26}
```

Verify that the loopback address of the remote CE router is in the routing table:

```
Router# show ip route vrf vpn2 10.2.0.0
Routing entry for 10.2.0.0/32
  Known via "bgp 200", distance 200, metric 0
  Tag 300, type internal
  Last update from hh.hh.hh.hh 20:38:49 ago
  Routing Descriptor Blocks:
  * hh.hh.hh.hh (Default-IP-Routing-Table), from hh.hh.hh.hh, 20:38:49 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1, BGP network version 0
```

Verify that the prefix of the remote CE router is in the MPLS forwarding table, and that an outgoing interface exists:

```
Router# show mpls forwarding-table vrf vpn2 10.2.0.0
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
None   26          jj.jj.jj.jj/32   0          Et3/0      nn.0.0.2
```

Verify that the prefix of the remote CE router is in the Cisco Express Forwarding table:

```
Router# show ip cef vrf vpn2 10.2.0.0
10.2.0.0/32, version 12, cached adjacency nn.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: VPN route head
    fast tag rewrite with Et3/0, nn.0.0.2, tags imposed {26 32}
  via hh.hh.hh.hh, 0 dependencies, recursive
    next hop nn.0.0.2, Ethernet3/0 via hh.hh.hh.hh/32
    valid cached adjacency
    tag rewrite with Et3/0, nn.0.0.2, tags imposed {26 32}
```

Verify that the prefix of the local PE router is in the Cisco Express Forwarding table:

```
Router# show ip cef 10.1.0.0
10.1.0.0/32, version 9, connected, receive
  tag information set
    local tag: implicit-null
```

Configuring CE Routers for Hierarchical VPNs Examples

The following example shows how to configure a CE router:

```
ip cef distributed
interface Loopback0
ip address 10.3.0.0 255.255.255.255
!
interface FastEthernet0/3/3
 ip address mm.0.0.1 255.0.0.0
!
router bgp 300
 no synchronization
 bgp log-neighbor-changes
 timers bgp 10 30
 redistribute connected                              !Redistributing routes into BGP
 neighbor mm.0.0.2 remote-as 200                     !to send to PE1
 neighbor mm.0.0.2 advertisement-interval 5
 no auto-summary
```

Verifying IP Connectivity in the Customer Site Examples

The following examples show how to verify IP connectivity at the customer site.

Verify that the loopback address of the remote CE router, learned from the PE router, is in the routing table of the local router:

```
Router# show ip route 10.2.0.0
Routing entry for 10.2.0.0/32
  Known via "bgp 300", distance 20, metric 0
  Tag 200, type external
  Redistributing via ospf 300
  Advertised by ospf 300 subnets
  Last update from mm.0.0.1 20:29:35 ago
  Routing Descriptor Blocks:
  * mm.0.0.1, from mm.0.0.1, 20:29:35 ago
```

```
Route metric is 0, traffic share count is 1
AS Hops 2
```

Additional References

Related Documents

Related Topic	Document Title
LDP	MPLS Label Distribution Protocol
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	Title
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 CSC with BGP

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 CSC with BGP

Feature Name	Releases	Feature Information
MPLS VPN--Carrier Supporting Carrier--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 Cisco IOS XE Release 2.2	This feature enables you to create an MPLS VPN CSC network that uses BGP to transport routes and MPLS labels. In 12.0(21)ST, this feature was introduced. In 12.0(22)S, this feature was integrated. In 12.0(23)S, this feature was integrated. In 12.2(13)T, this feature was integrated. 12.0(24)S, this feature was integrated. In 12.2(14)S, this feature was integrated. In 12.0(27)S, this feature was integrated. In 12.0(29)S, this feature was integrated. In Cisco IOS XE Release 2.2, this feature was implemented on the Cisco ASR 1000 Series Routers. This feature uses no new or modified commands.

Glossary

ASBR -- Autonomous System Boundary router. A router that connects one autonomous system to another.

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).

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.

CSC --Carrier Supporting Carrier. A hierarchical VPN model that allows small service providers, or customer carriers, to interconnect their IP or MPLS networks over an MPLS backbone. This eliminates the need for customer carriers to build and maintain their own MPLS backbone.

eBGP --external 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.

edge router--A router that is at the edge of the network. It defines the boundary of the MPLS network. It receives and transmits packets. Also referred to as edge label switch router and label edge router.

iBGP --internal 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.

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.

LDP --Label Distribution Protocol. A standard protocol between MPLS-enabled routers to negotiate the labels (addresses) used to forward packets.

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.

MP-BGP --Multiprotocol BGP.

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.

NSF --Nonstop forwarding enables routers to continuously forward IP packets following a Route Processor takeover or switchover to another Route Processor. NSF maintains and updates Layer 3 routing and forwarding information in the backup Route Processor to ensure that IP packets and routing protocol information are forwarded continuously during the switchover and route convergence process.

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. All MPLS VPN processing occurs in the PE router.

QoS --quality of service. Measure of performance for a transmission system that indicates its transmission quality and service availability.

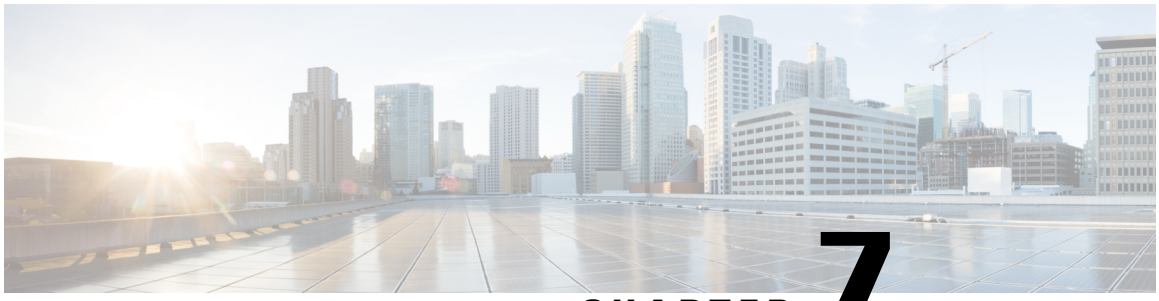
RD --route distinguisher. An 8-byte value that is concatenated with an IPv4 prefix to create a unique VPN-IPv4 prefix.

RT --route target. Extended community attribute used to identify the VRF routing table into which a prefix is imported.

SLA --Service Level Agreement given to VPN subscribers.

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 VPN site that is attached to a 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.



CHAPTER 7

MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs

The MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs feature allows MPLS VPN interautonomous (Inter-AS) and MPLS VPN Carrier Supporting Carrier (CSC) networks to load share traffic between adjacent label switch routers (LSRs) that are connected by multiple links. The LSRs can be a pair of Autonomous System Boundary Routers (ASBRs) or a CSC-provider edge (PE) and a CSC-customer edge (CE) device. Using directly connected loopback peering allows load sharing at the Interior Gateway Protocol (IGP) level so only one Border Gateway Protocol (BGP) session is needed between the LSRs. No other label distribution mechanism is needed between the adjacent LSRs except BGP.

- Finding Feature Information, page 247
- Prerequisites for MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs, page 248
- Restrictions for MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs, page 248
- Information About MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs, page 250
- How to Configure MPLS VPN Load Balancing Support for Inter-AS and CSC VPN, page 251
- Configuration Examples for MPLS VPN Load Balancing Support for Inter-AS and CSC VPN , page 281
- Additional References, page 282
- Feature Information for MPLS VPN Load Balancing Support for Inter-AS and CSC VPN, page 283

Finding Feature Information

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

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 Load Balancing Support for Inter-AS and CSC VPNs

Ensure that your Multiprotocol Label Switching (MPLS) virtual private network (VPN) network, including MPLS VPN interautonomous system (Inter-AS) or Carrier Supporting Carrier (CSC), is configured and working properly.

Restrictions for MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs

Load sharing using directly connected loopback peering does not apply to Carrier Supported Carrier (CSC) networks that use the Label Distribution Protocol (LDP) and an Interior Gateway Protocol (IGP) to distribute routes and Multiprotocol Label Switching (MPLS) labels.

The software does not support load balancing in interautonomous system (Inter-AS) and CSC when there are multiple links between provider edge (PE) or Autonomous System Boundary Router (ASBR) devices.

When you configure static routes in an MPLS or MPLS virtual private network (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 software releases that support the Tag Forwarding Information Base (TFIB. 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 software releases that support the MPLS Forwarding Infrastructure (MFI). 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 an 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 VPN 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 VPN 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 an MPLS VPN environment, and the next hop and interface are associated with the same virtual routing and forwarding (VRF) instance:

- **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 an 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 the interface are in the core.)

The following **ip route** commands are supported when you configure static routes in an 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 an 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 an 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 Device

The following **ip route vrf** command is supported when you configure static routes in an MPLS VPN environment, and the next hop is in the global table on the customer edge (CE) side. For example, the following command is supported when the destination-prefix is the CE device's loopback address, as in external Border Gateway Protocol (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 an MPLS VPN environment, the next hop is in the global table on the CE side, and you enable load sharing with static nonrecursive routes and a specific outbound interfaces:

- **ip route** *destination-prefix mask interface1 nexthop1*
- **ip route** *destination-prefix mask interface2 nexthop2*

Information About MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs

Load Sharing Using Directly Connected Loopback Peering

You use the MPLS VPN Load Balancing Support for Inter-AS and CSC VPN feature to load share traffic between adjacent label switched routers (LSRs) that are connected by multiple links. The LSRs could be a pair of Autonomous System Boundary Routers (ASBRs) or a carrier supporting carrier provider edge (CSC-PE) and a CSC-customer edge (CE).

Using directly connected loopback peering allows load sharing at the Interior Gateway Protocol (IGP) level so only one Border Gateway Protocol (BGP) session is needed between the LSRs. No other label distribution mechanism is needed between the adjacent LSRs except BGP.

Directly connected loopback peering enables load sharing of traffic as follows:

- A BGP session is established, using the loopback addresses of the LSRs.
- Multiprotocol Label Switching (MPLS) is enabled on the connecting links.
- Multiple static routes to the loopback address of the adjacent LSR allow IGP load sharing.
- The outgoing label to the loopback address of the adjacent LSR is an implicit null label and is inferred by the LSR.
- Because IGP load sharing is enabled on the loopback address of the adjacent LSR, any traffic destined to a prefix that is learned over the BGP session (and recurses over the loopback) is load shared.

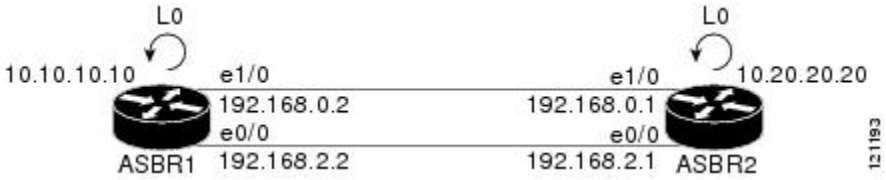
How to Configure MPLS VPN Load Balancing Support for Inter-AS and CSC VPN

Configuring Directly Connected Loopback Peering for MPLS VPN Inter-AS using ASBRs to Exchange VPN-IPv4 Addresses

This section describes the following tasks you need to do to configure peering of loopback interfaces of directly connected Autonomous System Boundary Routers (ASBRs):

The figure below shows the loopback configuration for directly connected ASBR1 and ASBR2. This configuration is used as the example in the tasks that follow.

Figure 27: Loopback Interface Configuration for Directly Connected ASBR1 and ASBR2



Configuring Loopback Interface Addresses for Directly Connected ASBRs

Perform this task to configure loopback interface addresses for directly connected Autonomous System Boundary Routers (ASBRs).

Note Loopback addresses need to be configured for each directly connected ASBR. That is, configure a loopback address for ASBR1 and for ASBR2 in the example shown in 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**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface- number* **Example:** `Device(config)# interface loopback 0`	Configures a software-only virtual interface that emulates an interface that is always up and enters interface configuration mode. • 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:** `Device(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:** `Device(config-if)# end`	Exits to privileged EXEC mode.

Configuring /32 Static Routes to the eBGP Neighbor Loopback

Perform this task to configure /32 static routes to the external Border Gateway Protocol (eBGP) neighbor loopback.

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 interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Device(config)# ip route 10.20.20.20 255.255.255.255 Ethernet 1/0 172.16.0.1`	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 and argument name a tag value that can be used as a "match" value for controlling redistribution through the use of route maps.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring Forwarding on Connecting Loopback Interfaces

Perform this task to configure forwarding on the connecting loopback interfaces.

This task is required for sessions between loopbacks. In the “Configuring /32 Static Routes to the eBGP Neighbor Loopback" section, Ethernet 1/0 and Ethernet 0/0 are the connecting interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **mpls bgp forwarding**
5. **exit**
6. Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Device(config)# interface ethernet 1/0`	Configures an interface type and enters interface configuration mode. • The *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* argument is the port number. Refer to the appropriate hardware manual for slot and port information.

	Command or Action	Purpose
Step 4	**mpls bgp forwarding** **Example:** `Device(config-if)# mpls bgp forwarding`	Configures the Border Gateway Protocol (BGP) to enable Multiprotocol Label Switching (MPLS) forwarding on connecting interfaces.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 6	Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).	
Step 7	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring an eBGP Session Between the Loopbacks

Perform this task to configure an external Border Gateway Protocol (eBGP) session between the loopbacks.

Note You need to configure an eBGP session between loopbacks on each directly connected Autonomous System Boundary Router (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**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 200`	Configures the BGP routing process. • The *as-number* indicates the number of an autonomous system that identifies the device to other BGP routers and tags the routing information passed along.
Step 4	**no bgp default route-target filter** **Example:** `Device(config)# no bgp default route-target filter`	Disables BGP route-target filtering, and enters router configuration mode. • All received BGP VPN-IPv4 routes are accepted by the device.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(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:** `Device(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.
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 10.20.20.20 update-source Loopback 0`	Allows BGP sessions 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.

	Command or Action	Purpose
		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:** `Device(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 **unicast** keyword specifies unicast prefixes.
Step 9	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Device(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 device. • 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. **Note** 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:** `Device(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 device. • 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:** `Device(config)# end`	Exits to privileged EXEC mode.

Verifying That Load Sharing Occurs Between Loopbacks

Perform this task to verify that load sharing occurs between loopbacks. You need to ensure that the Multiprotocol Label Switching (MPLS) Label Forwarding Information Base (LFIB) entry for the neighbor route lists the available paths and interfaces.

SUMMARY STEPS

1. **enable**
2. **show mpls forwarding-table** {*mask* | *length*} | **labels** *label* [*network label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]
3. **disable**

DETAILED STEPS

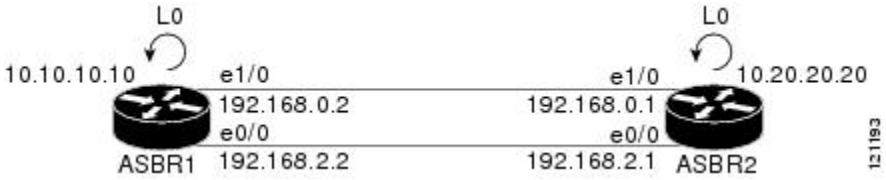
	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show mpls forwarding-table** {*mask* | *length*} | **labels** *label* [*network label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**] **Example:** `Device# show mpls forwarding-table`	Displays the contents of the MPLS LFIB. • Enter an optional keyword or argument if desired.
Step 3	**disable** **Example:** `Device# disable`	Exits to user EXEC mode.

Configuring Directly Connected Loopback Peering for MPLS VPN Inter-AS Using ASBRs to Exchange IPv4 Routes and Labels

The following sections describe how to configure peering of loopback interfaces of directly connected Autonomous System Boundary Routers (ASBRs) to achieve load sharing in an interautonomous system network:

The figure below shows the loopback configuration for directly connected ASBR1 and ASBR2. This configuration is used as the example in the tasks that follow.

Figure 28: Loopback Interface Configuration for Directly Connected ASBR1 and ASBR2



Configuring Loopback Interface Addresses for Directly Connected ASBRs

Note Loopback addresses need to be configured for each directly connected Autonomous System Boundary Router (ASBR). That is, configure a loopback address for ASBR1 and for ASBR2 as in the example shown in 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface number* **Example:** `Device(config)# interface loopback 0`	Configures a software-only virtual interface that emulates an interface that is always up and enters interface configuration mode. • 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.

	Command or Action	Purpose
Step 4	**ip address** *ip-address* [*mask* [**secondary**]] **Example:** `Device(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:** `Device(config-if)# end`	Exits to privileged EXEC mode.

Configuring /32 Static Routes to the eBGP Neighbor Loopback

Perform this task to configure /32 static routes to the external Border Gateway Protocol (eBGP) neighbor loopback.

Note You need to configure /32 static routes on each of the directly connected Autonomous System Boundary Routers (ASBRs).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*]
4. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Device(config)# ip route 10.20.20.20 255.255.255.255 Ethernet 1/0 172.16.0.1`	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 and argument name a tag value that can be used as a "match" value for controlling redistribution through the use of route maps.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring Forwarding on Connecting Loopback Interfaces

This task is required for sessions between loopbacks. In the "Configuring /32 Static Routes to the eBGP Neighbor Loopback" task, Ethernet1/0 and Ethernet0/0 are the connecting interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **mpls bgp forwarding**
5. **exit**
6. Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot/port* **Example:** `Device(config)# interface ethernet 1/0`	Configures an interface type and enters interface configuration mode. • The *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* argument is the port number. Refer to the appropriate hardware manual for slot and port information.
Step 4	**mpls bgp forwarding** **Example:** `Device(config-if)# mpls bgp forwarding`	Configures BGP to enable MPLS forwarding on connecting interfaces.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 6	Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).	

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring an eBGP Session Between the Loopbacks

Note You need to configure an external Border Gateway Protocol (eBGP) session between loopbacks on each directly connected Autonomous System Boundary Router (ASBR).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **bgp log-neighbor-changes**
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* | *peer-group-name*} **ebgp-multihop** [*ttl*]
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
9. **address-family ipv4** [**unicast**] **vrf** *vrf-name*
10. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
11. **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**
12. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 200`	Configures the BGP routing process, and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP routers and tags the routing information passed along.
Step 4	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(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 number of the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check** **Example:** `Device(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.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop** [*ttl*] **Example:** `Device(config-router)# neighbor bb.bb.bb.bb ebgp-multihop 255`	Accepts and attempts BGP connections to external peers residing on networks that are not directly connected. • The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument is the name of a BGP peer group. • The *ttl* argument the time-to-live in the range from 1 to 255 hops.
Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*	Allows BGP sessions to use any operational interface for TCP connections. • The *ip-address* argument is the IPv4 address of the BGP-speaking neighbor.

	Command or Action	Purpose
	Example: `Device(config-router)# neighbor 10.20.20.20 update-source Loopback 0`	• The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. **Note** 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 9	**address-family ipv4** [**unicast**] **vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode for configuring routing protocols such as BGP, Routing Information Protocol (RIP), and static routing. • The **unicast** keyword specifies unicast prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of a VPN routing/forwarding instance (VRF) to associate with submode commands.
Step 10	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Device(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 device. • The *peer-group-name* argument is the name of the BGP peer group. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. **Note** 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 11	**neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended** **Example:** `Device(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 device. • The *peer-group-name* argument is the name of the 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.

	Command or Action	Purpose
Step 12	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Verifying That Load Sharing Occurs Between Loopbacks

To verify that load sharing can occur between loopbacks, ensure that the Multiprotocol Label Switching (MPLS) Label Forwarding Information Base (LFIB) entry for the neighbor route lists the available paths and interfaces.

SUMMARY STEPS

1. **enable**
2. **show mpls forwarding-table** [*network* {*mask*|*length*} | **labels** *label* [*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]
3. **disable**

DETAILED STEPS

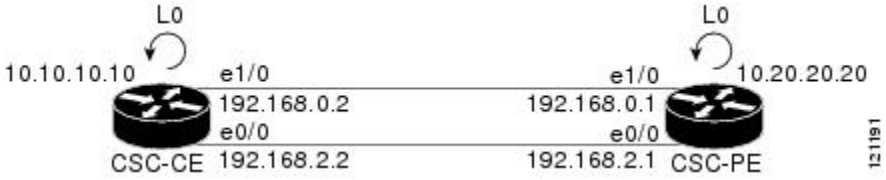
	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show mpls forwarding-table** [*network* {*mask*|*length*}	**labels** *label* [*label*]	**interface** *interface*	**next-hop** *address*	**lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**] **Example:** `Device# show mpls forwarding-table`	Displays the contents of the MPLS LFIB. • Enter a keyword or argument, if desired.
Step 3	**disable** **Example:** `Device# disable`	Exits to user EXEC mode.

Configuring Directly Connected Loopback Peering on MPLS VPN Carrier Supporting Carrier

The following sections explain how to load balance Carrier Supporting Carrier (CSC) traffic by peering loopback interfaces of directly connected CSC-provider edge (PE) and CSC-customer edge (CE) devices:

The figure below shows the loopback configuration for directly connected CSC-PE and CSC-CE devices. This configuration is used as the example in the tasks that follow.

Figure 29: Loopback Interface Configuration for Directly Connected CSC-PE and CSC-CE Devices



Configuring Loopback Interface Addresses on CSC-PE Devices

Note Configuration of a loopback interface address on the Carrier Supporting Carrier (CSC)-provider edge (PE) device requires the enabling of a virtual routing and forwarding (VRF) instance. The CSC-customer edge (CE) device loopback interface does not require enabling a VRF.

SUMMARY STEPS

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

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface number* **Example:** `Device(config)# interface loopback 0`	Configures a software-only virtual interface that emulates an interface that is always up, and enters interface configuration mode. • 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 vrf forwarding** *vrf-name* **Example:** `Device(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	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.20.20.20 255.255.255.255`	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 6	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring Loopback Interface Addresses for CSC-CE Routers

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface-number* **Example:** `Device(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:** `Device(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:** `Device(config-if)# end`	Exits to privileged EXEC mode.

Configuring /32 Static Routes to the eBGP Neighbor Loopback on the CSC-PE Device

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route vrf** *vrf-name prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**global**] [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Device(config)# ip route vrf vpn1 10.10.10.10 255.255.255.255 Ethernet 1/0 172.16.0.2`	Establishes static routes for a virtual routing and forwarding (VRF) instance. • The *vrf-name* argument is the name of the VRF for the static route. • 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 destination network. • The *interface-type* and *interface-number* arguments are the network interface type and interface number. • The **global** keyword specifies that the given next hop address is in the nonVRF routing table. • 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 and argument name a tag value that can be used as a “match” value for controlling redistribution via route maps.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring /32 Static Routes to the eBGP Neighbor Loopback on the CSC-CE Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Device(config)# ip route 10.20.20.20 255.255.255.255 Ethernet 1/0 172.16.0.1`	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 destination 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 and argument name a tag value that can be used as a "match" value for controlling redistribution via route maps.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring Forwarding on CSC-PE Interfaces That Connect to the CSC-CE Loopback

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot*/*port*
4. **ip vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask* [**secondary**]
6. **mpls bgp forwarding**
7. **exit**
8. Repeat Steps 3 through 6 for another connecting interface (Ethernet 0/0).
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot*/*port* **Example:** `Device(config)# interface ethernet 1/0`	Configures an interface type and enters interface configuration mode. • The *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* argument is the port number. Refer to the appropriate hardware manual for slot and port information.
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding vpn1`	Associates a virtual routing and forwarding (VRF) instance with an interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.

	Command or Action	Purpose
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 172.16.0.1 255.255.255.255`	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 6	**mpls bgp forwarding** **Example:** `Device(config-if)# mpls bgp forwarding`	Configures the Border Gateway Protocol (BGP) to enable Multiprotocol Label Switching (MPLS) forwarding on connecting interfaces.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 8	Repeat Steps 3 through 6 for another connecting interface (Ethernet 0/0).	
Step 9	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring Forwarding on CSC-CE Interfaces That Connect to the CSC-PE Loopback

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *typeslot/port*
4. **mpls bgp forwarding**
5. **exit**
6. Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *typeslot/port* **Example:** `Device(config)# interface ethernet 1/0`	Configures an interface type and enters interface configuration mode. • The *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* argument is the port number. Refer to the appropriate hardware manual for slot and port information.
Step 4	**mpls bgp forwarding** **Example:** `Device(config-if)# mpls bgp forwarding`	Configures the Border Gateway Protocol (BGP) to enable Multiprotocol Label Switching (MPLS) forwarding on connecting interfaces.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 6	Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).	
Step 7	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring an eBGP Session Between the CSC-PE Device and the CSC-CE Loopback

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **bgp log-neighbor-changes**
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 ipv4** [**unicast**] **vrf** *vrf-name*
9. **ip vrf forwarding** *vrf-name*
10. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
11. **neighbor** *ip-address* **send-label**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 200`	Configures the Border Gateway Protocol (BGP) routing process. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP routers and tags the routing information passed along.
Step 4	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument is the IP address of the neighbor.

	Command or Action	Purpose
	Example: `Device(config-router)# neighbor 10.10.10.10 remote-as 100`	• 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:** `Device(config-router)# neighbor 10.10.10.10 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.
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 10.10.10.10 update-source Loopback 0`	Allows BGP sessions 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 ipv4** [**unicast**] **vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf vpn1`	Enters address family configuration mode for configuring routing protocols such as BGP, Routing Information Protocol (RIP), and static routing. • The **ipv4** keyword configures sessions that carry standard IPv4 address prefixes. • The **unicast** keyword specifies unicast prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of a virtual routing and forwarding (VRF) instance to associate with submode commands.
Step 9	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-router-af)# ip vrf forwarding vpn1`	Associates a VRF with an interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.

	Command or Action	Purpose
Step 10	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Device(config-router-af)# neighbor 10.10.10.10 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring device. • The *peer-group-name* argument is the name of the BGP peer group. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. **Note** 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 11	**neighbor** *ip-address* **send-label** **Example:** `Device(config-router-af)# neighbor 10.10.10.10 send-label`	Enables a BGP device to send MPLS labels with BGP routes to a neighboring BGP device. • The *ip-address* argument is the IP address of the neighboring device.
Step 12	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Configuring an eBGP Session Between the CSC-CE Device and the CSC-PE Loopback

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **bgp log-neighbor-changes**
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 ipv4** [**unicast**] [**vrf** *vrf-name*]
9. **neighbor** {*ip-address* | *peer-group-name*|*ipv6-address*] **activate**
10. **neighbor** *ip-address* **send-label**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 200`	Configures the Border Gateway Protocol (BGP) routing process. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP routers and tags the routing information passed along.
Step 4	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(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:** `Device(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.
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 10.20.20.20 update-source Loopback 0`	Allows BGP sessions 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.

	Command or Action	Purpose
		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 ipv4** [**unicast**] [**vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode for configuring routing protocols such as BGP, RIP, and static routing. • The **ipv4** keyword configures sessions that carry standard IPv4 address prefixes. • The **unicast** keyword specifies unicast prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of a virtual routing and forwarding (VRF) instance to associate with submode commands.
Step 9	**neighbor** {*ip-address* | *peer-group-name*|*ipv6-address*] **activate** **Example:** `Device(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 device. • The *peer-group-name* argument is the name of the BGP peer group. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. **Note** 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* **send-label** **Example:** `Device(config-router-af)# neighbor 10.20.20.20 send-label`	Enables a BGP device to send Multiprotocol Label Switching (MPLS) labels with BGP routes to a neighboring BGP device. • The *ip-address* argument is the IP address of the neighboring device.
Step 11	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Verifying That Load Sharing Occurs Between Loopbacks

To verify that load sharing occurs between loopbacks, ensure that the Multiprotocol Label Switching (MPLS) Label Forwarding Information Base (LFIB) entry for the neighbor route lists the available paths and interfaces.

SUMMARY STEPS

1. **enable**
2. **show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [*-label*] | [**interface**] *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
3. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [*-label*] | [**interface**] *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:** `Device# show mpls forwarding-table`	Displays the contents of the MPLS LFIB.
Step 3	**disable** **Example:** `Device# disable`	Exits to user EXEC mode.

Configuration Examples for MPLS VPN Load Balancing Support for Inter-AS and CSC VPN

Examples: Configuring a 32 Static Route from an ASBR to the Loopback Address of Another ASBR

The following example configures a /32 static route from ASBR1 to the loopback address of ASBR2:

```
Device# configure terminal
Device(config)# ip route 10.20.20.20 255.255.255.255 e1/0 168.192.0.1
Device(config)# ip route 10.20.20.20 255.255.255.255 e0/0 168.192.2.1
```

The following example configures a /32 static route from ASBR2 to the loopback address of ASBR1:

```
Device# configure terminal
Device(config)# ip route vrf vpn1 10.10.10.10 255.255.255.255 e1/0 168.192.0.2
Device(config)# ip route vrf vpn1 10.10.10.10 255.255.255.255 e0/0 168.192.2.2
```

Example: Configuring BGP MPLS Forwarding on the Interfaces Connecting ASBRs

The following example configures the Border Gateway Protocol (BGP) and Multiprotocol Label Switching (MPLS) forwarding on the interfaces connecting ASBR2 with ASBR1:

```
Device# configure terminal
Device(config)# interface ethernet 1/0
Device(config-if)# ip vrf forwarding vpn1
Device(config-if)# ip address 168.192.0.1 255.255.255.255
Device(config-if)# mpls bgp forwarding
Device(config-if)# exit
Device(config)# interface ethernet 0/0
Device(config-if)# ip vrf forwarding vpn1
Device(config-if)# ip address 168.192.2.1 255.255.255.255
Device(config-if)# mpls bgp forwarding
Device(config-if)# exit
```

Example: Configuring VPNv4 Sessions on an ASBR

The following example configures VPNv4 sessions on ASBR2:

```
Device# configure terminal
Device(config)# router bgp 200
Device(config-router)# bgp log-neighbor-changes
Device(config-router)# neighbor 10.10.10.10 remote-as 100
Device(config-router)# neighbor 10.10.10.10 disable-connected-check
Device(config-router)# neighbor bb.bb.bb.bb ebgp-multihop 255
Device(config-router)# neighbor 10.10.10.10 update-source Loopback0
!
Device(config-router)# address-family vpnv4
Device(config-router-af)# neighbor 10.10.10.10 activate
Device(config-router-af)# neighbor 10.10.10.10 send-community extended
Device(config-router-af)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
Configuring MPLS VPN CSC with BGP	"MPLS VPN Carrier Supporting Carrier with BGP" module in the *MPLS: Layer 3 VPNs: Inter-AS and CSC Configuration Guide*
Configuring BGP	"Configuring BGP" module in the *IP Routing: BGP Configuration Guide*
Configuring BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN	"BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN" module in the *IP Routing: BGP Configuration Guide*

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 2373	IP Version 6 Addressing Architecture
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 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

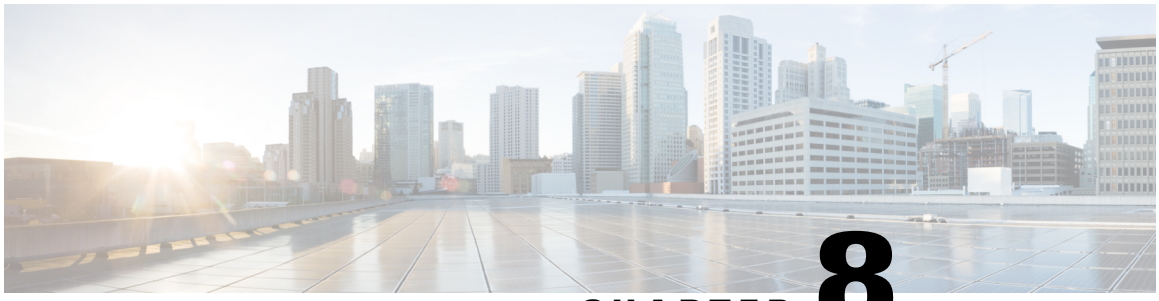
Feature Information for MPLS VPN Load Balancing Support for Inter-AS and CSC VPN

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 12: Feature Information for MPLS VPN Load Balancing Support for Inter-AS and CSC VPN

Feature Name	Releases	Feature Information
MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs	12.0(29)S 12.4(20)T 12.2(33)SRA 12.2(33)SXH Cisco IOS XE Release 2.2	The MPLS VPN Load Balancing Support for Inter-AS and CSC VPNs feature allows MPLS VPN Inter-AS and MPLS VPN 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. In Cisco IOS Release 12.0(29)S, this feature was introduced. In Cisco IOS Release 12.4(20)T, 12.2(33)SRA, and 12.2(33)SXH, this feature was integrated. In Cisco IOS XE Release 2.2, this feature was implemented on the Cisco ASR 1000 Series Routers. No commands were introduced or modified.



CHAPTER 8

MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs

The MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs feature enables you to configure external Border Gateway Protocol (eBGP) multipath with IPv4 labels. This creates an entry in the Multiprotocol Label Switching (MPLS) forwarding table with label information for each outgoing path installed in the routing table thereby allowing redundant connectivity and load balancing. Without this feature, the MPLS forwarding table contains the labels only for the BGP best path even though the routing table has more than one path for the prefix.

- Finding Feature Information, page 285
- Prerequisites for MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs, page 286
- Restrictions for MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs , page 286
- Information About MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs, page 288
- How to Configure MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs, page 288
- Configuration Examples for MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs, page 296
- Additional References, page 297
- Feature Information for MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs, page 298

Finding Feature Information

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

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 eBGP Multipath Support for CSC and Inter-AS MPLS VPNs

Ensure that your Multiprotocol Label Switching (MPLS) virtual private network (VPN) network, including MPLS VPN interautonomous system (Inter-AS) or Carrier Supporting Carrier (CSC), is configured and working properly.

Restrictions for MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs

The MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs feature is not supported on Multiprotocol Label Switching (MPLS) virtual private network (VPN) interautonomous system (Inter-AS) with Autonomous System Boundary Routers (ASBRs) that exchange VPNv4 routes.

When you configure static routes in an MPLS or MPLS virtual private network (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 software releases that support the Tag Forwarding Information Base (TFIB. 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 software releases that support the MPLS Forwarding Infrastructure (MFI). 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 an 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 VPN 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 VPN 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 an MPLS VPN environment, and the next hop and interface are associated with the same virtual routing and forwarding (VRF) instance:

- **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 an 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 the interface are in the core.)

The following **ip route** commands are supported when you configure static routes in an 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 an 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 an 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 Device

The following **ip route vrf** command is supported when you configure static routes in an MPLS VPN environment, and the next hop is in the global table on the customer edge (CE) side. For example, the following command is supported when the destination-prefix is the CE device's loopback address, as in external Border Gateway Protocol (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 an MPLS VPN environment, the next hop is in the global table on the CE side, and you enable load sharing with static nonrecursive routes and a specific outbound interfaces:

- **ip route** *destination-prefix mask interface1 nexthop1*
- **ip route** *destination-prefix mask interface2 nexthop2*

Information About MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs

Overview of MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs

When a device learns two identical external Border Gateway Protocol (eBGP) paths for a prefix from a neighboring autonomous system, it chooses the path with the lower route ID as the best path. This best path is installed in the IP routing table. You can enable eBGP multipath, which installs multiple paths in the IP routing table (instead of picking one best path) when the eBGP paths are learned from a neighboring autonomous system.

During packet switching, depending on the switching mode, either per-packet or per-destination load sharing is performed among the multiple paths. The **maximum-paths** router configuration command controls the number of paths allowed. By default, BGP installs only one path to the IP routing table.

How to Configure MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs

Configuring MPLS VPN eBGP Multipath Load Sharing with Inter-AS MPLS VPNs

Perform this task on the Autonomous System Boundary Routers (ASBRs) to configure external Border Gateway Protocol (eBGP) multipath for Multiprotocol Label Switching (MPLS) virtual private network (VPN) interautonomous systems with ASBRs exchanging IPv4 routes and MPLS labels.

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 ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
6. **maximum-paths** *number-paths*
7. **neighbor** {*ip-address* | *peer-group-name*} **activate**
8. **neighbor** *ip-address* **send-label**
9. **exit-address-family**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 100`	Configures a BGP routing process and places the device in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP routers and tags the routing information passed along. The range is 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:** `Device(config-router)# 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 5	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv4 address prefixes.

	Command or Action	Purpose
	Example: `Device(config-router)# address-family ipv4`	• 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 6	**maximum-paths** *number-paths* **Example:** `Device(config-router-af)# maximum-paths 2`	(Optional) Controls the maximum number of parallel routes an IP routing protocol can support. • The *number-paths* argument specifies the maximum number of parallel routes an IP routing protocol installs in a routing table.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a neighboring device. • 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 8	**neighbor** *ip-address* **send-label** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 send-label`	Enables a BGP device to send MPLS labels with BGP routes to a neighboring BGP device. • The *ip-address* argument specifies the IP address of the neighboring device.
Step 9	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 10	**end** **Example:** `Device(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Configuring MPLS VPN eBGP Multipath Load Sharing with Carrier Supporting Carrier on the CSC-PE Devices

Perform this task to configure external Border Gateway Protocol (eBGP) multipath load sharing on the carrier supporting carrier-provider edge (CSC-PE) devices that distribute BGP routes with Multiprotocol Label Switching (MPLS) labels.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(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 device to other BGP routers and tags the routing information passed along. The range is 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*]	Specifies the IPv4 address family type and enters address family configuration mode.

	Command or Action	Purpose
	Example: `Device(config-router)# address-family ipv4 vrf vpn1`	• 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	**maximum-paths** *number-paths* **Example:** `Device(config-router-af)# maximum-paths 2`	(Optional) Controls the maximum number of parallel routes an IP routing protocol can support. • On the CSC-PE device, this command is enabled in address family configuration mode. • The *number-paths* argument specifies the maximum number of parallel routes an IP routing protocol installs in a routing table.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(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 7	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a neighboring BGP device. • 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 8	**neighbor** *ip-address* **as-override** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 as-override`	Configures a PE device to override the autonomous system number (ASN) of a site with the ASN of a provider. • The *ip-address* argument specifies the IP address of the device that is to be overridden with the ASN provided.
Step 9	**neighbor** *ip-address* **send-label** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 send-label`	Enables a BGP device to send MPLS labels with BGP routes to a neighboring BGP device. • The *ip-address* argument specifies the IP address of the neighboring device.

	Command or Action	Purpose
Step 10	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 11	**end** **Example:** `Device(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Configuring MPLS VPN eBGP Multipath Load Sharing with Carrier Supporting Carrier on the CSC-CE Devices

Perform this task to configure external Border Gateway Protocol (eBGP) multipath load sharing on the carrier supporting carrier-customer edge (CSC-CE) devices.

SUMMARY STEPS

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

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 200`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP routers and tags the routing information passed along. The range is 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**maximum-paths** *number-paths* **Example:** `Device(config-router)# maximum-paths 2`	(Optional) Controls the maximum number of parallel routes an IP routing protocol can support. • On the CSC-CE routers, this command is issued in router configuration mode. • The *number-paths* argument specifies the maximum number of parallel routes an IP routing protocol installs in a routing table.
Step 5	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4`	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 6	**redistribute** *protocol* **Example:** `Device(config-router-af)# redistribute static`	Redistributes routes from one routing domain into another routing domain. • The *protocol* argument specifies the source protocol from which routes are being redistributed. It can be one of the following keywords: **bgp**, **connected**, **egp**, **igrp**, **isis**, **mobile**, **ospf**, **rip**, and **static** [**ip**]. • The **static** [**ip**] keyword redistributes IP static routes. **Note** The optional **ip** keyword is used when you redistribute static routes into Intermediate System- to-Intermediate System (IS-IS). • • The **connected** keyword refers to routes that are established automatically when IP is enabled on an interface. • For routing protocols such as Open Shortest Path First (OSPF) and Intermediate System-to-Intermediate System (IS-IS), these routes are redistributed as external to the autonomous system.

	Command or Action	Purpose
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router-af)# neighbor 10.0.0.2 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 8	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 10.0.0.2 activate`	Enables the exchange of information with a neighboring BGP device. • 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 9	**neighbor** *ip-address* **send-label** **Example:** `Device(config-router-af)# neighbor 10.0.0.2 send-label`	Enables a BGP device to send Multiprotocol Label Switching (MPLS) labels with BGP routes to a neighboring BGP device. • The *ip-address* argument specifies the IP address of the neighboring device.
Step 10	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 11	**end** **Example:** `Device(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Configuration Examples for MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs

Example: Configuring MPLS VPN eBGP Multipath Load Sharing with MPLS VPN Inter-AS

The following example shows how to configure external Border Gateway Protocol (eBGP) multipath for Multiprotocol Label Switching (MPLS) virtual private network (VPN) interautonomous systems with Autonomous System Boundary Routers (ASBRs) exchanging IPv4 routes and MPLS labels:

```
Device# configure terminal
Device(config)# router bgp 100
Device(config-router)# neighbor 10.0.0.1 remote-as 200
Device(config-router)# address-family ipv4
Device(config-router-af)# maximum-paths 2
Device(config-router-af)# neighbor 10.0.0.1 activate
Device(config-router-af)# neighbor 10.0.0.1 send-label
Device(config-router-af)# exit-address-family
Device(config-router-af)# end
```

Example: Configuring MPLS VPN eBGP Multipath Load Sharing with MPLS VPN Carrier Supporting Carrier on the CSC-PE Devices

The following example shows how to configure external Border Gateway Protocol (eBGP) multipath load sharing on the carrier supporting carrier-provider edge (CSC-PE) devices that distribute BGP routes with Multiprotocol Label Switching (MPLS) labels:

```
Device# configure terminal
Device(config)# router bgp 100
Device(config-router)# address-family ipv4 vrf vpn1
Device(config-router-af)# maximum-paths 2
Device(config-router-af)# neighbor 10.0.0.1 remote-as 200
Device(config-router-af)# neighbor 10.0.0.1 activate
Device(config-router-af)# neighbor 10.0.0.1 as-override
Device(config-router-af)# neighbor 10.0.0.1 send-label
Device(config-router-af)# exit-address-family
Device(config-router)# end
```

Example: Configuring MPLS VPN eBGP Multipath Load Sharing with MPLS VPN Carrier Supporting Carrier on the CSC-CE Devices

The following example shows how to configure external Border Gateway Protocol (eBGP) multipath load sharing on the carrier supporting carrier-customer edge (CSC-CE) devices:

```
Device# configure terminal
Device(config)# router bgp 200
Device(config-router)# maximum-paths 2
Device(config-router)# address-family ipv4
Device(config-router-af)# redistribute static
```

```
Device(config-router-af)# neighbor 10.0.0.2 remote-as 100
Device(config-router-af)# neighbor 10.0.0.2 activate
Device(config-router-af)# neighbor 10.0.0.2 send-label
Device(config-router-af)# exit-address-family
Device(config-router)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
Configuring MPLS VPN CSC with BGP	"MPLS VPN Carrier Supporting Carrier with BGP" module in the *MPLS: Layer 3 VPNs: Inter-AS and CSC Configuration Guide*
Configuring BGP	"Configuring BGP" module in the *IP Routing: BGP Configuration Guide*
Configuring BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN	"BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN" module in the *IP Routing: BGP Configuration Guide*

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 2373	IP Version 6 Addressing Architecture
RFC 2547	BGP/MPLS VPNs
RFC 2842	Capabilities Advertisement with BGP-4

RFC	Title
RFC 2858	Multiprotocol Extensions for BGP-4
RFC 3107	Carrying Label Information in BGP-4

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 eBGP Multipath Support for CSC and Inter-AS 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 13: Feature Information for MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs

Feature Name	Releases	Feature Information
MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs	12.0(27)S 12.2(30)S 12.2(33)SRA 12.2(33)SXH Cisco IOS XE Release 2.2	The MPLS VPN eBGP Multipath Support for CSC and Inter-AS MPLS VPNs feature installs multiple paths in the IP routing table when the eBGP paths are learned from a neighboring Autonomous System (AS), instead of picking one best path. In Cisco IOS Release 12.0(27)S, this feature was introduced. In Cisco IOS Release 12.2(30)S, 12.2(33)SRA, and 12.2(33)SXH, this feature was integrated. In Cisco IOS XE Release 2.2, this feature was implemented on the Cisco ASR 1000 Series Routers. No commands were introduced or modified.