MPLS Layer 3 VPNs: Inter-AS and CSC Configuration Guide, Cisco IOS Release 12.2SX

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

MPLS VPN - Interautonomous System Support

An autonomous system is a single network or group of networks that is controlled by a common system administration group and that uses a single, clearly defined routing protocol. The MPLS VPN - Interautonomous System Support feature allows an Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) to span service providers and autonomous systems.

This document explains how to enable Autonomous System Boundary Routers (ASBRs) to use exterior Border Gateway Protocol (eBGP) to exchange IPv4 Network Layer Reachability Information (NLRI) in the form of VPN-IPv4 addresses.

As VPNs grow, their requirements expand. In some cases, VPNs need to reside on different autonomous systems in different geographic areas. Also, some VPNs need to extend across multiple service providers (overlapping VPNs). Regardless of the complexity and location of the VPNs, the connection between autonomous systems must be seamless to the customer. The MPLS VPN - Interautonomous System Support feature provides this functionality.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the Feature Information for MPLS VPN - Interautonomous System Support, page 57.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Finding Feature Information

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

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

Prerequisites for MPLS VPN - Interautonomous System Support

Before you configure eBGP routing between autonomous systems or subautonomous systems in an MPLS VPN, ensure that you have properly configured all MPLS VPN routing instances and sessions. The configuration tasks outlined in the How to Configure MPLS VPN - Interautonomous System Support, page 13 build from those configuration tasks.

Perform (as appropriate to the existing network configuration) the following tasks as described in the the Configuring MPLS VPNs feature module.

- Define VPN routing instances
- Configure BGP routing sessions in the service provider (P) network
- Configure provider edge (PE) to PE routing sessions in the service provider (P) network
- Configure BGP PE to customer edge (CE) routing sessions

A VPN-IPv4 eBGP session must be configured between directly connected ASBRs.

This feature is supported on the Cisco IOS 12000 series line cards listed in the table below.

Table 1 Cisco I2000 Series Line Card Support Added for Cisco IOS Releases

Type	Line Cards	Cisco IOS Release Added
Packet over SONET (POS)	4-Port OC-3 POS	12.0(16)ST
	1-Port OC-12 POS	12.0(17)ST
	8-Port OC-3 POS	12.0(22)S
	16-Port OC-3 POS	
	4-Port OC-12 POS	
	1-Port OC-48 POS	
	4-Port OC-3 POS ISE	
	8-Port OC-3 POS ISE	
	16-Port OC-3 POS ISE	
	4-Port OC-12 POS ISE	
	1-Port OC-48 POS ISE	

Type	Line Cards	Cisco IOS Release Added
Electrical Interface	6-Port DS3	12.0(21ST
	12-Port DS3	12.0(22)S
	6-Port E3	
	12-Port E3	
Ethernet	3-Port GbE	12.0(23)S
	1-Port 10-GbE Modular GbE/FE	12.0(24)S
ATM	4-Port OC-3 ATM	12.0(16)ST
	1-Port OC12 ATM	12.0(17)ST
	4-Port OC-12 ATM	12.0(23)S
	8-Port OC-3 ATM	
Channelized Interface	2-Port CHOC-3	12.0(22)S
	6-Port Ch T3 (DS1)	
	1-Port CHOC-12 (DS3)	
	1-Port CHOC-12 (OC-3)	
	4-Port CHOC-12 ISE	
	1-Port CHOC-48 ISE	

Restrictions for MPLS VPN - Interautonomous System Support

Note the following restrictions to the MPLS VPN - Interautonomous System Support feature:

- A VPN-IPv4 eBGP session must be configured between directly connected ASBRs.
- For networks configured with eBGP multihop, a label switched path (LSP) must be established between nonadjacent routers (RFC 3107).
- PPP encapsulation on the ASBRs is not supported with this feature.

Information About MPLS VPN - Interautonomous System Support

MPLS VPN Interautonomous System Benefits

An MPLS VPN Inter-AS provides the following benefits:

- Allows a VPN to cross more than one service provider backbone—Service providers running separate autonomous systems can jointly offer MPLS VPN services to the same end customer. A VPN can begin at one customer site and traverse different VPN service provider backbones before arriving at another site of the same customer. Before the release of this feature, MPLS VPN could only traverse a single BGP autonomous system service provider backbone. The MPLS VPN - Interautonomous System Support feature allows multiple autonomous systems to form a continuous (and seamless) network between customer sites of a service provider.
- Allows a VPN to exist in different areas—A service provider can create a VPN in different geographic areas. Having all VPN traffic flow through one point (between the areas) allows for better rate control of network traffic between the areas.
- Allows confederations to optimize internal Border Gateway Protocol (iBGP) meshing—iBGP meshing in an autonomous system is more organized and manageable. You can divide an autonomous system into multiple, separate subautonomous systems and then classify them into a single confederation (even though the entire VPN backbone appears as a single autonomous system). This capability allows a service provider to offer MPLS VPNs across the confederation because it supports the exchange of labeled VPN-IPv4 NLRI between the subautonomous systems that form the confederation.

Interautonomous System Communication with ASBRs

Separate autonomous systems from different service providers can communicate by exchanging IPv4 NLRI in the form of VPN-IPv4 addresses. The ASBRs use eBGP to exchange that information. Then an Interior Gateway Protocol (IGP) distributes the network layer information for VPN-IPv4 prefixes throughout each VPN and each autonomous system. Routing information uses the following protocols:

- Within an autonomous system, routing information is shared using an IGP.
- Between autonomous systems, routing information is shared using an eBGP. An eBGP allows a service provider to set up an interdomain routing system that guarantees the loop-free exchange of routing information between separate autonomous systems.

The primary function of an eBGP is to exchange network reachability information between autonomous systems, including information about the list of autonomous system routes. The autonomous systems use EGBP border edge routers to distribute the routes, which include label switching information. Each border edge router rewrites the next hop and MPLS labels. See the How Information Is Exchanged in an MPLS VPN Inter-AS with ASBRs, page 5 section for more information.

Interautonomous System Configurations Supported in an MPLS VPN

Interautonomous system configurations supported in an MPLS VPN can include:

- Interprovider VPN—MPLS VPNs that include two or more autonomous systems, connected by separate border edge routers. The autonomous systems exchange routes using eBGP. No IGP or routing information is exchanged between the autonomous systems.
- BGP confederations—MPLS VPNs that divide a single autonomous system into multiple subautonomous systems, and classify them as a single, designated confederation. The network recognizes the confederation as a single autonomous system. The peers in the different autonomous systems communicate over eBGP sessions; however, they can exchange route information as if they were iBGP peers.

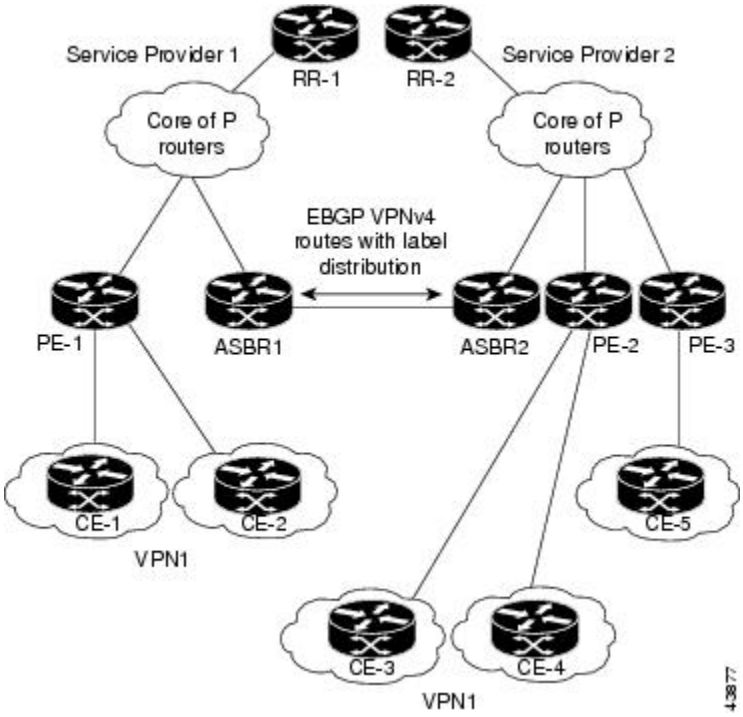
How Information Is Exchanged in an MPLS VPN Inter-AS with ASBRs

This section contains the following topics about how information is exchanged in an MPLS VPN Inter-AS with ASBRs exchanging VPN-IPv4 addresses:

Information Sent in an MPLS VPN Inter-AS with ASBRs

The figure below illustrates one MPLS VPN consisting of two separate autonomous systems. Each autonomous system operates under different administrative control and runs a different IGP. Service providers exchange routing information through eBGP border edge routers (ASBR1, ASBR2).

Figure 1 eBGP Connection Between Two MPLS VPN Interautonomous Systems with ASBRs Exchanging VPN-IPv4 Addresses



The table below describes the process to transmit information in an Inter-As configuration with ASBRs exchanging VPN-IPv4 addresses.

Table 2 Information Transmission Process in an Inter-AS with ASBRs Exchanging VPN-IPv4 AddressesMPLS VPN - Interautonomous System Support

Inter-AS Component	Process Competed During Information Transmission
Provider edge router: PE-1	Assigns a label for a route before distributing that route. The PE router uses the multiprotocol extensions of BGP to transmit label mapping information. The PE router distributes the route as a VPN-IPv4 address. The address label and the VPN identifier are encoded as part of the NLRI.
Route reflectors: RR-1 and RR-2	Reflects VPN-IPv4 internal routes within the autonomous system. The autonomous systems' border edge routers (ASBR1 and ASBR2) advertise the VPN-IPv4 external routes.
eBGP border edge router: ASBR1	Redistributes the route to the next autonomous system (ASBR2). ASBR1 specifies its own address as the value of the eBGP next-hop attribute and assigns a new label. The address ensures the following: • That the next-hop router is always reachable in the service provider (P) backbone network. • That the label assigned by the distributing router is properly interpreted. (The label associated with a route must be assigned by the corresponding next-hop router.)
eBGP border edge router: ASBR2	Redistributes the route in one of the following ways, depending on its configuration: • If the iBGP neighbors are configured with the **neighbor next-hop-self** command, ASBR2 changes the next-hop address of updates received from the eBGP peer, then forwards it. • If the iBGP neighbors are not configured with the **neighbor next-hop-self** command, the next-hop address does not get changed. ASBR2 must propagate a host route for the eBGP peer through the IGP. To propagate the eBGP VPN-IPv4 neighbor host route, use the **redistribute connected subnets** command. The eBGP VPN-IPv4 neighbor host route is automatically installed in the routing table when the neighbor comes up. This is essential to establish the label-switched path between PE routers in different autonomous systems

VPN Routing Information Exchange in an MPLS VPN Inter-AS with ASBRs

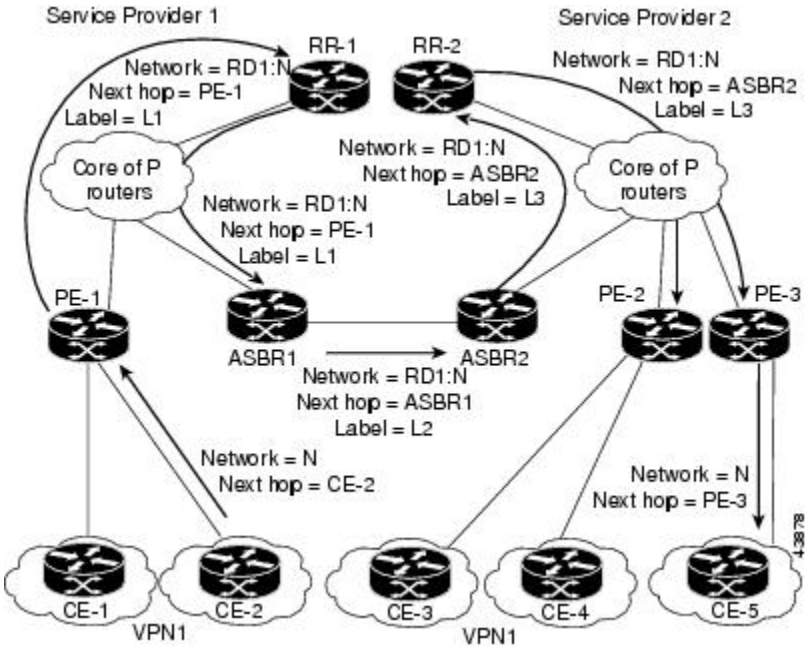
Autonomous systems exchange VPN routing information (routes and labels) to establish connections. To control connections between autonomous systems, the PE routers and eBGP border edge routers maintain a Label Forwarding Information Base (LFIB).

The LFIB manages the labels and routes that the PE routers and eBGP border edge routers receive during the exchange of VPN information.

The figure below illustrates the exchange of VPN route and label information between autonomous systems. The autonomous systems use the following guidelines to exchange VPN routing information:

- Routing information:
  - The destination network (N)
  - The next-hop field associated with the distributing router
  - A local MPLS label (L)
- An RD1: route distinguisher is part of a destination network address. It makes the VPN-IPv4 route globally unique in the VPN service provider environment.
- The ASBRs are configured to change the next hop (next-hop-self) when sending VPN-IPv4 NLRIs to the iBGP neighbors. Therefore, the ASBRs must allocate a new label when they forward the NLRI to the iBGP neighbors.

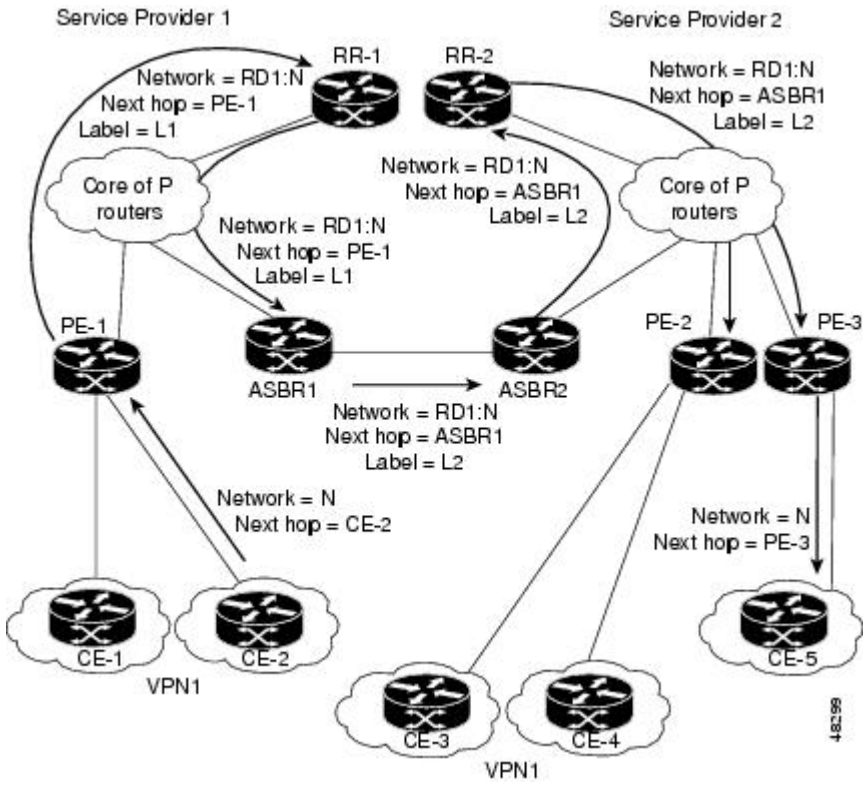
Figure 2 Exchanging Routes and Labels Between MPLS VPN Inter-AS Systems with ASBRs Exchanging VPN-IPv4 Addresses



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 redistributed connected Command in an MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses



Packet Forwarding Between MPLS VPN Interautonomous Systems with ASBRs

The figure below illustrates how packets are forwarded between autonomous systems in an interprovider network using the following packet forwarding method.

Packets are forwarded to their destination by means of MPLS. Packets use the routing information stored in the LFIB of each PE router and eBGP border edge router.

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

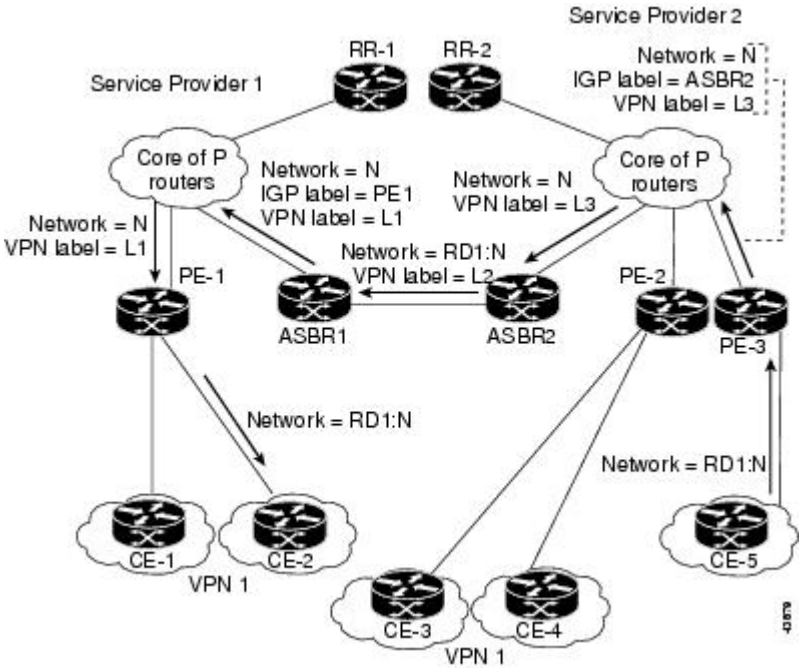
Each autonomous system uses standard multilevel labeling to forward packets between the edges of the autonomous system routers (for example, from CE-5 to PE-3). Between autonomous systems, only a single level of labeling is used, corresponding to the advertised route.

A data packet carries two levels of labels when traversing the VPN backbone:

- The first label (IGP route label) directs the packet to the correct PE router or eBGP border edge router. (For example, the IGP label of ASBR2 points to the ASBR2 border edge router.)

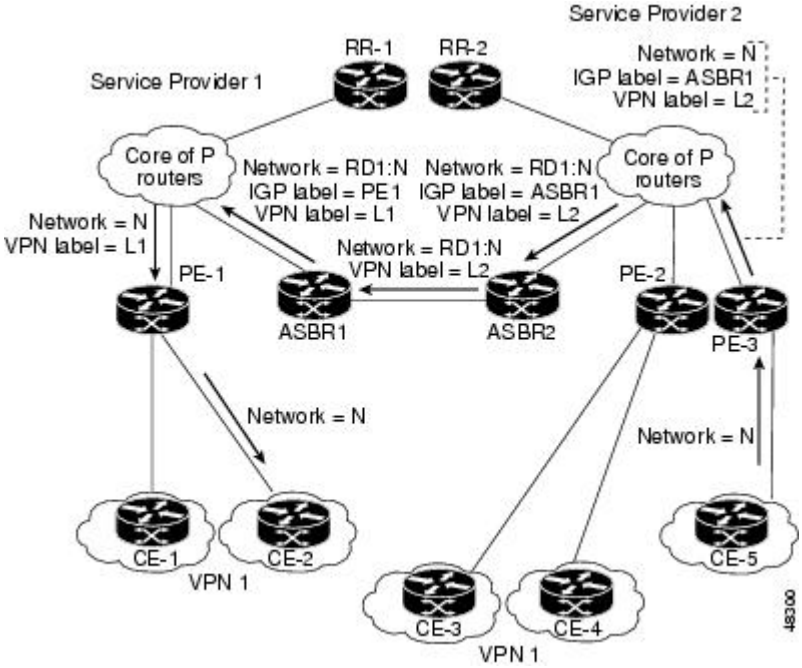
- The second label (VPN route label) directs the packet to the appropriate PE router or eBGP border edge router.

Figure 4 Packet Forwarding Between MPLS VPN Interautonomous Systems with ASBRs Exchanging VPN-IPv4 Addresses



The figure below shows the same packet forwarding method, except the eBGP router (ASBR1) forwards the packet without reassigning it a new label.

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



Confederation Configuration for MPLS VPN Inter-AS with ASBRs

A confederation is multiple subautonomous systems grouped together. A confederation reduces the total number of peer devices in an autonomous system. A confederation divides an autonomous system into subautonomous systems and assigns a confederation identifier to the autonomous systems. A VPN can span service providers running in separate autonomous systems or in multiple subautonomous systems that form a confederation.

In a confederation, each subautonomous system is fully meshed with other subautonomous systems. The subautonomous systems communicate using an IGP, such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS). Each subautonomous system also has an eBGP connection to the other subautonomous systems. The confederation eBGP (CeBGP) border edge routers forward next-hop-self addresses between the specified subautonomous systems. The next-hop-self address forces the BGP to use a specified address as the next hop rather than letting the protocol choose the next hop.

You can configure a confederation with separate subautonomous systems in either of two ways:

- You can configure a router to forward next-hop-self addresses between only the CeBGP border edge routers (both directions). The subautonomous systems (iBGP peers) at the subautonomous system border do not forward the next-hop-self address. Each subautonomous system runs as a single IGP domain. However, the CeBGP border edge router addresses are known in the IGP domains.
- You can configure a router to forward next-hop-self addresses between the CeBGP border edge routers (both directions) and within the iBGP peers at the subautonomous system border. Each subautonomous system runs as a single IGP domain but also forwards next-hop-self addresses between the PE routers in the domain. The CeBGP border edge router addresses are known in the IGP domains.

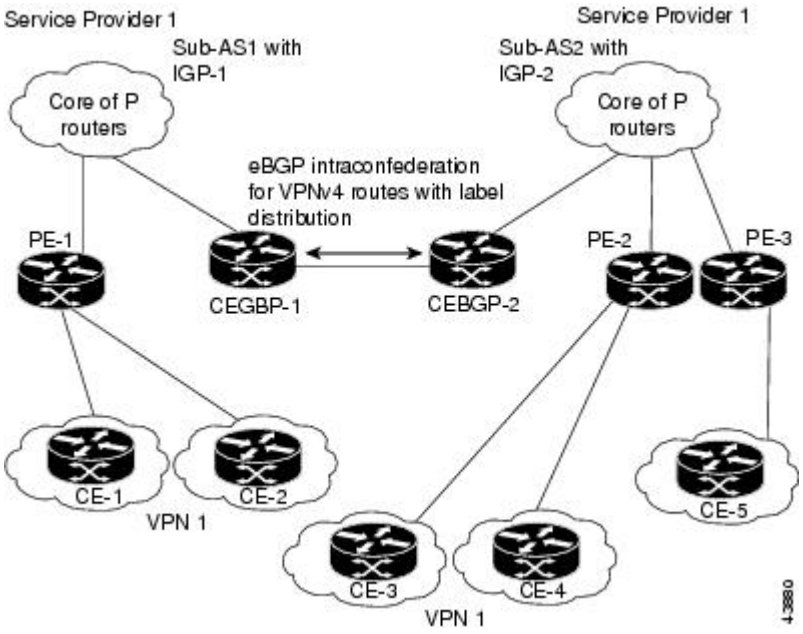
Note The second and third figures above illustrate how two autonomous systems exchange routes and forward packets. Subautonomous systems in a confederation use a similar method of exchanging routes and forwarding packets.

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

- The two CeBGP border edge routers exchange VPN-IPv4 addresses with labels between the two subautonomous systems.
- The distributing router changes the next-hop addresses and labels and uses a next-hop-self address.

IGP-1 and IGP-2 know the addresses of CeBGP-1 and CeBGP-2.

Figure 6 eBGP Connection Between Two Subautonomous Systems in a Confederation



In this confederation configuration:

- CeBGP border edge routers function as neighboring peers between the subautonomous systems. The subautonomous systems use eBGP to exchange route information.
- Each CeBGP border edge router (CeBGP-1, CeBGP-2) assigns a label for the route before distributing the route to the next subautonomous system. The CeBGP border edge router distributes the route as a VPN-IPv4 address by using the multiprotocol extensions of BGP. The label and the VPN identifier are encoded as part of the NLRI.
- Each PE and CeBGP border edge router assigns its own label to each VPN-IPv4 address prefix before redistributing the routes. The CeBGP border edge routers exchange VPN-IPv4 addresses with the labels. The next-hop-self address is included in the label (as the value of the eBGP next-hop attribute). Within the subautonomous systems, the CeBGP border edge router address is distributed throughout the iBGP neighbors, and the two CeBGP border edge routers are known to both confederations.

Load Sharing with MPLS VPN Inter-AS ASBRs

Before the MPLS VPN - Interautonomous System Support feature, if multiple paths existed across ASBRs, BGP executed the best path algorithm and marked only one of the paths as the best path. This path was added to the routing table and became the only path that was used for forwarding traffic between ASBRs.

The MPLS VPN—Multipath Support for Inter-AS VPNs feature extends the functionality of BGP so that it can pick one path as the best path and mark the other legitimate paths between ASBRs as multipath. This allows the load sharing of traffic among the different multipaths and the best path to reach the destination. No Routing Information Base (RIB) or Cisco Express Forwarding entries are associated with the VPN-IPv4 prefixes.

The MPLS VPN—Multipath Support for Inter-AS VPNs feature applies to ASBRs that do not have a VPN routing and forwarding (VRF) instance configuration. BGP installs a number of learned VPN-IPv4 prefixes

into the MPLS forwarding table (LFIB). VPN-IPv4 entries in the LFIB consist of the Route Distinguisher (RD) and the IPv4 prefix and are called VPNv4 entries.

The **maximum-paths** command is used to set the number of parallel (equal-cost) routes that BGP installs in the routing table to configure multipath load sharing. The number of paths that can be configured is determined by the version of Cisco IOS software. The following list shows the limits:

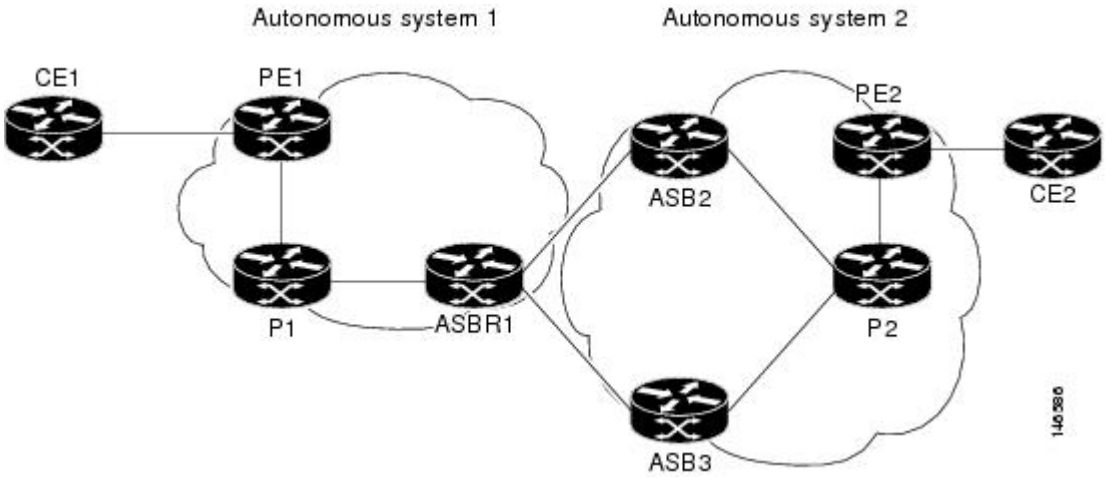
- Cisco IOS Release 12.0S-based software: 8 paths
- Cisco IOS Release 12.3T-based software: 16 paths
- Cisco IOS Release 12.2S-based software: 32 paths

The MPLS VPN—Multipath Support for Inter-AS VPNs feature requires that you configure the **maximum-paths** *number-of-paths* command in address family configuration mode.

Note The **maximum-paths** command cannot be configured with the **maximum-paths eibgp** command for the same BGP routing process.

The figure below shows an example of VPNv4 load balancing for ASBRs in an Inter-AS network. In this example, ASBR1 load balances the traffic from the CE router CE1 to CE2 using the two available links—ASBR2 and ASBR3.

Figure 7 Example of VPNv4 Load Balancing for ASBRs in an Inter-AS Network



When you configure an ASBR for VPNv4 load balancing, you must configure the **next-hop-self** command for the iBGP peers. Without this command, the next hop that is propagated to the iBGP peer is the ASBR2 address or the ASBR3 address, depending on which one BGP selects as the best path. Configuring the **next-hop-self** command provides direct VPNv4 forwarding entries in the MPLS forwarding table for the VPNv4 prefixes learned from the remote ASBRs. VPNv4 forwarding entries are not created if you do not configure the **next-hop-self** command.

Note If the number of forwarding entries in the MPLS forwarding table on the system or on a line card is a concern for your network, we recommend that you do not enable VPNv4 multipath on ASBRs.

How to Configure MPLS VPN - Interautonomous System Support

Perform the following tasks to configure MPLS VPN Inter-AS with ASBRs exchanging VPN-IPv4 addresses:

Configuring an eBGP ASBR to Exchange MPLS VPN-IPv4 Addresses

Perform one of the following tasks to configure an eBGP ASBR to exchange MPLS VPN-IPv4 routes with another autonomous system:

Configuring Peering with Directly Connected Interfaces Between ASBRs

Perform this task to configure peering with directly connected interfaces between ASBRs so that the ASBRs can distribute BGP routes with MPLS labels.

The figure below shows the configuration for the peering with directly connected interfaces between ASBRs. This configuration is used as the example in the tasks that follow.

Figure 8 Configuration for Peering with Directly Connected Interfaces Between ASBRs

Note When eBGP sessions come up, BGP automatically generates the **mpls bgp forwarding** command on the connecting interface.

Note Issue the **redistribute connected subnets** command in the IGP configuration portion of the router to propagate host routes for VPN-IPv4 eBGP neighbors to other routers and provider edge routers. Alternatively, you can specify the next-hop-self address when you configure iBGP neighbors.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and places the router in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535. In this instance an eBGP routing process is configured.
Step 4	**no bgp default route-target filter** **Example:** `Router(config-router)# no bgp default route-target filter`	Disables BGP route-target community filtering. All received BGP VPN-IPv4 routes are accepted by the router. Accepting VPN-IPv4 routes is the desired behavior for a router configured as an ASBR.

	Command or Action	Purpose
Step 5	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode. • The **unicast** keyword specifies a unicast prefix. This command configures a routing session to carry VPN-IPv4 addresses across the VPN backbone. Each address is globally unique by the addition of an 8-byte RD.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor 10.0.0.1 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs. The address of the eBGP neighbor or the eBGP peer group is identified to the specified autonomous system.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. These commands activate the advertisement of the VPNv4 address family to a neighboring eBGP router or an eBGP peer group.
Step 8	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits from the address family configuration mode.
Step 9	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

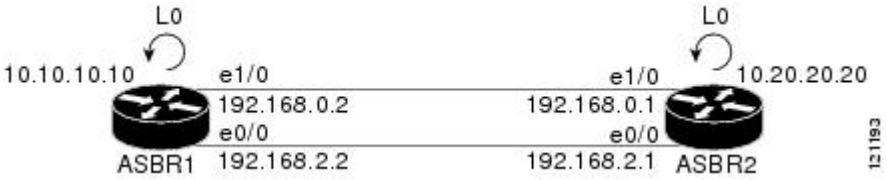
Configuring Peering of the Loopback Interface of Directly Connected ASBRs

This functionality is provided with the release of the MPLS VPN - Interautonomous System Support feature on Cisco IOS Release 12.0(29)S and later releases. An eBGP session configured between loopbacks of directly connected ASBRs allows load sharing between loopback addresses.

Perform the following tasks in this section to configure peering of loopback interfaces of directly connected ASBRs:

The figure below shows the loopback configuration for directly connected ASBR1 and ASBR2 routers. This configuration is used as the example in the tasks that follow.

Figure 9 Loopback Interface Configuration for Directly Connected ASBR1 and ASBR2 Routers



Configuring Loopback Interface Addresses for Directly Connected ASBRs

Perform the following task to configure loopback interface addresses for directly connected ASBRs.

Note Loopback addresses need to be configured for each directly connected ASBR. That is, configure a loopback address for ASBR1 and for ASBR2 (see the figure above).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *interface number*
4. **ip address** *ip-address mask* [**secondary**]
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface number* **Example:** `Router(config)# interface loopback 0`	Configures a software-only virtual interface that emulates an interface that is always up. • The *interface-number* argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces that you can create.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.10.10.10 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Examples

The following example shows the configuration of a loopback address for ASBR1:

```
configure terminal
interface loopback 0
 ip address 10.10.10.10 255.255.255.255
```

The following example shows the configuration of a loopback address for ASBR2:

```
configure terminal
interface loopback 0
 ip address 10.20.20.20 255.255.255.255
```

Configuring Static Routes to the eBGP Neighbor Loopback

Perform the following task to configure /32 static routes to the eBGP neighbor loopback.

A /32 static route is established with the following commands:

```
Router(config)# ip route X.X.X.X 255.255.255.255 Ethernet 1/0 Y.Y.Y.Y
Router(config)# ip route X.X.X.X 255.255.255.255 Ethernet 1/0 Z.Z.Z.Z
```

Where *X.X.X.X* is the neighboring loopback address and Ethernet 1/0 and Ethernet 0/0 are the links connecting the peering routers. *Y.Y.Y.Y* and *Z.Z.Z.Z* are the respective next-hop addresses on the interfaces.

Note You need to configure /32 static routes on each of the directly connected ASBRs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type ip-address interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type ip-address interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Router(config)# ip route 10.20.20.20 255.255.255.255 Ethernet 1/0 192.168.0.1`	Establishes static routes. • The *prefix* argument is the IP route prefix for the destination. • The *mask* argument is the prefix mask for the destination. • The *ip-address* argument is the IP address of the next hop that you can use to reach the specified network. • The *interface-type* and *interface-number* arguments are the network interface type and interface number. • The *distance* argument is an administrative distance. • The *name* argument applies a name to the specified route. • The **permanent** keyword specifies that the route is not to be removed, even if the interface shuts down. • The **tag** *tag* keyword-argument pair names a tag value that can be used as a "match" value for controlling redistribution through the use of route maps.
Step 4	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Examples

The following example shows the configuration of a /32 static route from the ASBR1 router to the loopback address of the ASBR2 router:

```
configure terminal
ip route 10.20.20.20 255.255.255.255 e1/0 192.168.0.1
ip route 10.20.20.20 255.255.255.255 e0/0 192.168.2.1
```

The following example shows the configuration of a /32 static route from the ASBR2 router to the loopback address of the ASBR1 router:

```
configure terminal
ip route vrf vpn1 10.10.10.10 255.255.255.255 Ethernet 1/0 192.168.0.2
ip route vrf vpn1 10.10.10.10 255.255.255.255 Ethernet 0/0 192.168.2.2
```

Configuring Forwarding on the Directly Connected Interfaces

Perform this task to configure forwarding on the directly connected interfaces.

This task is required for sessions between loopbacks. In the Configuring Static Routes to the eBGP Neighbor Loopback, page 17 task, Ethernet 1/0 and Ethernet 0/0 are the connecting interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type slot/port*
4. **ip address** *ip-address mask* [**secondary**]
5. **mpls bgp forwarding**
6. **exit**
7. Repeat Steps 3, 4, and 5 for another connecting interface (Ethernet 0/0).
8. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**interface** *interface-type slot/port* **Example:** `Router(config)# interface ethernet 1/0`	Configures an interface type and enters interface configuration mode. • The *interface-type* argument is the type of interface to be configured. • The *slot* argument is the slot number. Refer to the appropriate hardware manual for slot and port information. • The */port* keyword and argument are the port number. Refer to the appropriate hardware manual for slot and port information.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 192.168.0.2 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 5	**mpls bgp forwarding** **Example:** `Router(config-if)# mpls bgp forwarding`	Configures BGP to enable MPLS forwarding on connecting interfaces.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits to global configuration mode.
Step 7	Repeat Steps 3, 4, and 5 for another connecting interface (Ethernet 0/0).	
Step 8	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Examples

The following example shows the configuration of BGP MPLS forwarding on the interfaces connecting the ASBR1 router with the ASBR2 router:

```
configure terminal
interface ethernet 1/0
 ip address 192.168.0.2 255.255.255.0
 mpls bgp forwarding
 exit
!
interface ethernet 0/0
 ip address 192.168.2.2 255.255.255.0
```

```
 mpls bgp forwarding
 exit
```

The following example shows the configuration of BGP MPLS forwarding on the interfaces connecting the ASBR2 router with the ASBR1 router:

```
configure terminal
interface ethernet 1/0
 ip address 192.168.0.1 255.255.255.0
 mpls bgp forwarding
 exit
!
interface ethernet 0/0
 ip address 192.168.2.1 255.255.255.0
 mpls bgp forwarding
 exit
```

Configuring an eBGP Session Between the Loopbacks

Perform the following tasks to configure an eBGP session between the loopbacks.

Note You need to configure an EGBP session between loopbacks on each directly connected ASBR.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default route-target filter**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check**
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
8. **address-family vpnv4** [**unicast**]
9. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
10. **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
11. **end**
12. **show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**]

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 200`	Configures the BGP routing process. • The *as-number* indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along.
Step 4	**no bgp default route-target filter** **Example:** `Router(config-router)# no bgp default route-target filter`	Disables BGP route-target filtering. All received BGP VPN-IPv4 routes are accepted by the router.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.20.20.20 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group. • The *as-number* argument is the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check** **Example:** `Router(config-router)# neighbor 10.20.20.20 disable-connected-check`	Allows peering between loopbacks. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group.

	Command or Action	Purpose
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Router(config-router)# neighbor 10.20.20.20 update-source loopback 0`	Allows BGP sessions in Cisco IOS releases to use any operational interface for TCP connections. • The *ip-address* argument is the IPv4 address of the BGP-speaking neighbor. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons. • The *peer-group-name* argument is the name of a BGP peer group. • The *interface-type* argument is the interface type. • The *interface-number* argument is the interface number.
Step 8	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing protocols such as BGP, Routing Information Protocol (RIP), and static routing. • The **vpnv4** keyword configures sessions that carry customer VPN-IPv4 prefixes, each of which has been made globally unique by the addition of an 8-byte route distinguisher. • The **unicast** keyword specifies unicast prefixes.
Step 9	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Router(config-router-af)# neighbor 10.20.20.20 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring router. • The *peer-group-name* argument is the name of a BGP peer group. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.
Step 10	**neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **Example:** `Router(config-router-af)# neighbor 10.20.20.20 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring router. • The *peer-group-name*argument is the name of a BGP peer group. • The **both** keyword specifies that both standard and extended communities will be sent. • The **standard** keyword specifies that only standard communities will be sent. • The **extended** keyword specifies that only extended communities will be sent.
Step 11	**end** **Example:** `Router(config-router-af)# end`	Exits to privileged EXEC mode.

Command or Action	Purpose
Step 12 **show mpls forwarding-table** [*network* {*mask* | *length*} | **labels** *label* [*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**detail**] **Example:** `Router# show mpls forwarding-table`	Displays the contents of the MPLS LFIB. Use this command to verify that load balancing occurs between loopbacks. You need to ensure that the MPLS LFIB entry for the neighbor route lists the available paths and interfaces.

Examples

The following example shows the configuration for VPNv4 sessions on the ASBR1 router:

```
configure terminal
router bgp 200
 bgp log-neighbor-changes
 neighbor 10.20.20.20 remote-as 100
 neighbor 10.20.20.20 disable-connected-check
 neighbor 10.20.20.20 update-source loopback 0
!
 address-family vpnv4
 neighbor 10.20.20.20 activate
 neighbor 10.20.20.20 send-community extended
 end
```

The following example shows the configuration for VPNv4 sessions on the ASBR2:

```
configure terminal
router bgp 100
 bgp log-neighbor-changes
 neighbor 10.10.10.10 remote-as 200
 neighbor 10.10.10.10 disable-connected-check
 neighbor 10.10.10.10 update-source Loopback 0
!
 address-family vpnv4
 neighbor 10.10.10.10 activate
 neighbor 10.10.10.10 send-community extended
 end
```

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

Perform this task to configure eBGP routing to exchange MPLS VPN routes between subautonomous systems in a confederation.

Note To ensure that the host routes for VPN-IPv4 eBGP neighbors are propagated (by means of the IGP) to the other routers and provider edge routers, specify the **redistribute connected** command in the IGP configuration portion of the CeBGP router. If you are using OSPF, make sure that the OSPF process is not enabled on the CeBGP interface where the “redistribute connected” subnet exists.

Note In this confederation, subautonomous system IGP domains must know the addresses of CeBGP-1 and CeBGP-2. If you do not specify a next-hop-self address as part of the router configuration, ensure that the addresses of all PE routers in the subautonomous system are distributed throughout the network, not just the addresses of CeBGP-1 and CeBGP-2.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *sub-autonomous-system*
4. **bgp confederation identifier** *as-number*
5. **bgp confederation peers** *sub-autonomous-system*
6. **no bgp default route-target filter**
7. **address-family vpnv4** [**unicast**]
8. **neighbor** *peer-group-name* **remote-as** *as-number*
9. **neighbor** *peer-group-name* **next-hop-self**
10. **neighbor** *peer-group-name* **activate**
11. **exit-address-family**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *sub-autonomous-system* **Example:** `Router(config)# router bgp 2`	Enters router configuration mode, creates an eBGP routing process, and assigns it an autonomous system number. The subautonomous system number is passed along to identify the router to eBGP routers in other subautonomous systems.

	Command or Action	Purpose
Step 4	**bgp confederation identifier** *as-number* **Example:** `Router(config-router)# bgp confederation identifier 100`	Defines an eBGP confederation by specifying a confederation identifier associated with each subautonomous system. The subautonomous systems appear as a single autonomous system.
Step 5	**bgp confederation peers** *sub-autonomous-system* **Example:** `Router(config-router)# bgp confederation peers 1`	Specifies the subautonomous systems that belong to the confederation (identifies neighbors of other subautonomous systems within the confederation as special eBGP peers).
Step 6	**no bgp default route-target filter** **Example:** `Router(config-router)# no bgp default route-target filter`	Disables BGP route-target community filtering. All received BGP VPN-IPv4 routes are accepted by the router.
Step 7	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode and configures a routing session to carry VPNv4 addresses across the VPN backbone. Each address has been made globally unique by the addition of an 8-byte route distinguisher (RD). • The **unicast** keyword specifies a unicast prefix.
Step 8	**neighbor** *peer-group-name* **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor R remote-as 1`	Specifies a neighboring eBGP peer group. This eBGP peer group is identified to the specified subautonomous system.
Step 9	**neighbor** *peer-group-name* **next-hop-self** **Example:** `Router(config-router-af)# neighbor R next-hop-self`	Advertises the router as the next hop for the specified neighbor. If you specify a next-hop-self address as part of the router configuration, you do not need to use the **redistribute connected** command.
Step 10	**neighbor** *peer-group-name* **activate** **Example:** `Router(config-router-af)# neighbor R activate`	Activates the advertisement of the VPNv4 address family to a neighboring PE router in the specified subautonomous system.

	Command or Action	Purpose
Step 11	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 12	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Verifying Inter-AS for ASBRs Exchanging MPLS VPN-IPv4 Addresses

Perform this task to verify that Inter-AS for ASBRs Exchanging MPLS VPN-IPv4 addresses operates as you expected.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4 all**
3. **show ip bgp vpnv4 all labels**
4. **show mpls forwarding-table**
5. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if required. For example:

Example:

```
Router> enable
Router#
```

Step 2 **show ip bgp vpnv4 all**

Use this command to verify that all VPNv4 information in the BGP table on the ASBR is as you expected. For example:

Example:

```
Router# show ip bgp vpnv4 all

BGP table version is 99, local router ID is 172.16.10.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin coeds: i - IGP, e - EGP, ? incomplete
```

```
  Network          Next Hop       Metric  LocPrf  Weight Path
Route Distinguisher 100:1
*> 10.1.1.0/24     10.1.1.1           50     100       0 200 ?
* i                 10.1.1.5          100     100       0 200 ?
Route Distinguisher 100:2
* 192.168.1.0/24   10.1.1.1          100     100       0 200 ?
*>i                10.1.1.5           50     100       0 200 ?
* 172.16.1.0/24     10.1.1.1          100     100       0 200 ?
+>i                10.1.1.5           50     100       0 200 ?
Route Distinguisher 200:1
*>i172.16.1.0/24    10.1.1.2           50     100       0 200 ?
*> 10.2.1.0/24      0.0.0.0.           0           32768 ?
Route Distinguisher 200:2
*>i172.16.1.0/24   10.1.1.5           50     100       0 200 ?
*>i172.16.1.0/24   10.1.1.5           50     100       0 200 ?
*> 10.2.1.0/24      0.0.0.0            0           32768 ?
```

Step 3 **show ip bgp vpnv4 all labels**

Use this command to display information about all VPNv4 labels. For example:

Example:

```
Router# show ip bgp vpnv4 all labels
      Network          Next Hop        In label/Out label
Route Distinguisher 100:1
      10.1.1.0/24      172.16.10.3       20/29
Route Distinguisher 100:2
      10.1.1.0/24      172.16.10.3       21/35
      10.2.1.0/24      172.16.10.3       24/36
Route Distinguisher 200:1
      10.30.1.0/24       10.1.1.2        23/164
Route Distinguisher 200:2
      10.31.1.0/24       10.1.1.2        27/165
```

Step 4 **show mpls forwarding-table**

Use this command to display the contents of the MPLS LFIB (such as VPNv4 prefix/length and BGP next-hop destination for the route) and see how the VPN-IPv4 LFIB entries appear. For example:

Example:

```
Router# show mpls forwarding-table
Local Outgoing      Prefix            Bytes tag Outgoing       Next Hop
tag   tag or VC     or Tunnel Id      switched  interface
33    33            10.120.4.0/24     0         Hs0/0          point2point
35    27            100:12:10.200.0.1/32 \
                                      0         Hs0/0          point2point
```

In this example, the Prefix field appears as a VPN-IPv4 RD, plus the prefix. If the value is longer than the width of the Prefix column (as illustrated in the last line of the example), the output automatically wraps onto the next line in the forwarding table, preserving column alignment.

Step 5 **exit**

Use this command to exit to user EXEC mode. For example:

Example:

```
Router# exit
Router>
```

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

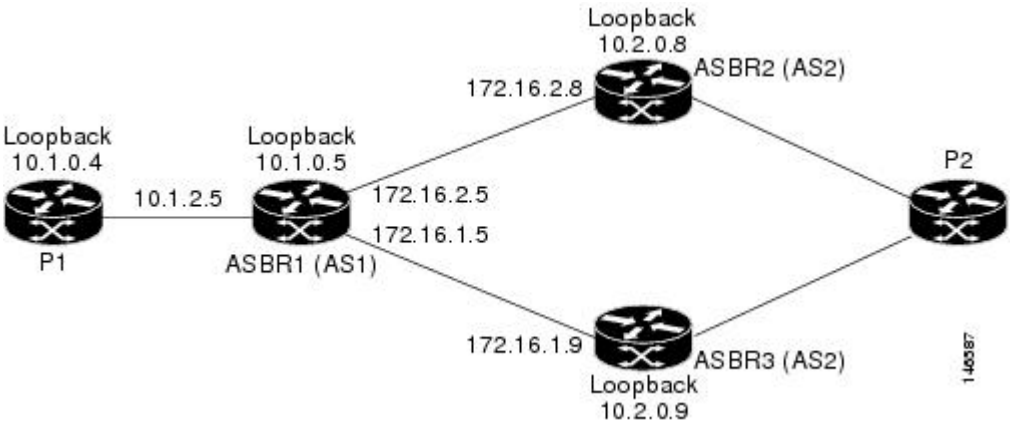
Perform this task to configure eBGP multipath load sharing for MPLS VPN Inter-AS ASBRs exchanging VPN-IPv4 routes. This allows for more efficient use of the LSPs in an interautonomous system network because you can set up the load sharing of traffic among the different multipaths and the best path to reach the destination.

Note The following restrictions apply to configuring multipath load sharing for MPLS VPN Inter-AS ASBRS exchanging VPN-IPv4 routes:

- Per packet load balancing is not supported for this feature. Load balancing for this features works on the IP source and destination hash or on the bottom label in the label stack, depending on the platform and depth of the MPLS label stack.
- If MPLS scalability is an issue for you, we recommend that you do not enable VPNv4 multipath on ASBRs.

The figure below shows an eBGP multipath configuration for three VPN-IPv4 ASBRs. The links from ASBR1 to ASBR2 and ASBR3 have an eBGP VPN-IPv4 session configured. In the figure below, eBGP multipath load sharing is configured on ASBR1. You configure the number of sessions from ASBR1 to ASBR2 and ASBR3 with the **maximum-paths** command in address family configuration mode.

Figure 10 eBGP Multipath Configuration for Three VPN-IPv4 ASBRs



The configurations in the figure above is used as an example for this task and for the task in the Verifying eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs, page 34.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default route-target filter**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type interface-number*
7. **neighbor** {*ip-address* | *peer-group-name*} **next-hop-self**
8. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
9. Repeat Step 8 for each BGP neighbor.
10. **address-family vpnv4** [**unicast**]
11. **neighbor** {*ip-address* | *peer-group-name*} **activate**
12. **neighbor** {*ip-address* | *peer-group-name*} **next-hop-self**
13. **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
14. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
15. **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
16. Repeat Steps 14 and 15 for each BGP neighbor.
17. **maximum-paths** *number-paths*
18. **exit-address-family**
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 1`	Configures an eBGP routing process and places the router in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.

	Command or Action	Purpose
Step 4	**no bgp default route-target filter** **Example:** `Router(config-router)# no bgp default route-target filter`	Disables BGP route-target community filtering. All received VPN-IPv4 routes are accepted by the configured router. Accepting VPN-IPv4 routes is the desired behavior for a router configured as an ASBR.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.1.0.4 remote-as 1`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Router(config-router)# neighbor 10.1.0.4 update-source loopback 0`	Allows BGP sessions to use any operational interface for TCP connections. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *interface-typeinterface-number* arguments specify the type and number for the operational interface. This example shows how to set up BGP TCP connections for the specified neighbor with the IP address of the loopback interface rather than the best local address.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self** **Example:** `Router(config-router)# neighbor 10.1.0.4 next-hop-self`	Configures the router as the next hop for a BGP neighbor or peer group. • The *ip-address* argument specifies the IP address of the BGP neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 172.16.1.9 remote-as 2`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 9	Repeat Step 8 for each BGP neighbor.	—

Command or Action	Purpose
Step 10 **address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode. • The **unicast** keyword specifies a unicast prefix. This command configures a routing session to carry VPN-IPv4 addresses across the VPN backbone. Each address is globally unique by the addition of an 8-byte RD.
Step 11 **neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.1.0.4 activate`	Enables the exchange of information with a neighboring router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 12 **neighbor** {*ip-address* | *peer-group-name*} **next-hop-self** **Example:** `Router(config-router-af)# neighbor 10.1.0.4 next-hop-self`	Configures the router as the next hop for a BGP neighbor or peer group. • The *ip-address* argument specifies the IP address of the BGP neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 13 **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **Example:** `Router(config-router-af)# neighbor 10.1.0.4 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring router. • The *peer-group-name* argument is the name of a BGP peer group. • The **both** keyword specifies that both standard and extended communities will be sent. • The **standard** keyword specifies that only standard communities will be sent. • The **extended** keyword specifies that only extended communities will be sent.
Step 14 **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Router(config-router-af)# neighbor 172.16.1.9 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring router. • The *peer-group-name* argument is the name of a BGP peer group. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.

	Command or Action	Purpose
Step 15	**neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **Example:** `Router(config-router-af)# neighbor 172.16.1.9 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring router. • The *peer-group-name* argument is the name of a BGP peer group. • The **both** keyword specifies that both standard and extended communities will be sent. • The **standard** keyword specifies that only standard communities will be sent. • The **extended** keyword specifies that only extended communities will be sent.
Step 16	Repeat Steps 14 and 15 for each BGP neighbor.	
Step 17	**maximum-paths** *number-paths* **Example:** `Router(config-router-af)# maximum-paths 2`	Configures the maximum number of parallel routes that an IP routing protocol will install into the routing table. • The *number-paths* argument specifies the number of routes to install to the routing table. • See the Load Sharing with MPLS VPN Inter-AS ASBRs, page 11 for information on the number of parallel routes allowed by a specific Cisco IOS release.
Step 18	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits from address family configuration mode.
Step 19	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

- Examples, page 33

Examples

The following example shows the configuration for eBGP multipath for VPNv4 sessions on the ASBR1 router:

```
configure terminal
router bgp 1
 no bgp default route-target filter
 neighbor 10.1.0.4 remote-as 1
 neighbor 10.1.0.4 update-source Loopback 0
 neighbor 10.1.0.4 next-hop-self
 neighbor 172.16.1.9 remote-as 2
 neighbor 172.16.2.8 remote-as 2
```

```
!
 address-family vpnv4
 neighbor 10.1.0.4 activate
 neighbor 10.1.0.4 next-hop-self
 neighbor 10.1.0.4 send-community extended
 neighbor 172.16.1.9 activate
 neighbor 172.16.1.9 send-community extended
 neighbor 172.16.2.8 activate
 neighbor 172.16.2.8 send-community extended
 maximum-paths 2
 exit-address-family
 end
```

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

Perform the following task to verify that eBGP multipath load sharing for MPLS VPN Inter-AS ASBRs is operating as you expect.

The configurations in the figure above are used as an example for the task that follows.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4 all** [**summary**]
3. **show ip bgp vpnv4 all**
4. **show ip bgp vpnv4** [*network*]
5. **show mpls forwarding-table**
6. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if required. For example:

Example:

```
Router> enable
Router#
```

Step 2 **show ip bgp vpnv4 all** [**summary**]

Use this command to verify that all peers are up. for example:

Example:

```
Router# show ip bgp vpnv4 all summary
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.1.0.4        4     1      87      86        5    0    0 01:24:56        2
172.16.1.9      4     2      88      88        5    0    0 01:25:49        2
172.16.2.8      4     2      88      88        5    0    0 01:25:49        2
```

The output shows that all peers expected to be up are up and sending and receiving messages.

Step 3 **show ip bgp vpnv4 all**

Use this command to verify that BGP has paths from both remote ASBRs. For example:

Example:

```
Router# show ip bgp vpnv4 all
   Network          Next Hop            Metric LocPrf Weight Path
.
.
Route Distinguisher: 1:105
*>i192.168.0.1/32   10.1.0.3                11    100      0 ?
*> 192.168.0.2/32   172.16.2.8                             0 2 ?
*                   172.16.1.9                             0 2 ?
*>i192.168.1.0      10.1.0.3                 0    100      0 ?
*> 192.168.2.0      172.16.2.8                             0 2 ?
*                   172.16.1.9                             0 2 ?
```

The bold entries in the output confirm that BGP has a path to ASBR2 (172.16.2.8) and to ASBR3 (172.16.1.9).

Step 4 **show ip bgp vpnv4** [*network*]

Use this command to verify that paths are marked as multipath. For example:

Example:

```
Router# show ip bgp vpnv4 192.168.2.0
BGP routing table entry for 1:105:192.168.2.0/24, version 3
Paths: (2 available, best #1, no table)
  Advertised to update-groups:
     2          3
  2
    172.16.2.8 from 172.16.2.8 (10.2.0.8)
      Origin incomplete, localpref 100, valid, external, multipath
, best
      Extended Community: RT:1:100 OSPF DOMAIN ID:0x0005:0x0000000A0200
        OSPF RT:0.0.0.0:2:0 OSPF ROUTER ID:192.168.2.2:512,
      mpls labels in/out 21/25
  2
    172.16.1.9 from 172.16.1.9 (10.2.0.9)
      Origin incomplete, localpref 100, valid, external, multipath
      Extended Community: RT:1:100 OSPF DOMAIN ID:0x0005:0x0000000A0200
        OSPF RT:0.0.0.0:2:0 OSPF ROUTER ID:192.168.2.2:512,
      mpls labels in/out 21/25
```

In the output, the “multipath” and “mpls labels in/out 21/25” are in bold text for example purposes only.

Step 5 **show mpls forwarding-table**

Use this command to verify that MPLS forwarding is properly set up and counters are increasing when traffic is present. For example:

Example:

```
Router# show mpls forwarding-table
Local  Outgoing      Prefix            Bytes Label   Outgoing   Next Hop
Label  Label or VC   or Tunnel Id      Switched      interface
.
.
16     Pop Label     172.16.1.9/32     0             Et1/0      172.16.1.9
17     Pop Label     172.16.2.8/32     0             Et2/0      172.16.2.8
18     Pop Label     10.1.1.0/24       0             Et0/0      10.1.2.4
19     16            10.1.0.3/32       0             Et0/0      10.1.2.4
20     Pop Label     10.1.0.4/32       0             Et0/0      10.1.2.4
21     25            1:105:192.168.2.0/24   \
                                       26658         Et1/0      172.16.1.9
       25            1:105:192.168.2.0/24   \
                                       1180          Et2/0      172.16.2.8
22     24            1:105:192.168.0.2/32   \
                                       15740         Et1/0      172.16.1.9
       24            1:105:192.168.0.2/32   \
                                       0             Et2/0      172.16.2.8
```

```
23      19              1:105:192.168.0.1/32    \
                                        15638           Et0/0       10.1.2.4
24      20              1:105:192.168.1.0/24    \
                                        32740           Et0/0       10.1.2.4
```

Step 6 **exit**

Use this command to exit to user EXEC mode. For example:

Example:

```
Router# exit
Router>
```

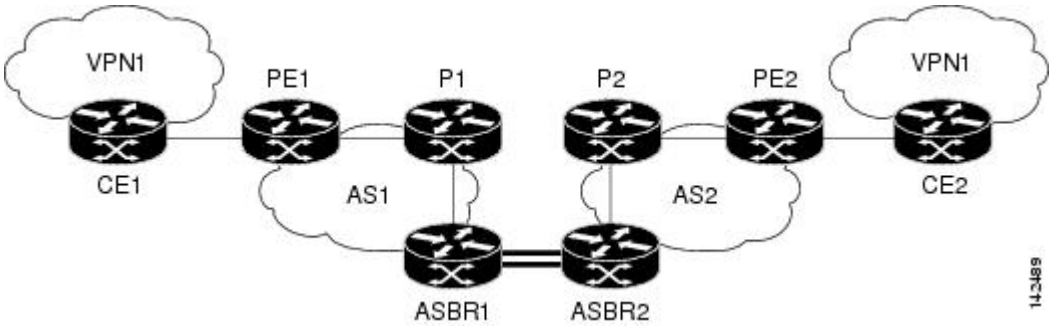
Configuration Examples for MPLS VPN - Interautonomous System Support

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

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

- Autonomous system 1 (AS1) contains PE1, P1, ASBR1. The IGP is OSPF.
- Autonomous system 2 (AS2) contains PE2, P2, ASBR2. The IGP is IS-IS.
- CE1 and CE2 belong to the same VPN, which is called VPN1.
- The P routers are route reflectors.
- ASBR1 is configured with the **redistribute connected subnets** command.
- ASBR2 is configured with the **neighbor next-hop-self** command.

Figure 11 Configuring Two Autonomous Systems



Configuration for Autonomous System 1 CE1 Example for Two Autonomous Systems

The following example shows how to configure the CE1 router in VPN1 in a topology with two autonomous systems (see the figure above):

```
!
hostname CE1
!
interface Loopback 1
 ip address 192.168.0.1 255.255.255.255
!
interface Ethernet 1/0
 description Link to PE1
 ip address 192.168.1.1 255.255.255.0
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
 end
```

Configuration for Autonomous System 1 PE1 Example for Two Autonomous Systems

The following example shows how to configure the PE1 router in autonomous system 1 in a topology with two autonomous systems (see the figure above):

```
!
hostname PE1
!
ip cef
!
ip vrf VPN1
 rd 1:105
 route-target export 1:100
 route-target import 1:100
!
interface Loopback 0
 ip address 10.1.0.3 255.255.255.255
!
interface Ethernet 0/0
 description Link to CE1
 ip vrf forwarding VPN1
 ip address 192.168.1.2 255.255.255.0
!
interface Ethernet 1/0
 description Link to P1
 ip address 10.1.1.3 255.255.255.0
 mpls ip
!
router ospf 10 vrf VPN1
 log-adjacency-changes
 redistribute bgp 1 metric 100 subnets
 network 192.168.0.0 0.0.255.255 area 0
!
router ospf 1
```

```
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 1
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor 10.1.0.4 peer-group R
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor 10.1.0.4 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
 redistribute ospf 10 vrf VPN1
 no auto-summary
 no synchronization
 exit-address-family
!
end
```

Configuration for Autonomous System 1 P1 Example for Two Autonomous Systems

The following example shows how to configure the P1 router in autonomous system 1 in a topology with two autonomous systems (see the figure above):

```
!
hostname P1
!
ip cef
!
interface Loopback 0
 ip address 10.1.0.4 255.255.255.255
!
interface Ethernet 0/0
 description Link to PE1
 ip address 10.1.1.4 255.255.255.0
 mpls ip
!
interface Ethernet 1/0
 description Link to ASBR1
 ip address 10.1.2.4 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
no synchronization
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 1
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor R route-reflector-client
 neighbor 10.1.0.3 peer-group R
 neighbor 10.1.0.5 peer-group R
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor R route-reflector-client
 neighbor 10.1.0.3 activate
 neighbor 10.1.0.5 activate
 exit-address-family
```

```
!
end
```

Configuration for Autonomous System 1 ASBR1 Example for Two Autonomous Systems

The following example shows how to configure ASBR1 in autonomous system 1 in a topology with two autonomous systems (see the figure above):

```
hostname ASBR1
!
ip cef
!
interface Loopback 0
 ip address 10.1.0.5 255.255.255.255
!
interface Ethernet 0/0
 description Link to P1
 ip address 10.1.2.5 255.255.255.0
 mpls ip
!
interface Ethernet 1/0
 description Link to ASBR2
 ip address 172.16.0.1 255.255.255.255
 mpls bgp forwarding
!
router ospf 1
 log-adjacency-changes
 redistribute connected subnets
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 1
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor 10.1.0.4 peer-group R
 neighbor 172.16.0.2 remote-as 2
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor R next-hop-self
 neighbor 10.1.0.4 activate
 neighbor 172.16.0.2 activate
 neighbor 172.16.0.2 send-community extended
 exit-address-family
!
end
```

Configuration for Autonomous System 2 ASBR2 Example for Two Autonomous Systems

The following example shows how to configure ASBR2 in autonomous system 2 in a topology with two autonomous systems (see the figure above):

```
!
hostname ASBR2
!
ip cef
!
interface Loopback 0
 ip address 10.2.0.8 255.255.255.255
 ip router isis
!
interface Ethernet 0/0
 description Link to ASBR1
```

```
 ip address 172.16.0.2 255.255.255.255
 mpls bgp forwarding
!
interface Serial 2/0
 description Link to P2
 ip address 10.2.2.8 255.255.255.0
 ip router isis
 mpls ip
 no fair-queue
 serial restart-delay 0
!
router isis
 net 49.0002.0000.0000.0003.00
!
router bgp 2
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor 10.2.0.7 remote-as 2
 neighbor 10.2.0.7 update-source Loopback 0
 neighbor 10.2.0.7 next-hop-self
 neighbor 172.16.0.1 remote-as 1
 no auto-summary
!
 address-family vpnv4
 neighbor 10.2.0.7 activate
 neighbor 10.2.0.7 send-community extended
 neighbor 10.2.0.7 next-hop-self
 neighbor 172.16.0.1 activate
 neighbor 172.16.0.1 send-community extended
 exit-address-family
!
end
```

Configuration for Autonomous System 2 P2 Example for Two Autonomous Systems

The following example shows how to configure the P2 router in autonomous system 2 in a topology with two autonomous systems (see the figure above):

```
!
hostname P2
!
ip cef
!
interface Loopback 0
 ip address 10.2.0.7 255.255.255.255
 ip router isis
!
interface Ethernet 1/0
 description Link to PE2
 ip address 10.2.1.7 255.255.255.0
 ip router isis
 mpls ip
!
interface Serial 2/0
 description Link to ASBR2
 ip address 10.2.2.7 255.255.255.0
 ip router isis
 mpls ip
 no fair-queue
 serial restart-delay 0
!
router isis
 net 49.0002.0000.0000.0008.00
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 2
 no neighbor R transport path-mtu-discovery
```

```
 neighbor R update-source Loopback 0
 neighbor R route-reflector-client
 neighbor 10.2.0.6 peer-group R
 neighbor 10.2.0.8 peer-group R
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor R route-reflector-client
 neighbor 10.2.0.6 activate
 neighbor 10.2.0.8 activate
 exit-address-family
!
end
```

Configuration for Autonomous System 2 PE2 Example for Two Autonomous Systems

The following example shows how to configure the PE2 router in autonomous system 2 in a topology with two autonomous systems (see the figure above):

```
!
hostname PE2
!
ip cef
!
ip vrf VPN1
 rd 1:105
 route-target export 1:100
 route-target import 1:100
!
interface Loopback 0
 ip address 10.2.0.6 255.255.255.255
 ip router isis
!
interface Ethernet 0/0
 description Link to P2
 ip address 10.2.1.6 255.255.255.0
 ip router isis
 mpls ip
!
interface Serial 2/0
 description Link to CE2
 ip vrf forwarding VPN1
 ip address 192.168.2.2 255.255.255.0
 no fair-queue
 serial restart-delay 0
!
router ospf 10 vrf VPN1
 log-adjacency-changes
 redistribute bgp 2 subnets
 network 192.168.0.0 0.0.255.255 area 0
!
router isis
 net 49.0002.0000.0000.0009.00
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.2.0.7 remote-as 2
 neighbor 10.2.0.7 update-source Loopback 0
 no auto-summary
 !
address-family vpnv4
 neighbor 10.2.0.7 activate
 neighbor 10.2.0.7 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf VPN1
 redistribute connected
 redistribute ospf 10 vrf VPN1
 no auto-summary
```

```
 no synchronization
 exit-address-family
!
end
```

Configuration for Autonomous System 2 CE2 Example for Two Autonomous Systems

The following example shows how to configure the CE2 router in autonomous system 2 in a topology with two autonomous systems (see the figure above):

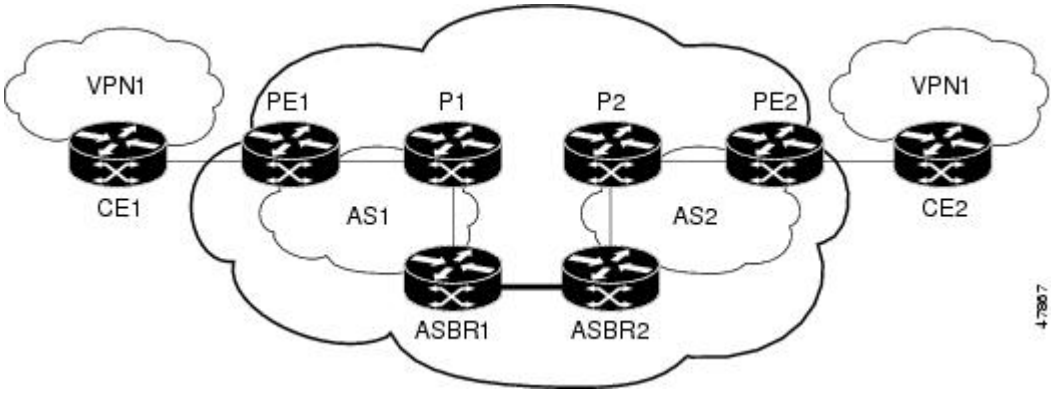
```
!
hostname CE2
!
interface Loopback 0
 ip address 192.168.0.2 255.255.255.255
!
interface Serial 2/0
 description Link to PE2
 ip address 192.168.2.1 255.255.255.0
 no fair-queue
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
end
```

Configuring Inter-AS with ASBRs in a Confederation Example

The network topology in the figure below shows a single Internet service provider (ISP), which is partitioning the backbone with confederations. The autonomous system number of the provider is 100. The two autonomous systems run their own IGPs and are configured as follows:

- Autonomous system 1 (AS1) contains PE1, P1, ASBR1. The IGP is OSPF.
- Autonomous system 2 (AS2) contains PE2, P2, ASBR2. The IGP is IS-IS.
- CE1 and CE2 belong to the same VPN, which is called VPN1.
- The P routers are route reflectors.
- ASBR1 is configured with the **redistribute connected subnets** command.
- ASBR2 is configured with the **neighbor next-hop-self** command.

Figure 12 Configuring Two Autonomous Systems in a Confederation



- Inter-AS Confederation Configuration for Autonomous System 1 CE1 Example, page 43

Inter-AS Confederation Configuration for Autonomous System 1 CE1 Example

The following example shows how to configure CE1 in VPN1 in an Inter-AS confederation (see the figure above):

```
!
hostname CE1
!
interface Loopback 1
 ip address 192.168.0.1 255.255.255.255
!
interface Ethernet 1/0
 description Link to PE1
 ip address 192.168.1.1 255.255.255.0
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
end
```

Inter-AS Confederation Configuration for Autonomous System 1 PE1 Example

The following example shows how to configure PE1 in autonomous system 1 in an Inter-AS confederation (see the figure above):

```
hostname PE1
!
ip cef
!
ip vrf VPN1
 rd 1:105
 route-target export 1:100
 route-target import 1:100
!
interface Loopback 0
 ip address 10.1.0.3 255.255.255.255
!
interface Ethernet 0/0
 description Link to CE1
 ip vrf forwarding VPN1
 ip address 192.168.1.2 255.255.255.0
!
interface Ethernet 1/0
 description Link to P1
 ip address 10.1.1.3 255.255.255.0
 mpls ip
!
router ospf 10 vrf VPN1
 log-adjacency-changes
 redistribute bgp 1 metric 100 subnets
 network 192.168.0.0 0.0.255.255 area 0
!
router ospf 1
 log-adjacency-changes
```

```
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 100
 neighbor R peer-group
 neighbor R remote-as 1
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor 10.1.0.4 peer-group R
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor 10.1.0.4 activate
 exit-address-family
 !
 address-family ipv4 vrf VPN1
 redistribute ospf 10 vrf VPN1
 no auto-summary
 no synchronization
 exit-address-family
!
end
```

Inter-AS Confederation Configuration for Autonomous System 1 P1 Example

The following example shows how to configure P1 in autonomous system 1 in a confederation topology (see the figure above):

```
!
hostname P1
!
ip cef
!
interface Loopback 0
 ip address 10.1.0.4 255.255.255.255
!
interface Ethernet 0/0
 description Link to PE1
 ip address 10.1.1.4 255.255.255.0
 mpls ip
!
interface Ethernet 1/0
 description Link to ASBR1
 ip address 10.1.2.4 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 100
 neighbor R peer-group
 neighbor R remote-as 1
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor R route-reflector-client
 neighbor 10.1.0.3 peer-group R
 neighbor 10.1.0.5 peer-group R
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor R route-reflector-client
 neighbor 10.1.0.3 activate
 neighbor 10.1.0.5 activate
```

```
 exit-address-family
!
end
```

Inter-AS Confederation Configuration for Autonomous System 1 ASBR1 Example

The following example shows how to configure ASBR1 in autonomous system 1 in a confederation topology (see the figure above):

```
!
hostname ASBR1
!
ip cef
!
interface Loopback 0
 ip address 10.1.0.5 255.255.255.255
!
interface Ethernet 0/0
 description Link to P1
 ip address 10.1.2.5 255.255.255.0
 mpls ip
!
interface Ethernet 1/0
 description Link to ASBR2
 ip address 172.16.0.1 255.255.255.255
 mpls bgp forwarding
!
router ospf 1
 log-adjacency-changes
 redistribute connected subnets
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 bgp confederation identifier 100
 bgp confederation peers 2
 neighbor R peer-group
 neighbor R remote-as 1
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor 10.1.0.4 peer-group R
 neighbor 172.16.0.2 remote-as 2
 neighbor 172.16.0.2 next-hop-self
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor R next-hop-self
 neighbor 10.1.0.4 activate
 neighbor 172.16.0.2 activate
 neighbor 172.16.0.2 send-community extended
 neighbor 172.16.0.2 next-hop-self
 exit-address-family
!
end
```

Inter-AS Confederation Configuration for Autonomous System 2 ASBR2 Example

The following example shows how to configure ASBR2 in autonomous system 2 in a confederation topology (see the figure above):

```
!
hostname ASBR2
!
ip cef
!
```

```
interface Loopback 0
 ip address 10.2.0.8 255.255.255.255
 ip router isis
!
interface Ethernet 0/0
 description Link to ASBR1
 ip address 172.16.0.2 255.255.255.255
 mpls bgp forwarding
!
interface Serial 2/0
 description Link to P2
 ip address 10.2.2.8 255.255.255.0
 ip router isis
 mpls ip
no fair-queue
 serial restart-delay 0
!
router isis
 net 49.0002.0000.0000.0003.00
!
router bgp 2
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 bgp confederation identifier 100
 bgp confederation peers 1
 neighbor 10.2.0.7 remote-as 2
 neighbor 10.2.0.7 update-source Loopback 0
 neighbor 10.2.0.7 next-hop-self
 neighbor 172.16.0.1 remote-as 1
 neighbor 172.16.0.1 next-hop-self
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.2.0.7 activate
 neighbor 10.2.0.7 send-community extended
 neighbor 10.2.0.7 next-hop-self
 neighbor 172.16.0.1 activate
 neighbor 172.16.0.1 send-community extended
 neighbor 172.16.0.1 next-hop-self
 exit-address-family
!
end
```

Inter-AS Confederation Configuration for Autonomous System 2 P2 Example

The following example shows how to configure P2 in autonomous system 2 in a confederation topology (see the figure above):

```
!
hostname P2
!
ip cef
!
interface Loopback 0
 ip address 10.2.0.7 255.255.255.255
 ip router isis
!
interface Ethernet 1/0
 description Link to PE2
 ip address 10.2.1.7 255.255.255.0
 ip router isis
 mpls ip
!
interface Serial 2/0
 description Link to ASBR2
 ip address 10.2.2.7 255.255.255.0
 ip router isis
 mpls ip
 no fair-queue
 serial restart-delay 0
```

```
!
router isis
 net 49.0002.0000.0000.0008.00
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 100
 neighbor R peer-group
 neighbor R remote-as 2
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor R route-reflector-client
 neighbor 10.2.0.6 peer-group R
 neighbor 10.2.0.8 peer-group R
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor R route-reflector-client
 neighbor 10.2.0.6 activate
 neighbor 10.2.0.8 activate
 exit-address-family
!
end
```

Inter-AS Confederation Configuration for Autonomous System 2 PE2 Example

The following example shows how to configure PE2 in autonomous system 2 in a confederation topology (see the figure above):

```
!
hostname PE2
!
ip cef
!
ip vrf VPN1
 rd 1:105
 route-target export 1:100
 route-target import 1:100
!
interface Loopback 0
 ip address 10.2.0.6 255.255.255.255
 ip router isis
!
interface Ethernet 0/0
 description Link to P2
 ip address 10.2.1.6 255.255.255.0
 ip router isis
 mpls ip
!
interface Serial 2/0
 description Link to CE2
 ip vrf forwarding VPN1
 ip address 192.168.2.2 255.255.255.0
 no fair-queue
 serial restart-delay 0
!
router ospf 10 vrf VPN1
 log-adjacency-changes
 redistribute bgp 2 subnets
 network 192.168.0.0 0.0.255.255 area 0
!
router isis
 net 49.0002.0000.0000.0009.00
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 100
 neighbor 10.2.0.7 remote-as 2
```

```
 neighbor 10.2.0.7 update-source Loopback 0
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.2.0.7 activate
 neighbor 10.2.0.7 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf VPN1
 redistribute connected
 redistribute ospf 10 vrf VPN1
 no auto-summary
 no synchronization
 exit-address-family
!
end
```

Inter-AS Confederation Configuration for Autonomous System 2 CE2 Example

The following example shows how to configure CE2 in VPN1 in a confederation topology (see the figure above):

```
!
hostname CE2
!
interface Loopback 0
 ip address 192.168.0.2 255.255.255.255
!
interface Serial 2/0
 description Link to PE2
 ip address 192.168.2.1 255.255.255.0
 no fair-queue
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
end
```

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

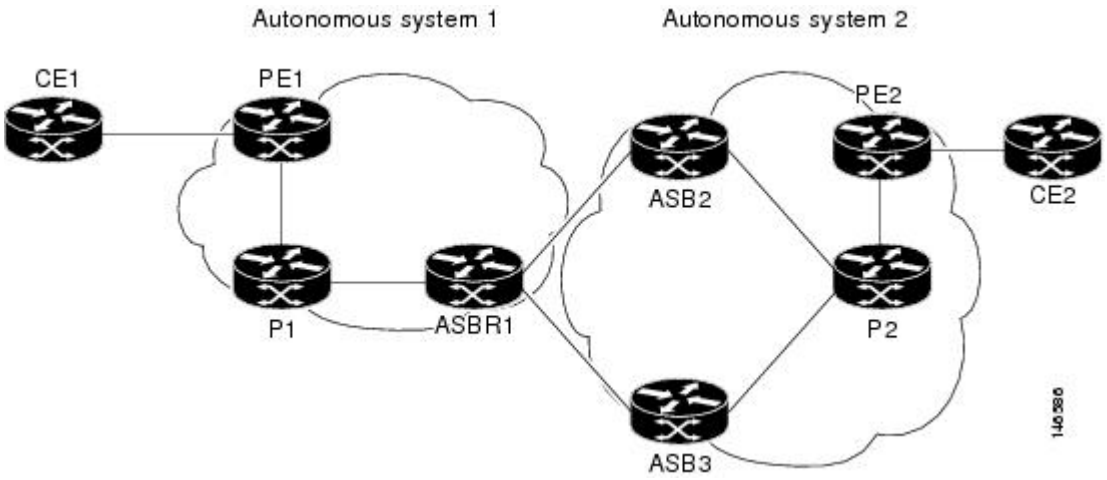
This section includes examples that show how to configure eBGP multipath load sharing for MPLS VPN Inter-AS ASBRS that exchange VPN-IPv4 routes. These configurations support the MPLS VPN - Interautonomous System Support feature.

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

- Autonomous system 1 contains PE1, P1, and ASBR1.
- Autonomous system 2 contains PE2, P2, ASBR2, and ASBR3.
- CE1 and CE2 belong to the same VPN, which is called VPN1.
- The P routers are route reflectors.
- ASBR1 and ASBR2 are configured with the **neighbor next-hop-self** command for the iBGP neighbors.

- ASBR1 and ASBR2 are configured with the **maximum paths** commands to set up eBGP multipath load sharing.

Figure 13 Configuring eBGP Multipath Load Sharing Between MPLS Inter-AS ASBRs Exchanging VPN-IPv4 Routes



The following examples shows how to configure eBGP multipath load sharing for MPLS VPN Inter-AS ASBRs that exchange VPN-IPv4 routes. This section includes sample configurations for P1, ASBR1, ASBR2, and P2 routers.

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

The following example shows how to configure CE1 in VPN1 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
!
```

```
hostname CE1
!
interface Loopback 1
 ip address 192.168.0.1 255.255.255.255
!
interface Ethernet 1/0
 description Link to PE1
 ip address 192.168.1.1 255.255.255.0
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
end
```

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

The following example shows how to configure PE1 in autonomous system 1 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
!
hostname PE1
!
ip cef
!
ip vrf V1
 rd 1:105
 route-target export 1:100
 route-target import 1:100
!
interface Loopback 0
 ip address 10.1.0.3 255.255.255.255
!
interface Ethernet 0/0
 description Link to CE1
 ip vrf forwarding V1
 ip address 192.168.1.2 255.255.255.0
!
interface Ethernet 1/0
 description Link to P1
 ip address 10.1.1.3 255.255.255.0
 mpls ip
!
router ospf 10 vrf V1
 log-adjacency-changes
 redistribute bgp 1 metric 100 subnets
 network 192.168.0.0 0.0.255.255 area 0
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.1.0.4 remote-as 1
 no neighbor 10.1.0.4 transport path-mtu-discovery
 neighbor 10.1.0.4 update-source Loopback 0
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.1.0.4 activate
 neighbor 10.1.0.4 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf V1
 redistribute ospf 10 vrf V1
 no auto-summary
 no synchronization
 exit-address-family
```

```
!
end
```

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

The following example shows how to configure P1 in autonomous system 1 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
!
hostname P1
!
ip cef
!
interface Loopback 0
 ip address 10.1.0.4 255.255.255.255
!
interface Ethernet 0/0
 description Link to PE1
 ip address 10.1.1.4 255.255.255.0
 mpls ip
!
interface Ethernet 1/0
 description Link to ASBR1
 ip address 10.1.2.4 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 1
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor R route-reflector-client
 neighbor 10.1.0.3 peer-group R
 neighbor 10.1.0.5 peer-group R
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor R route-reflector-client
 neighbor 10.1.0.3 activate
 neighbor 10.1.0.5 activate
 exit-address-family
!
end
```

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

The following example shows how to configure ASBR1 in autonomous system 1 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
hostname ASBR1
!
ip cef
!
interface Loopback 0
 ip address 10.1.0.5 255.255.255.255
!
interface Ethernet 0/0
 description Core link to P1
 ip address 10.1.2.5 255.255.255.0
```

```
 mpls ip
!
interface Ethernet 1/0
 description Link to ASBR2
 ip address 172.16.2.5 255.255.255.0
 mpls bgp forwarding
!
interface Serial 3/0
 description Link to ASBR3
 ip address 172.16.1.5 255.255.255.0
 mpls bgp forwarding
 serial restart-delay 0
!
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor 10.1.0.4 remote-as 1
 neighbor 172.16.1.9 remote-as 2
 neighbor 172.16.2.8 remote-as 2
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.1.0.4 activate
 neighbor 10.1.0.4 send-community extended
 neighbor 10.1.0.4 next-hop-self
 neighbor 172.16.1.9 activate
 neighbor 172.16.1.9 send-community extended
 neighbor 172.16.2.8 activate
 neighbor 172.16.2.8 send-community extended
 maximum-paths 2
 exit-address-family
!
end
```

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

The following example shows how to configure ASBR2 in autonomous system 2 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
!
hostname ASBR2
!
ip cef
!
interface Loopback 0
 ip address 10.2.0.8 255.255.255.255
!
interface Loopback 1
 no ip address
 shutdown
!
interface Ethernet 0/0
 description Link to ASBR1
 ip address 172.16.2.8 255.255.255.0
 mpls bgp forwarding
!
interface Serial 2/0
 description Link to P2
 ip address 10.2.2.8 255.255.255.0
 mpls ip
 no fair-queue
 serial restart-delay 0
!
```

```
router ospf 1
 log-adjacency-changes
 redistribute connected subnets
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 2
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor 10.2.0.7 remote-as 2
 neighbor 10.2.0.7 update-source Loopback 0
 neighbor 10.2.0.7 next-hop-self
 neighbor 172.16.2.5 remote-as 1
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.2.0.7 activate
 neighbor 10.2.0.7 send-community extended
 neighbor 10.2.0.7 next-hop-self
 neighbor 172.16.2.5 activate
 neighbor 172.16.2.5 send-community extended
 exit-address-family
!
end
```

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

The following example shows how to configure ASBR3 in autonomous system 2 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
!
hostname ASBR3
!
ip cef
!
interface Loopback 0
 ip address 10.2.0.9 255.255.255.255
!
interface Ethernet 0/0
 description Link to ASBR1
 ip address 172.16.1.9 255.255.255.0
 mpls bgp forwarding
!
interface Serial 3/0
 description Link to P2
 ip address 10.2.3.9 255.255.255.0
 mpls ip
 no fair-queue
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 redistribute connected subnets
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 2
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor 10.2.0.7 remote-as 2
 neighbor 10.2.0.7 update-source Loopback 0
 neighbor 10.2.0.7 next-hop-self
 neighbor 172.16.1.5 remote-as 1
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.2.0.7 activate
 neighbor 10.2.0.7 send-community extended
 neighbor 10.2.0.7 next-hop-self
```

```
 neighbor 172.16.1.5 activate
 neighbor 172.16.1.5 send-community extended
 exit-address-family
!
end
```

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

The following example shows how to configure P2 in autonomous system 2 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
!
hostname P2
!
ip cef
!
interface Loopback 0
 ip address 10.2.0.7 255.255.255.255
!
interface Ethernet 1/0
 description Link to PE2
 ip address 10.2.1.7 255.255.255.0
 mpls ip
!
interface Serial 2/0
 description Link to ASBR2
 ip address 10.2.2.7 255.255.255.0
 mpls ip
 no fair-queue
 serial restart-delay 0
!
interface Serial 3/0
 description Link to ASBR3
 ip address 10.2.3.7 255.255.255.0
 mpls ip
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 2
 no neighbor R transport path-mtu-discovery
 neighbor R update-source Loopback 0
 neighbor R route-reflector-client
 neighbor 10.2.0.6 peer-group R
 neighbor 10.2.0.8 peer-group R
 neighbor 10.2.0.9 peer-group R
 no auto-summary
 !
 address-family vpnv4
 neighbor R send-community extended
 neighbor R route-reflector-client
 neighbor 10.2.0.6 activate
 neighbor 10.2.0.8 activate
 neighbor 10.2.0.9 activate
 exit-address-family
!
end
!
```

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

The following example shows how to configure PE2 in autonomous system 2 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
hostname PE2
!
ip cef
!
ip vrf V1
 rd 1:105
 route-target export 1:100
 route-target import 1:100
!
interface Loopback 0
 ip address 10.2.0.6 255.255.255.255
!
interface Ethernet 0/0
 description Link to P2
 ip address 10.2.1.6 255.255.255.0
 mpls ip
!
interface Serial 2/0
 description Link to CE2
 ip vrf forwarding V1
 ip address 192.168.2.2 255.255.255.0
 no fair-queue
 serial restart-delay 0
!
router ospf 10 vrf V1
 log-adjacency-changes
 redistribute bgp 2 subnets
 network 192.168.0.0 0.0.255.255 area 0
!
router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.2.0.7 remote-as 2
 neighbor 10.2.0.7 update-source Loopback 0
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.2.0.7 activate
 neighbor 10.2.0.7 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf V1
 redistribute connected
 redistribute ospf 10 vrf V1
 no auto-summary
 no synchronization
 exit-address-family
!
end
```

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

The following example shows how to configure CE2 in VPN1 for the MPLS VPN - Interautonomous System Support feature (see the figure above):

```
hostname CE2
!
interface Loopback 0
 ip address 192.168.0.2 255.255.255.255
!
interface Serial 2/0
 description Link to PE2
```

```
 ip address 192.168.2.1 255.255.255.0
 no fair-queue
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
end
```

Additional References

Related Documents

Related Topic	Document Title
Configuration tasks for basic MPLS VPNs	Configuring MPLS VPNs
Configuration tasks for MPLS VPN Inter-AS system exchanging IPv4 routes and MPLS labels	MPLS VPN - Inter-AS—IPv4 BGP Label Distribution
Information about monitoring MPLS VPNs with MIBs	MPLS VPN—SNMP MIB Support

Standards

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

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1164	*Application of the Border Gateway Protocol in the Internet*
RFC 1700	*Assigned Numbers*
RFC 1771	*A Border Gateway Protocol 4*

RFC	Title
RFC 1965	*Autonomous System Confederation for BGP*
RFC 1966	*BGP Route Reflection: An Alternative to Full Mesh iBGP*
RFC 2547	*BGP/MPLS VPNs*
RFC 2842	*Capabilities Advertisement with BGP-4*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 3107	*Carrying Label Information in BGP-4*

Technical Assistance

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

Feature Information for MPLS VPN - Interautonomous System Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 3 Feature Information for MPLS VPN - Interautonomous System SupportMPLS VPN - Interautonomous System Support

Feature Name	Releases	Feature Information
MPLS VPN - Interautonomous System Support	12.1(5)T 12.0(16)ST 12.0(17)ST 12.0(22)S 12.0(23)S 12.2(13)T 12.0(24)S 12.2(14)S 12.0(29)S 12.2(33)SRA 12.2(33)SXH	The MPLS VPN - Interautonomous System Support feature allows an MPLS VPN to span service providers and autonomous systems. This feature module explains how to configure the Inter-AS using the ASBRs to exchange VPNv4 Addresses. In 12.1(5)T, this feature was introduced. In 12.0(16)ST, support for the Cisco 12000 series 4-Port OC-3c/STM-1c ATM line card (4-Port OC-3 ATM) and the Cisco 12000 series 4-Port OC-3c/STM-1c POS/SDH line card (4-port OC-3 POS) was added. In 12.0(17)ST, support for the Cisco 12000 series was added (See Feature Information for MPLS VPN - Interautonomous System Support, page 57 for the Cisco 12000 series line cards supported.) In 12.0(22)S, support for the Cisco 12000 series, the Cisco 10000 series edge services routers (ESRs), and the Cisco 10720 Internet routers was added. (See Feature Information for MPLS VPN - Interautonomous System Support, page 57 for the Cisco 12000 series line cards supported.) In 12.0(23)S, support was added for the Cisco 12000 series 8-port OC-3c/STM-1c ATM line card (8-Port OC-3 ATM) and the Cisco 12000 series 3-port Gigabit Ethernet line card (3-Port GbE). This feature was integrated into Cisco IOS Release 12.2(13)T. In 12.0(24)S, support was added for the Cisco 12000 series 1-port

Feature Name	Releases	Feature Information
		10-Gigabit Ethernet line card (1-Port 10-GbE) and the Cisco 12000 series modular Gigabit Ethernet/Fast Ethernet line card (modular GbE/FE) and this feature was implemented on Cisco IOS 12.0(24)S. This feature was integrated into Cisco IOS Release 12.2(14)S and implemented on Cisco 7200 and Cisco 7500 series routers. In 12.0(29)S, support was added for eBGP sessions between loopbacks of directly connected MPLS-enabled routers to provide for load sharing between neighbors. This feature was integrated into Cisco IOS Release 12.2(33)SRA. Support was added for load balancing of Virtual Private Network (VPN) traffic for VPNv4 peering. This feature was integrated into Cisco IOS Release 12.2(33)SXH.
MPLS VPN - Loadbalancing support for Inter-AS and CSC VPNs	12.0(29)S 12.2(33)SRA	This feature allows MPLS VPN Inter-AS and MPLS VPN Carrier Supporting Carrier (CSC) networks to load share traffic between adjacent LSRs that are connected by multiple links. The LSRs can be a pair of ASBRs or a CSC-PE and a CSC-CE. Using directly connected loopback peering allows load sharing at the IGP level, so more than one BGP session is not needed between the LSRs. No other label distribution mechanism is needed between the adjacent LSRs than BGP.

Feature Name	Releases	Feature Information
MPLS VPN—Multipath Support for Inter-AS VPNs	12.2(33)SRA 12.2(33)SXH	This feature supports Virtual Private Network (VPN)v4 multipath for Autonomous System Border Routers (ASBRs) in the interautonomous system (Inter-AS) Multiprotocol Label Switching (MPLS) VPN environment. It allows load balancing of VPN traffic when you use the VPNv4 peering model for Inter-AS VPNs.

Glossary

autonomous system—A collection of networks under a common administration sharing a common routing strategy.

BGP —Border Gateway Protocol. An interdomain routing protocol that exchanges network reachability information with other BGP systems (which may be within the same autonomous system or between multiple autonomous systems).

CeBGP —confederation exterior Border Gateway Protocol. A BGP between routers located within different subautonomous systems of a confederation. See *eBGP* and *iBGP* .

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

confederation —An autonomous system divided into multiple, separate subautonomous systems and classified as a single unit.

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

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

IGP —Interior Gateway Protocol. Internet protocol used to exchange routing information within a single autonomous system. Examples of common Internet IGP protocols include IGRP, OSPF, IS-IS, and RIP.

LFIB —Label Forwarding Information Base. Data structure used in MPLS to hold information about incoming and outgoing labels and associated Forwarding Equivalence Class (FEC) packets.

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

NLRI —Network Layer Reachability Information. The BGP sends routing update messages containing NLRI to describe a route and how to get there. In this context, an NLRI is a prefix. A BGP update message carries one or more NLRI prefixes and the attributes of a route for the NLRI prefixes; the route attributes include a BGP next hop gateway address and extended community values.

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

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

VPN —Virtual Private Network. A secure MPLS-based network that shares resources on one or more physical networks (typically implemented by one or more service providers). A VPN contains geographically dispersed sites that can communicate securely over a shared backbone network.

VRF —VPN routing and forwarding instance. Routing information that defines a Virtual Private Network (VPN) site that is attached to a provider edge (PE) router. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table.

MPLS VPN 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

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

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

Prerequisites for MPLS VPN 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 4 Cisco 12000 Series Line Card Support in Cisco IOS S Releases

Type	Line Cards	Cisco IOS Release Supported
ATM	4-Port OC-3 ATM	12.0(22)S
	1-Port OC-12 ATM	12.0(23)S
	4-Port OC-12 ATM	12.0(27)S
	8-Port OC-3 ATM	
Channelized interface	2-Port CHOC-3	12.0(22)S
	6-Port Ch T3 (DS1)	12.0(23)S
	1-Port CHOC-12 (DS3)	12.0(27)S
	1-Port CHOC-12 (OC-3)	
	4-Port CHOC-12 ISE	
	1-Port CHOC-48 ISE	
Electrical interface	6-Port DS3	12.0(22)S
	12-Port DS3	12.0(23)S
	6-Port E3	12.0(27)S
	12-Port E3	
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	12.0(22)S
	8-Port OC-3 POS	12.0(23)S
	16-Port OC-3 POS	12.0(27)S
	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	

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 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 travers only e a single 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 IBGP meshing: Internal Border Gateway Protocol (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 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:

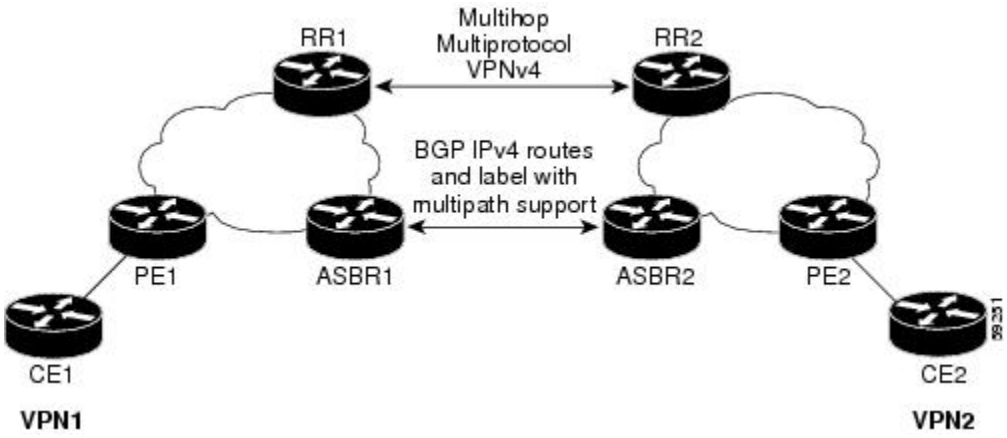
- Saves the ASBRs from having to store all the VPN-IPv4 routes. Using the route reflectors to store the VPN-IPv4 routes and forward them to the PE routers results in improved scalability compared wtih configurations where the ASBR holds all of the VPN-IPv4 routes and forwards the routes based on VPN-IPv4 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, page 67
- Types of BGP Messages and MPLS Labels, page 68
- How BGP Sends MPLS Labels with Routes, page 68

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.

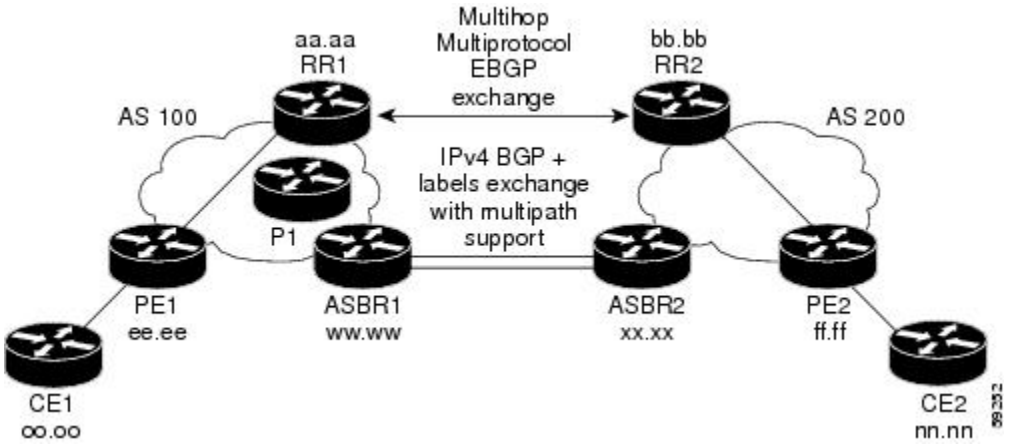
When you issue the **neighbor send-label** command on both BPG routers, the routers advertise to each other that they can then send MPLS labels with the routes. If the routers successfully negotiate their ability to send MPLS labels, the routers add MPLS labels to all outgoing BGP updates.

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.

	Command or Action	Purpose
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor bb.bb.bb.bb 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 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.

	Command or Action	Purpose
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. • 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.

	Command or Action	Purpose
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.
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.

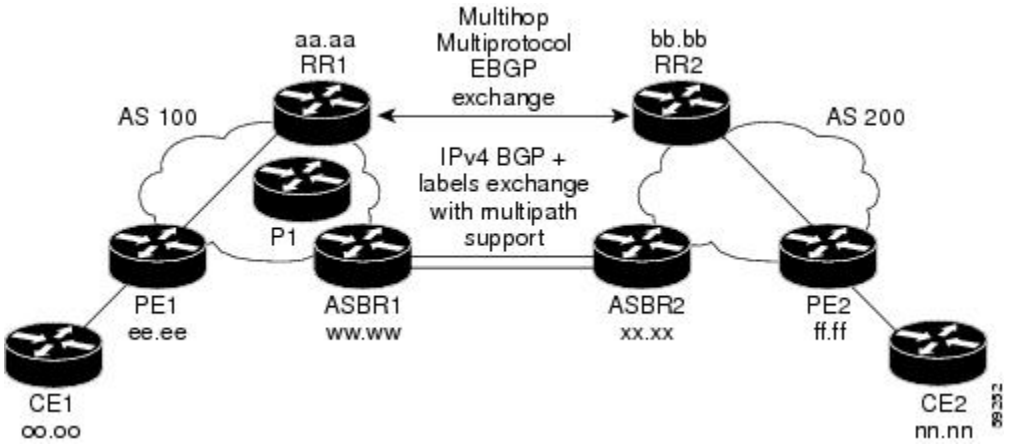
Step 12	**exit-address-family**
	
	Example:
	`Router(config-router-af)#`
	`exit-address-family`
		Exits address family configuration mode.

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.

	Command or Action	Purpose
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.

Command or Action	Purpose
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.

Command or Action	Purpose
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`
 	
Example:
`Router# show ip bgp vpnv4 all 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.

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).

Command or Action	Purpose
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.

	Command or Action	Purpose
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**] **Example:** `Router# show ip cef vpn1 nn.nn.nn.nn`	(Optional) Displays entries in the FIB or displays a summary of the FIB. • 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 , page 83

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. ◦ 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

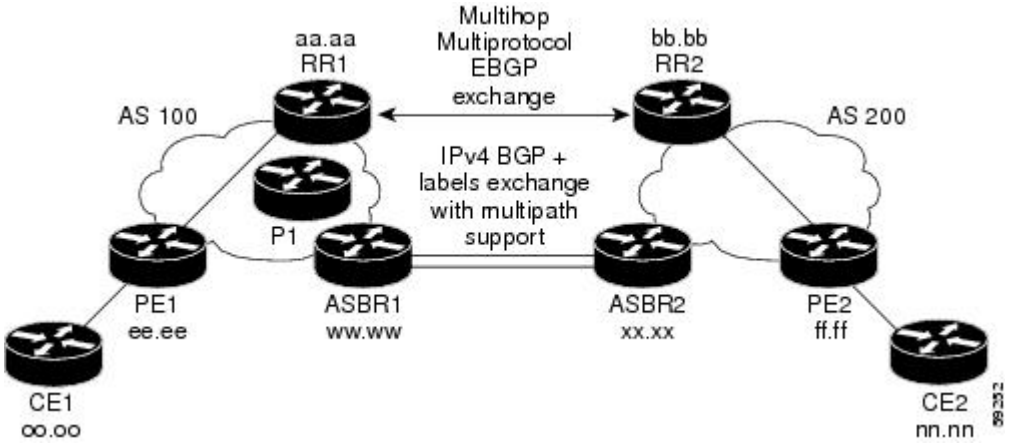
Configuring MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels over an MPLS VPN Service Provider Examples

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.

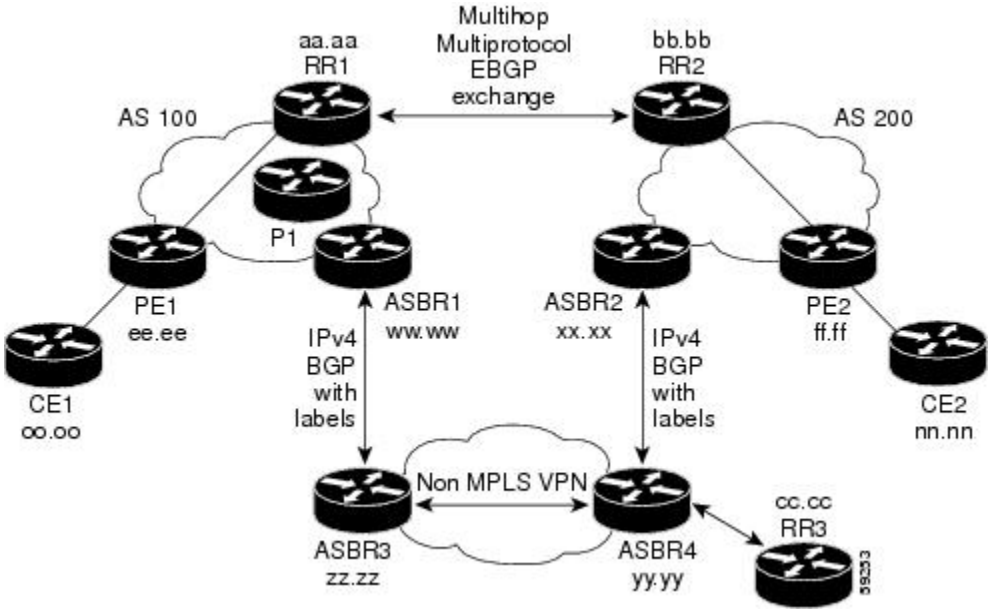
```
ip subnet-zero
ip cef
!
```

```
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 5 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	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). This feature uses no new or modified commands.

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

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

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

Prerequisites for MPLS VPN 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	12.0(16)ST
	1-Port OC-12 POS	12.0(21)ST
	8-Port OC-3 POS	12.0(22)S
	16-Port OC-3 POS	
	4-Port OC-12 POS	
	1-Port OC-48 POS	
	4-Port OC-3 POS ISE	
	8-Port OC-3 POS ISE	
	16 x OC-3 POS ISE	
	4 Port OC-12 POS ISE	
	1-Port OC-48 POS ISE	
Electrical Interface	6- Port DS3	12.0(16)ST
	12- Port DS3	12.0(21)ST
	6-Port E3	
ATM	4-Port OC-3 ATM	12.0(22)S
	1-Port OC12 ATM	
	4-Port OC-12 ATM	

Type	Line Cards	Cisco IOS Release Added
Channelized Interface	2-Port CHOC-3	12.0(22)S
	6-Port Ch T3 (DS1)	
	1-Port CHOC-12 (DS3)	
	1-Port CHOC-12 (OC-3)	
	4-Port CHOC-12 ISE	
	1-Port CHOC-48 ISE	

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, page 104

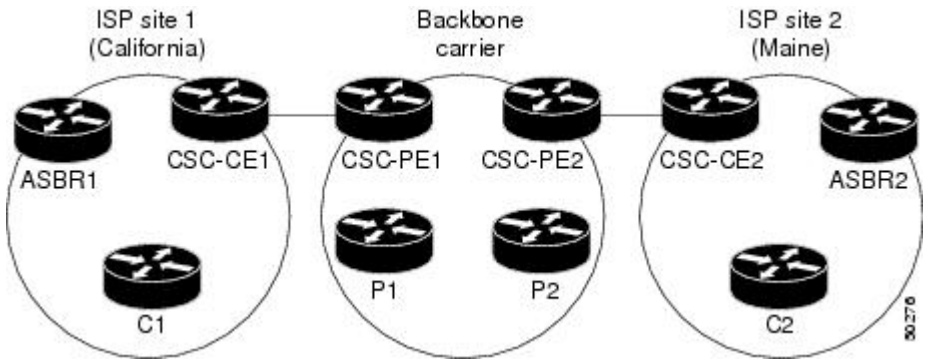
Customer Carrier Is an ISP

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.

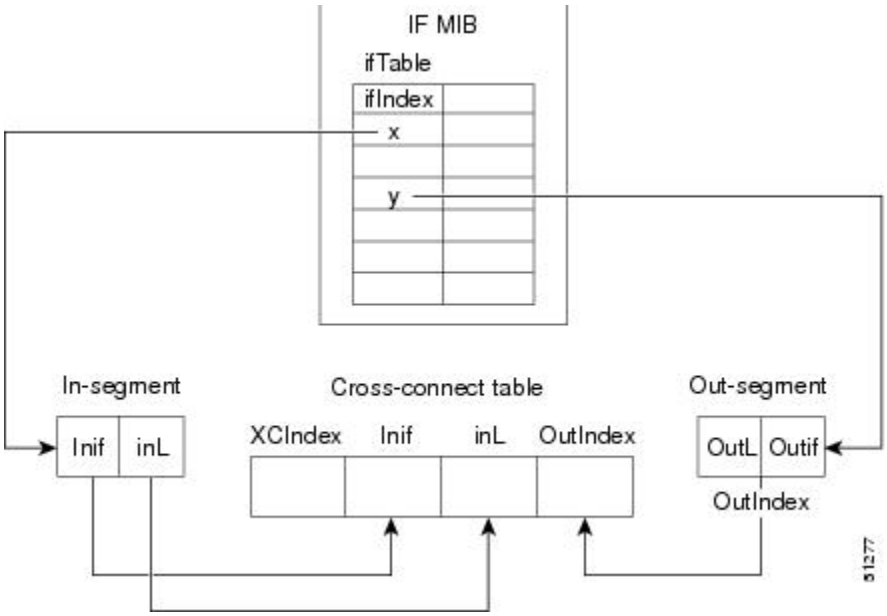
Internal and external routes are differentiated this way:

- 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.

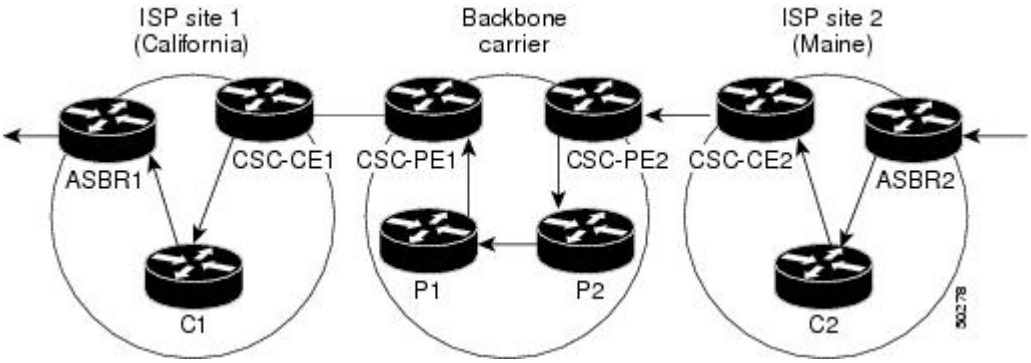
Because the PE routers do not have to carry external routes in the VRF routing table, they can use the incoming label in the packet to forward the customer carrier Internet traffic. Adding MPLS to the routers provides a consistent method of transporting packets from the customer carrier to the backbone 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.

Step	Description
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.

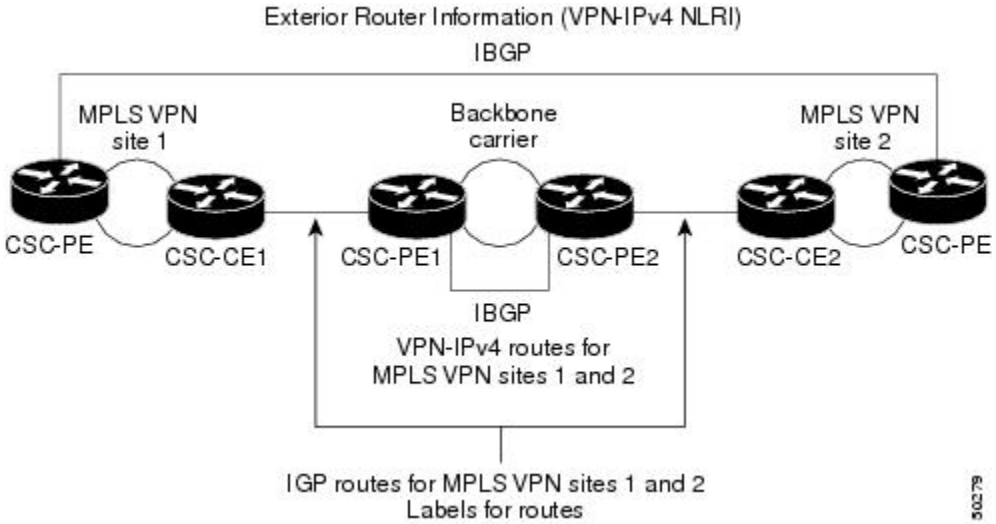
Customer Carrier Is a BGP MPLS VPN Service Provider

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.

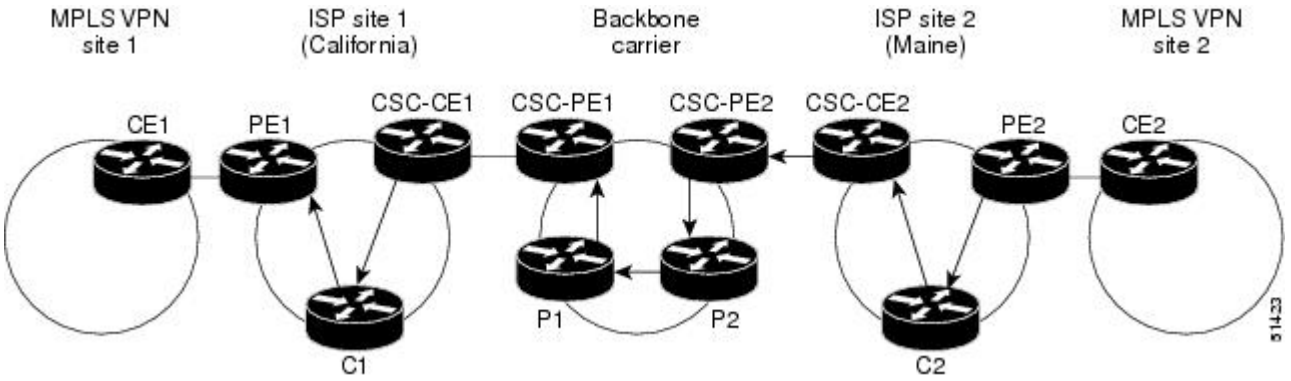
The figure below shows how information is exchanged when MPLS VPN services reside on all customer carrier sites and on the backbone carrier.

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.
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.

Step	Description
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, page 109.

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* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]] [**vrf** *vrf-name*] [**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.

	Command or Action	Purpose
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, page 111

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* **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

	Command or Action	Purpose
Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf)# route-target import 100:1`	Creates a route-target extended community for a VRF. • The **import** keyword imports routing information from the target VPN extended community. • The **export** keyword exports routing information to the target VPN extended community. • The **both** keyword imports routing information from and exports routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities.
Step 6	**import map** *route-map* **Example:** `Router(config-vrf)# import map vpn1-route-map`	(Optional) Configures an import route map for a VRF. • The *route-map* argument specifies the route map to be used as an import route map for the VRF.
Step 7	**exit** **Example:** `Router(config-vrf)# exit`	(Optional) Exits to global configuration mode.
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.
Step 10	**end** **Example:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.

- Troubleshooting Tips, page 114

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.
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.

	Command or Action	Purpose
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.
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.

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

- Troubleshooting Tips, page 116
- Troubleshooting Tips, page 179

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, page 116
- Configuring LDP on the CSC-PE and CSC-CE Routers, page 116
- Enabling MPLS Encapsulation on the CSC-PE and CSC-CE Routers, page 118

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.

	Command or Action	Purpose
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.

	Command or Action	Purpose
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 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:

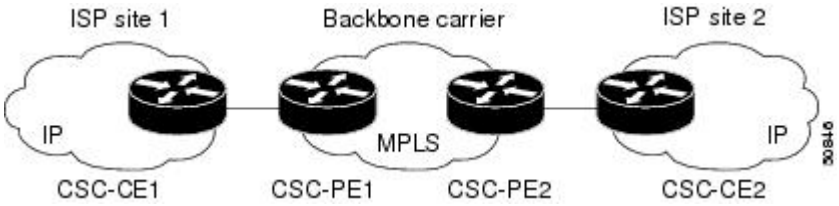
- xmit indicates that the interface is transmitting LDP discovery hello packets.
- recv indicates that the interface is receiving LDP discovery hello packets.

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

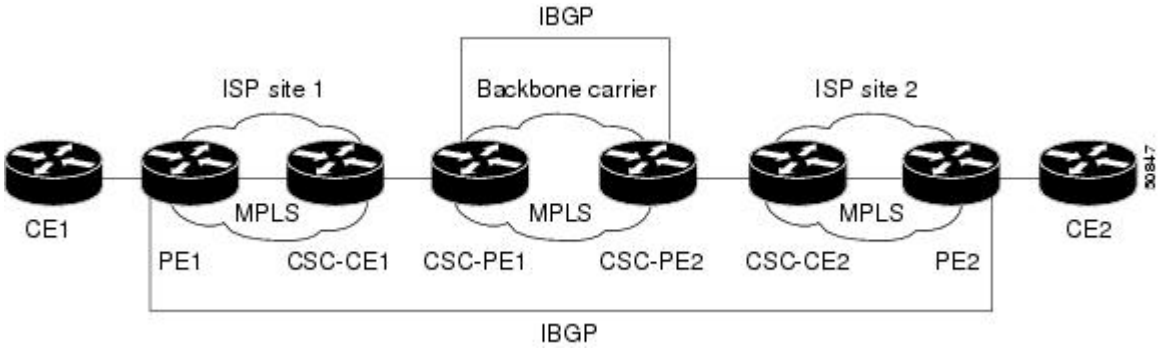
```
ip cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address 10.16.16.16 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 atm clock INTERNAL
 atm sonet stm-1
 no atm enable-ilmi-trap
 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, page 125
- PE1 Configuration, page 126
- CSC-CE1 Configuration, page 127
- CSC-PE1 Configuration, page 128
- CSC-PE2 Configuration, page 129
- CSC-CE2 Configuration, page 130
- PE2 Configuration, page 131
- CE2 Configuration, page 132

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
```

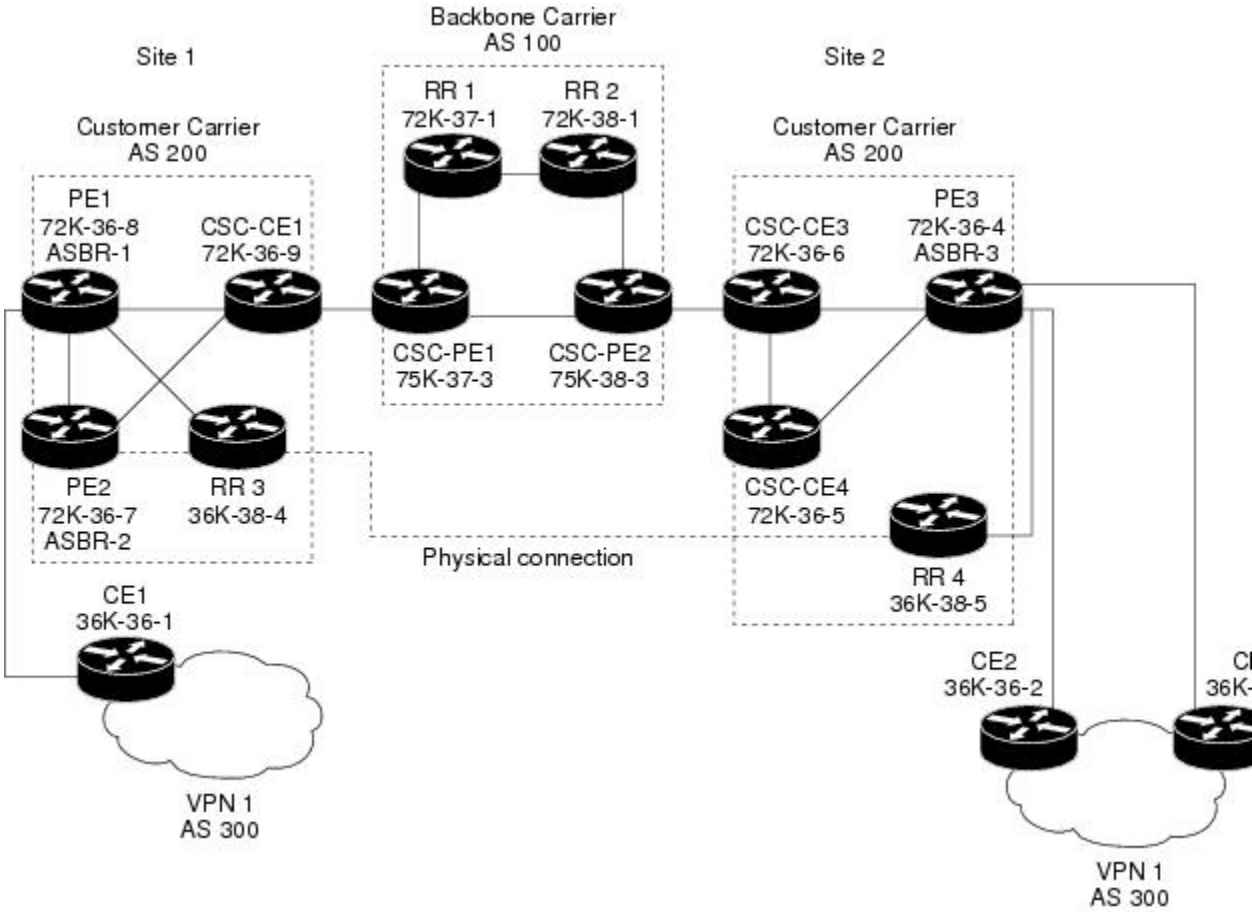
CE2 Configuration

```
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, page 134
- Customer Carrier Site 1 Configuration, page 139

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

- CE2 (36K-36-2) Configuration, page 147
- CE3 (36K-36-3) Configuration, page 147

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

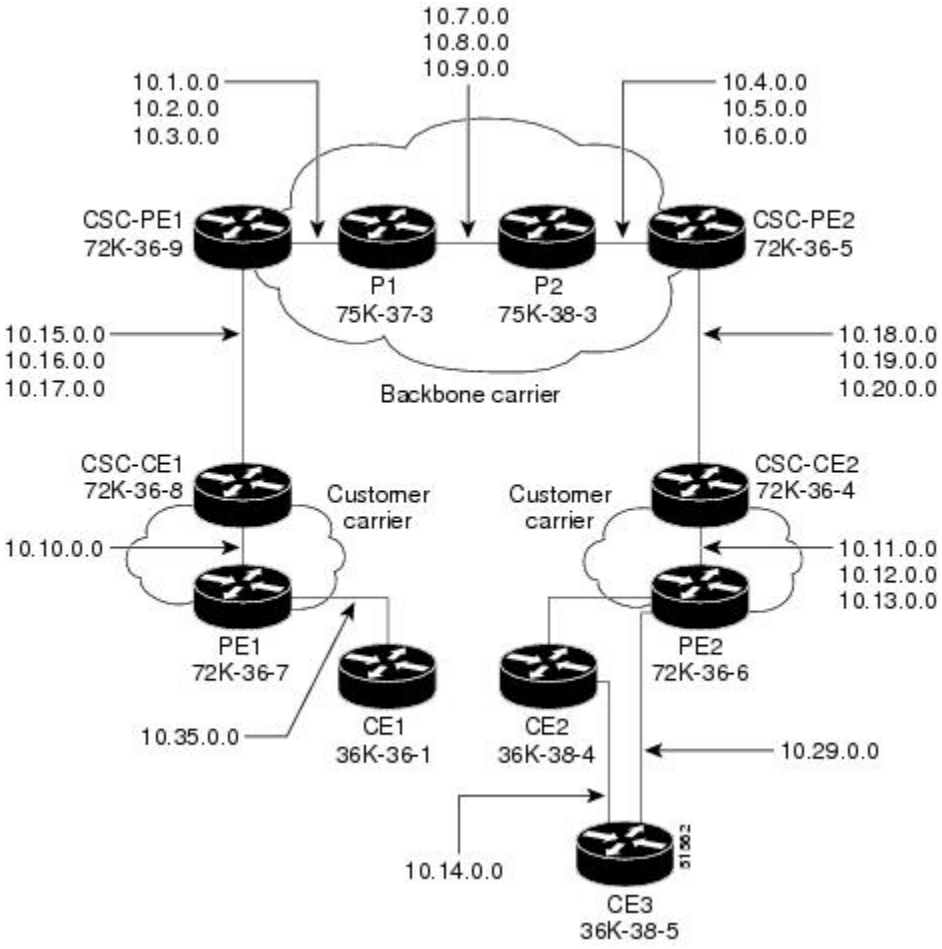
```
ip cef
!
```

```
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

The following sections provide references related to MPLS VPNs.

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 2547	BGP/MPLS VPNs

Technical Assistance

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

Feature Information for MPLS VPN 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	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. 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.

and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

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

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

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

Prerequisites for MPLS VPN 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

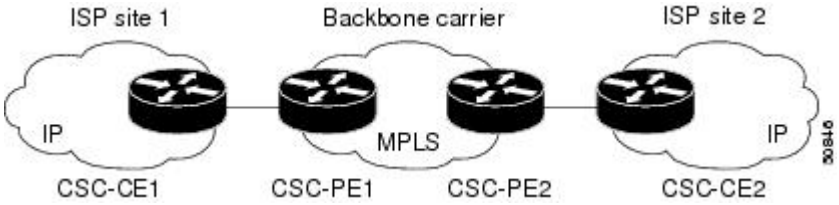
The following sections explain how the backbone and customer carriers distribute IPv4 routes and MPLS labels. The backbone carrier offers BGP and MPLS VPN services. The customer carrier can be either of the following:

- Customer Carrier Is an ISP with an IP Core, page 169
- Customer Carrier Is an MPLS Service Provider With or Without VPN Services, page 170

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



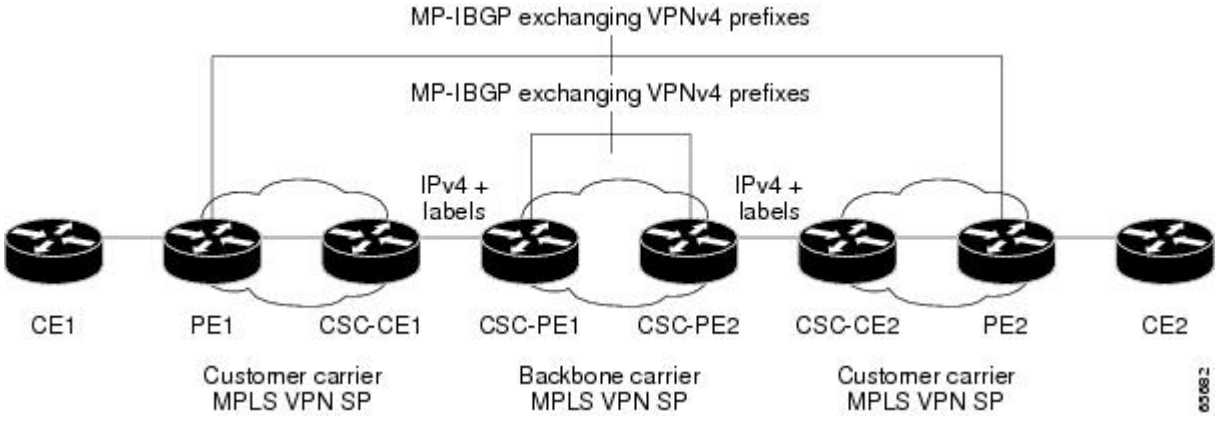
The links between the CE and PE routers use EBGP to distribute IPv4 routes and MPLS labels. Between the links, the PE routers use multiprotocol IBGP to distribute VPNv4 routes.

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).

	Command or Action	Purpose
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, page 172

What to Do Next

Set up your carrier supporting carrier networks with the Configuring the Backbone Carrier Core, page 172.

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, page 172
- Verifying IP Connectivity and LDP Configuration in the CSC Core, page 172
- Configuring VRFs for CSC-PE Routers, page 175
- Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier, page 114

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 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.

	Command or Action	Purpose
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, page 174

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.
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

	Command or Action	Purpose
Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf)# route-target import 100:1`	Creates a route-target extended community for a VRF. • The **import** keyword imports routing information from the target VPN extended community. • The **export** keyword exports routing information to the target VPN extended community. • The **both** keyword imports routing information from and exports routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities.
Step 6	**import map** *route-map* **Example:** `Router(config-vrf)# import map vpn1-route-map`	(Optional) Configures an import route map for a VRF. • The *route-map* argument specifies the route map to be used as an import route map for the VRF.
Step 7	**exit** **Example:** `Router(config-vrf)# exit`	(Optional) Exits to global configuration mode.
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.
Step 10	**end** **Example:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.

- Troubleshooting Tips, page 177

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.
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.

	Command or Action	Purpose
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.
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.

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

- Troubleshooting Tips, page 116
- Troubleshooting Tips, page 179

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, page 179
- Configuring CSC-CE Routers, page 182
- Verifying Labels in the CSC-PE Routers, page 184
- Verifying Labels in the CSC-CE Routers, page 186

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.

	Command or Action	Purpose
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor 10.0.0.1 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.2 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, page 181

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.

	Command or Action	Purpose
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. • 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.

	Command or Action	Purpose
Step 9	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits from the address family configuration mode.
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.

Command or Action	Purpose
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.
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.

Command or Action	Purpose
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. **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.
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.

	Command or Action	Purpose
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*} **Example:** `Router# ping ip 10.2.0.0`	Diagnoses basic network connectivity on AppleTalk, CLNS, IP, Novell, Apollo, VINES, DECnet, or XNS networks. • Use the **ping** command to verify the connectivity from one 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.

	Command or Action	Purpose
Step 5	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes. • The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 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.
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.

	Command or Action	Purpose
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. • 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.

	Command or Action	Purpose
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** **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 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.

	Command or Action	Purpose
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.
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**] **Example:** `Router# show ip cef 10.2.0.0`	(Optional) Displays specific entries in the FIB based on IP address information. • 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** **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 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.

	Command or Action	Purpose
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* **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 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

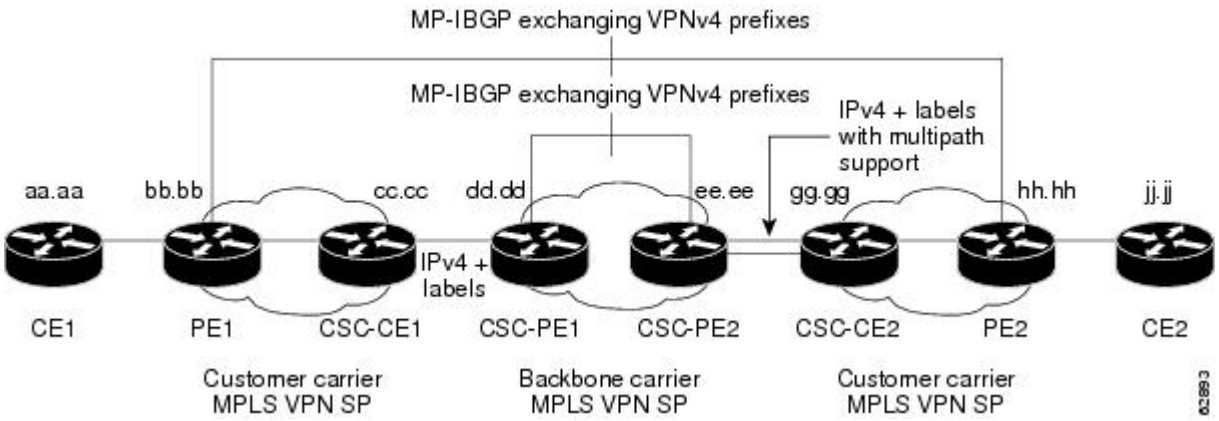
	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip route** [*ip-address* [*mask*]] [**longer-prefixes**]	*protocol* [*process-id*]	**list** [*access-list-number*	*access-list-name*]	**static download** **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.

	Command or Action	Purpose
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.

Routers	Description
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.
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, page 201
- Configuring the Links Between CSC-PE and CSC-CE Routers Examples, page 203
- Configuring the Customer Carrier Network Examples, page 209
- Configuring the Customer Site for Hierarchical VPNs Examples, page 210

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, page 201
- Configuring VRFs for CSC-PE Routers Example, page 203
- Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier Example, page 203

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, page 207.

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, page 209
- Configuring a Customer Carrier Core Router as a Route Reflector Example, page 210

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 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	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. 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.

Load Sharing MPLS VPN Traffic

Load sharing distributes traffic so that no individual router is overburdened. In a Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) network, you can achieve load sharing through the following methods:

- BGP multipath options
- Directly connected loopback peering

Finding Feature Information

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

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

Prerequisites for Load Sharing MPLS VPN Traffic

Before configuring load sharing, ensure that your MPLS VPN network (including MPLS VPN carrier supporting carrier or interautonomous system) is configured and working properly. See the Prerequisites for Load Sharing MPLS VPN Traffic, page 219 for references related to MPLS VPNs.

Restrictions for Load Sharing MPLS VPN Traffic

- Configuring BGP multipath for eBGP and iBGP is only for basic MPLS Layer 3 VPNs. MPLS VPN Inter-AS and MPLS VPN carrier supporting carrier do not support this multipath configuration.

- With multiple iBGP paths installed in a routing table, a route reflector advertises only one of the paths (one next hop). If a router is behind a route reflector, all routers that are connected to multihomed sites are not advertised unless separate VRFs with different RDs are configured for each VRF.
- Each IP routing table entry for a BGP prefix that has multiple iBGP paths uses additional memory. We recommend not using this feature on a router with a low amount of available memory and especially when the router is carrying a full Internet routing table.
- eBGP Multipath is not supported on MPLS VPN Inter-AS with ASBRs that exchange VPNv4 routes.
- Load sharing using directly connected loopback peering does not apply to CSC networks that use LDP and an IGP to distribute routes and MPLS labels.

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

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 router's loopback address, as in EBGP multihop cases.

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

The following **ip route** commands are supported when you configure static routes in 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 Load Sharing MPLS VPN Traffic

Overview of Load Sharing Using BGP Multipath Options

A variety of Border Gateway Protocol (BGP) multipath options exist that enable you to configure load sharing on your MPLS VPN that uses BGP.

To load share traffic at the iBGP multipath level, it is recommended that you configure BGP labeling using the **neighbor send-label** command in router configuration mode. When you configure the iBGP multipath feature, the following message is displayed as a reminder to use the neighbor send-label command functionality:

```
WARNING: Using iBGP multipath feature with LDP or TE based LSPs towards the BGP nexthop,
paths taken by forwarding may not be as expected. Please consider configuring BGP
labeling (RFC 3107) for proper forwarding behavior.
```

The following sections describe some BGP multipath options:

Internal BGP Multipath Load Sharing

When a BGP-speaking router with no local policy configured receives multiple network layer reachability information (NLRI) from the internal BGP (iBGP) for the same destination, the router chooses one iBGP path as the best path. The best path is then installed in the IP routing table of the router. The iBGP multipath feature enables the BGP-speaking router to select multiple iBGP paths as the best paths to a destination. The best paths are then installed in the IP routing table of the router. To enable iBGP multipath load sharing, you issue the **maximum-paths ibgp** command in router configuration mode. For more information about iBGP multipath load sharing, see Configuring BGP.

BGP Multipath for eBGP and iBGP

The BGP multipath load sharing for both eBGP and iBGP in an MPLS VPN feature allows multihomed autonomous systems and provider edge (PE) routers to be configured to distribute traffic across both external BGP (eBGP) and iBGP paths.

BGP installs up to the maximum number of paths allowed (configured using the **maximum-paths** command). BGP uses the best path algorithm to select one multipath as the best path, inserts the best path into the routing information base (RIB), and advertises the best path to BGP peers. Other multipaths can be inserted into the RIB, but only one path is selected as the best path.

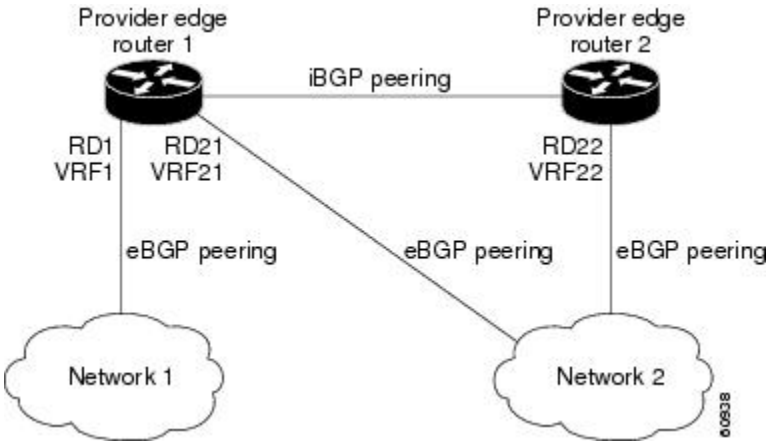
Cisco Express Forwarding uses mutlipaths to perform load balancing on a per-packet or per-source or destination pair basis. To enable the load sharing feature, configure the router with MPLS VPNs that contain VPN routing and forwarding instances (VRFs) that import both eBGP and iBGP paths. You can configure the number of multipaths separately for each VRF.

Note This feature operates within the configuration parameters of the existing outbound routing policy.

- eBGP and iBGP Multipath Load Sharing in an MPLS Network Using BGP, page 223
- eBGP and iBGP Multipath Load Sharing with Route Reflectors, page 223

eBGP and iBGP Multipath Load Sharing in an MPLS Network Using BGP

The figure below shows an MPLS service provider network using BGP that connects two remote networks to PE1 and PE2, which are both configured for VPNv4 unicast iBGP peering. Network 2 is a multihomed network that is connected to PE1 and PE2. Network 2 also has extranet VPN services configured with Network 1. Both Network 1 and Network 2 are configured for eBGP peering with the PE routers.



You can configure PE1 so that both iBGP and eBGP paths can be selected as multipaths and imported into the VRF of Network 1. Cisco Express Forwarding uses the mutlipaths to perform load balancing. Traffic is distributed as follows:

- IP traffic that is sent from Network 2 to PE1 and PE2 is sent across the eBGP paths as IP traffic.
- IP traffic that is sent from PE1 to PE2 is sent across the iBGP path as MPLS traffic.
- MPLS traffic that is sent across an eBGP path is sent as IP traffic.

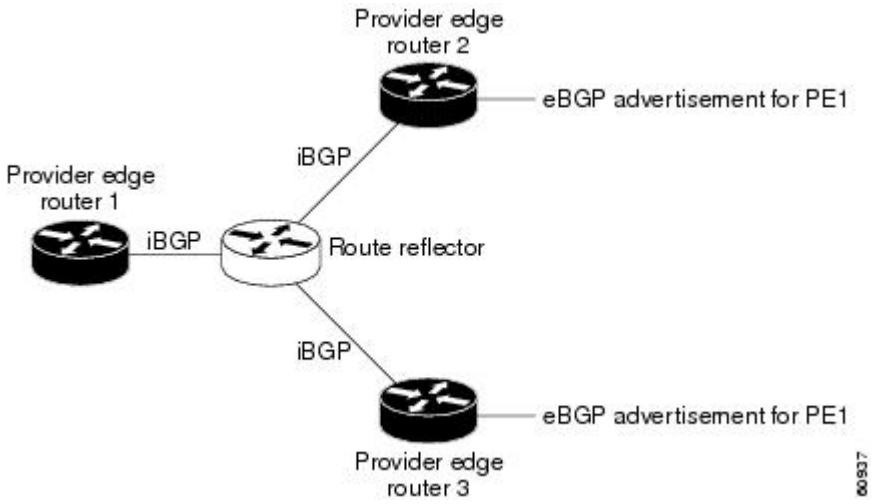
Any prefix that is advertised from Network 2 will be received by PE1 through route distinguisher (RD) 21 and RD22.

- The advertisement through RD21 is carried in IP packets.
- The advertisement through RD22 is carried in MPLS packets.

Both paths can be selected as multipaths for VRF1 and inserted into the VRF1 RIB.

eBGP and iBGP Multipath Load Sharing with Route Reflectors

The figure below shows a topology that contains three PE routers and a route reflector, all configured for iBGP peering. PE2 and PE3 each advertise an equal preference eBGP path to PE1. By default, the route reflector chooses only one path and advertises PE1.



For all equal preference paths to PE1 to be advertised through the route reflector, you must configure each VRF with a different RD. The prefixes received by the route reflector are recognized differently and advertised to PE1.

eBGP Multipath Load Sharing

When a router learns two identical 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 when the eBGP paths are learned from a neighboring autonomous system, instead of picking one best path.

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.

Load Sharing Using Directly Connected Loopback Peering

You use this feature with MPLS VPN Inter-AS and MPLS VPN carrier supporting carrier (CSC) networks 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 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.

Directly connected loopback peering enables load sharing of traffic as follows:

- A BGP session is established, using the loopback addresses of the LSRs.
- 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 Load Sharing

Configuring BGP Multipath Load Sharing for eBGP and iBGP

To configure iBGP and eBGP routes for multipath load sharing, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **maximum-paths eibgp** *number-of-paths*

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 1`	Enters router configuration mode and configures the router to run a BGP routing process.
Step 4	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 vrf vrf1`	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv4 address prefixes. **Note** For this task you must create the VRF and specify the **vrf** keyword. • 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 eibgp** *number-of-paths* **Example:** `Router(config-router-af)# maximum-paths eibgp 6`	Configures the number of parallel iBGP and eBGP routes that can be installed into a routing table.

Verifying BGP Multipath Load Sharing for eBGP and iBGP

To verify the configuration of iBGP and eBGP routes for multipath load sharing, perform this task.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} | [**rib-failure**] [*ip-prefix/length* [**longer-prefixes**]] [*network-address* [*mask*] [**longer-prefixes**]] [**cidr-only**] [**community**] [**community-list**] [**dampened-paths**] [**filter-list**] [**flap-statistics**] [**inconsistent-as**] [**neighbors**] [**paths** [*line*]] [**peer-group**] [**quote-regexp**] [**regexp**] [**summary**] [**labels**]

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} | [**rib-failure**] [*ip-prefix/length* [**longer-prefixes**]] [*network-address* [*mask*] [**longer-prefixes**]] [**cidr-only**] [**community**] [**community-list**] [**dampened-paths**] [**filter-list**] [**flap-statistics**] [**inconsistent-as**] [**neighbors**] [**paths** [*line*]] [**peer-group**] [**quote-regexp**] [**regexp**] [**summary**] [**labels**] **Example:** `Router# show ip bgp vpnv4 all`	Displays attributes and multipaths for a specific network in an MPLS VPN. • Enter one or more keywords or arguments.

Configuring eBGP Multipath Load Sharing with MPLS VPN Inter-AS

Perform this task on the ASBRs to configure eBGP Multipath for MPLS 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:** `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 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 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*] **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 **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:** `Router(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:** `Router(config-router-af)# neighbor 10.0.0.1 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 8	**neighbor** *ip-address* **send-label** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 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 9	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 10	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Configuring eBGP Multipath Load Sharing with MPLS VPN Carrier Supporting Carrier on the CSC-PE Routers

Perform this task to configure eBGP Multipath load sharing on the CSC-PE routers that distribute BGP routes with 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**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	**maximum-paths** *number-paths* **Example:** `Router(config-router-af)# maximum-paths 2`	(Optional) Controls the maximum number of parallel routes an IP routing protocol can support. • On the CSC-PE router, 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:** `Router(config-router-af)# neighbor 10.0.0.1 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a 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 8	**neighbor** *ip-address* **as-override** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 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 9	**neighbor** *ip-address* **send-label** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 send-label`	Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighboring router.
Step 10	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 11	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Configuring eBGP Multipath Load Sharing with MPLS VPN Carrier Supporting Carrier on the CSC-CE Routers

Perform this task to configure eBGP Multipath load sharing on the CSC-CE routers.

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:** `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	**maximum-paths** *number-paths* **Example:** `Router(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.

	Command or Action	Purpose
Step 5	**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. • 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:** `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**, **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 IS-IS, these routes are redistributed as external to the autonomous system.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(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:** `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 9	**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.

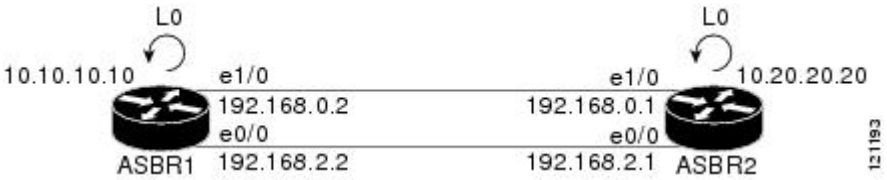
	Command or Action	Purpose
Step 10	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 11	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Configuring DCLP 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 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 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** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface- number* **Example:** `Router(config)# interface loopback 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:** `Router(config-if)# ip address 10.10.10.10 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Configuring 32 Static Routes to the eBGP Neighbor Loopback

Perform this task to configure /32 static routes to the 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Router(config)# ip route 10.20.20.20 255.255.255.255 Ethernet 1/0 172.16.0.1`	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:** `Router(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, page 235 task, 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:** `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 slot/port* **Example:** `Router(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:** `Router(config-if)# mpls bgp forwarding`	Configures BGP to enable MPLS forwarding on connecting interfaces.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(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:** `Router(config)# end`	Exits to privileged EXEC mode.

Configuring an eBGP Session Between the Loopbacks

Perform this task to configure an eBGP session between the loopbacks.

Note You need to configure an eBGP session between loopbacks on each directly connected ASBR.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default route-target filter**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check**
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
8. **address-family vpnv4** [**unicast**]
9. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
10. **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard extended**]
11. **end**

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 the BGP routing process. • The *as-number* indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along.
Step 4	**no bgp default route-target filter** **Example:** `Router(config)# 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 router.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.20.20.20 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group. • The *as-number* argument is the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check** **Example:** `Router(config-router)# neighbor 10.20.20.20 disable-connected-check`	Allows peering between loopbacks. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group.

	Command or Action	Purpose
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Router(config-router)# neighbor 10.20.20.20 update-source Loopback 0`	Allows BGP sessions to use any operational interface for TCP connections. • The *ip-address* argument is the IPv4 address of the BGP-speaking neighbor. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons. • The *peer-group-name* argument is the name of a BGP peer group. • The *interface-type* argument is the interface type. • The *interface-number* argument is the interface number.
Step 8	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing protocols such as BGP, Routing Information Protocol (RIP), and static routing. • The **unicast** keyword specifies unicast prefixes.
Step 9	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Router(config-router-af)# neighbor 10.20.20.20 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring router. • The *peer-group-name* argument is the name of a BGP peer group. • The *ipv6-address* argument is the IPv6 address of the BGP-speaking neighbor. **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:** `Router(config-router-af)# neighbor 10.20.20.20 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring router. • The *peer-group-name* argument is the name of a BGP peer group. • The **both** keyword specifies that both standard and extended communities will be sent. • The **standard** keyword specifies that only standard communities will be sent. • The **extended** keyword specifies that only extended communities will be sent.
Step 11	**end** **Example:** `Router(config)# 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 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**

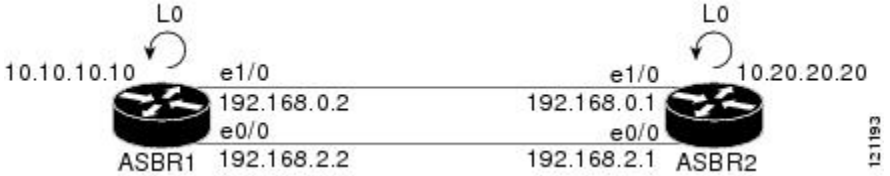
DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> 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:** `Router# show mpls forwarding-table`	Displays the contents of the MPLS LFIB. • Enter an optional keyword or argument if desired.
Step 3	**disable** **Example:** `Router# disable`	Exits to user EXEC mode.

Configuring DCLP 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 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

Perform this task to configure loopback interface addresses.

Note Loopback addresses need to be configured for each directly connected 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface number* **Example:** `Router(config)# interface loopback 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:** `Router(config-if)# ip address 10.10.10.10 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Configuring 32 Static Routes to the eBGP Neighbor Loopback

Perform this task to configure /32 static routes to the 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Router(config)# ip route 10.20.20.20 255.255.255.255 Ethernet 1/0 172.16.0.1`	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:** `Router(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, page 243 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:** `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 slot/port* **Example:** `Router(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:** `Router(config-if)# mpls bgp forwarding`	Configures BGP to enable MPLS forwarding on connecting interfaces.
Step 5	**exit** **Example:** `Router(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:** `Router(config)# end`	Exits to privileged EXEC mode.

Configuring an eBGP Session Between the Loopbacks

Perform the following tasks to configure an eBGP session between the loopbacks.

Note You need to configure an eBGP session between loopbacks on each directly connected 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* | *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* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**
11. **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 the 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.
Step 4	**bgp log-neighbor-changes** **Example:** `Router(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.

	Command or Action	Purpose
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.20.20.20 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group. • The *as-number* argument is the number of the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check** **Example:** `Router(config-router)# neighbor 10.20.20.20 disable-connected-check`	Allows peering between loopbacks. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group.
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Router(config-router)# neighbor 10.20.20.20 update-source Loopback 0`	Allows BGP sessions 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. **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 8	**address-family ipv4** [**unicast**] **vrf** *vrf-name* **Example:** `Router(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 9	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Router(config-router-af)# neighbor 10.20.20.20 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring router. • The *peer-group-name* argument is the name of 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.

	Command or Action	Purpose
Step 10	**neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended** **Example:** `Router(config-router-af)# neighbor 10.20.20.20 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring router. • The *peer-group-name* argument is the name of 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.
Step 11	**end** **Example:** `Router(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 MPLS 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:** `Router> 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:** `Router# show mpls forwarding-table`	Displays the contents of the MPLS LFIB. • Enter a keyword or argument, if desired.

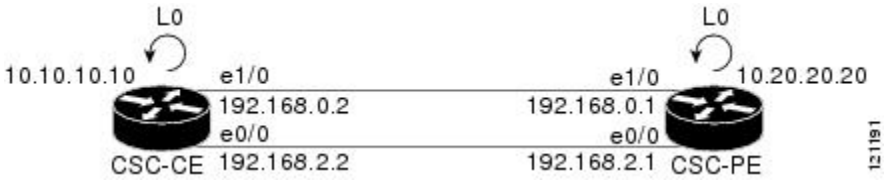
	Command or Action	Purpose
Step 3	**disable** **Example:** `Router# disable`	Exits to user EXEC mode.

Configuring Directly Connected Loopback Peering on MPLS VPN Carrier Supporting Carrier

The following sections explain how to load balance CSC traffic by peering loopback interfaces of directly connected CSC-PE and CSC-CE routers:

The figure below shows the loopback configuration for directly connected CSC-PE and CSC-CE routers. 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 Routers



- Configuring Loopback Interface Addresses on CSC-PE Routers, page 249
- Configuring Loopback Interface Addresses for CSC-CE Routers, page 251
- Configuring 32 Static Routes to the eBGP Neighbor Loopback on the CSC-PE Router, page 252
- Configuring 32 Static Routes to the eBGP Neighbor Loopback on the CSC-CE Router, page 253
- Configuring Forwarding on CSC-PE Interfaces That Connect to the CSC-CE Loopback, page 254
- Configuring Forwarding on CSC-CE Interfaces That Connect to the CSC-PE Loopback, page 256
- Configuring an eBGP Session Between the CSC-PE Router and the CSC-CE Loopback, page 257
- Configuring an eBGP Session Between the CSC-CE Router and the CSC-PE Loopback, page 260
- Verifying That Load Sharing Occurs Between Loopbacks, page 262

Configuring Loopback Interface Addresses on CSC-PE Routers

Perform this task to configure loopback interface addresses on the CSC-PE router.

Note Configuration of a loopback interface address on the CSC-PE router requires the enabling of a VRF. The CSC-CE router loopback interface does not require the enabling a of VRF.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface number* **Example:** `Router(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:** `Router(config-if)# ip vrf forwarding vpn1`	Associates a VRF with the specified interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.
Step 5	**ip addres** *ip-address mask* [**secondary**] **Example:** `Router(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.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Configuring Loopback Interface Addresses for CSC-CE Routers

Perform this task to configure 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface-number* **Example:** `Router(config)# interface loopback 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.

	Command or Action	Purpose
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.10.10.10 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.

Configuring 32 Static Routes to the eBGP Neighbor Loopback on the CSC-PE Router

Perform the following task to configure /32 static routes to the eBGP neighbor loopback on the CSC-PE router.

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

	Command or Action	Purpose
Step 3	**ip route vrf** *vrf-name prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**global**] [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Router(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 VRF. • 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:** `Router(config)# end`	Exits to privileged EXEC mode.

Configuring 32 Static Routes to the eBGP Neighbor Loopback on the CSC-CE Router

Perform the following task to configure /32 static routes to the eBGP neighbor loopback for the CSC-CE router.

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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent**] [**tag** *tag*] **Example:** `Router(config)# ip route 10.20.20.20 255.255.255.255 Ethernet 1/0 172.16.0.1`	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:** `Router(config)# end`	Exits to privileged EXEC mode.

Configuring Forwarding on CSC-PE Interfaces That Connect to the CSC-CE Loopback

Perform this task to configure 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:** `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 slot/port* **Example:** `Router(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:** `Router(config-if)# 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 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(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:** `Router(config-if)# mpls bgp forwarding`	Configures BGP to enable MPLS forwarding on connecting interfaces.
Step 7	**exit** **Example:** `Router(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:** `Router(config)# end`	Exits to privileged EXEC mode.

Configuring Forwarding on CSC-CE Interfaces That Connect to the CSC-PE Loopback

Perform this task to configure 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:** `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** *typeslot/port* **Example:** `Router(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:** `Router(config-if)# mpls bgp forwarding`	Configures BGP to enable MPLS forwarding on connecting interfaces.
Step 5	**exit** **Example:** `Router(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:** `Router(config)# end`	Exits to privileged EXEC mode.

Configuring an eBGP Session Between the CSC-PE Router and the CSC-CE Loopback

Perform this task to configure an eBGP session between the CSC-PE router 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:** `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 the BGP routing process. • 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.
Step 4	**bgp log-neighbor-changes** **Example:** `Router(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.

	Command or Action	Purpose
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.10.10.10 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group. • The *as-number* argument is the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check** **Example:** `Router(config-router)# neighbor 10.10.10.10 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:** `Router(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:** `Router(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 VRF to associate with submode commands.
Step 9	**ip vrf forwarding** *vrf-name* **Example:** `Router(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:** `Router(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 router. • 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:** `Router(config-router-af)# neighbor 10.10.10.10 send-label`	Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router. • The *ip-address*argument is the IP address of the neighboring router.
Step 12	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.

Configuring an eBGP Session Between the CSC-CE Router and the CSC-PE Loopback

Perform this task to configure an eBGP session between the CSC-CE router 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:** `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 the BGP routing process. • 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.
Step 4	**bgp log-neighbor-changes** **Example:** `Router(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:** `Router(config-router)# neighbor 10.20.20.20 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group. • The *as-number* argument is the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **disable-connected-check** **Example:** `Router(config-router)# neighbor 10.20.20.20 disable-connected-check`	Allows peering between loopbacks. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group.

	Command or Action	Purpose
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Router(config-router)# neighbor 10.20.20.20 update-source Loopback 0`	Allows BGP sessions 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:** `Router(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 VRF to associate with submode commands.
Step 9	**neighbor** {*ip-address* | *peer-group-name*| *ipv6-address*] **activate** **Example:** `Router(config-router-af)# neighbor 10.20.20.20 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring router. • The *peer-group-name* argument is the name of 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:** `Router(config-router-af)# neighbor 10.20.20.20 send-label`	Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router. • The *ip-address* argument is the IP address of the neighboring router.
Step 11	**end** **Example:** `Router(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 MPLS 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:** `Router> 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:** `Router# show mpls forwarding-table`	Displays the contents of the MPLS LFIB.
Step 3	**disable** **Example:** `Router# disable`	Exits to user EXEC mode.

Configuration Examples for Load Sharing MPLS VPN Traffic

Configuring a Router to Select eBGP or iBGP Paths as Multipaths Example

The following example configures a router in address family configuration mode to select six eBGP or iBGP paths as multipaths:

```
Router(config)# router bgp 100
Router(config-router)# address-family ipv4 vrf try
Router(config-router-af)# maximum-paths eibgp 6
Router(config-router-af)# end
```

Configuring a 32 Static Route from an ASBR to the Loopback Address of Another ASBR Examples

The following example configures a /32 static route from ASBR1 to the loopback address of ASBR2:

```
Router# configure terminal
Router(config)# ip route 10.20.20.20 255.255.255.255 e1/0 168.192.0.1
Router(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:

```
Router# configure terminal
Router(config)# ip route vrf vpn1 10.10.10.10 255.255.255.255 e1/0 168.192.0.2
Router(config)# ip route vrf vpn1 10.10.10.10 255.255.255.255 e0/0 168.192.2.2
```

Configuring BGP MPLS Forwarding on the Interfaces Connecting ASBRs Example

The following example configures BGP/MPLS forwarding on the interfaces connecting ASBR2 with ASBR1:

```
Router# configure terminal
Router(config)# interface ethernet 1/0
Router(config-if)# ip vrf forwarding vpn1
Router(config-if)# ip address 168.192.0.1 255.255.255.255
Router(config-if)# mpls bgp forwarding
Router(config-if)# exit
Router(config)# interface ethernet 0/0
Router(config-if)# ip vrf forwarding vpn1
Router(config-if)# ip address 168.192.2.1 255.255.255.255
Router(config-if)# mpls bgp forwarding
Router(config-if)# exit
```

Configuring VPNv4 Sessions on an ASBR Example

The following example configures VPNv4 sessions on ASBR2:

```
Router# configure terminal
Router(config)# router bgp 200
Router(config-router)# bgp log-neighbor-changes
Router(config-router)# neighbor 10.10.10.10 remote-as 100
Router(config-router)# neighbor 10.10.10.10 disable-connected-check
Router(config-router)# neighbor 10.10.10.10 update-source Loopback0
!
Router(config-router)# address-family vpnv4
Router(config-router-af)# neighbor 10.10.10.10 activate
Router(config-router-af)# neighbor 10.10.10.10 send-community extended
Router(config-router-af)# end
```

Verifying VPN NLRI for a Specified Network Example

If you enter the **all** keyword with the **show ip bgp vpnv4** command, the output displays information about all VPN network layer reachability information (NLRI) for a specified network:

```
Router# show ip bgp vpnv4 all 10.22.22.0
BGP routing table entry for 10:1:22.22.22.0/24, version 19
Paths:(5 available, best #5)
Multipath: eiBGP
  Advertised to non peer-group peers:
  10.0.0.2 10.0.0.3 10.0.0.4 10.0.0.5
  22
    10.0.0.2 (metric 20) from 10.0.0.4 (10.0.0.4)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:0x0:0:0 RT:100:1 0x0:0:0
      Originator:10.0.0.2, Cluster list:10.0.0.4
  22
    10.0.0.2 (metric 20) from 10.0.0.5 (10.0.0.5)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:0x0:0:0 RT:100:1 0x0:0:0
      Originator:10.0.0.2, Cluster list:10.0.0.5
  22
    10.0.0.2 (metric 20) from 10.0.0.2 (10.0.0.2)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:RT:100:1 0x0:0:0
  22
    10.0.0.2 (metric 20) from 10.0.0.3 (10.0.0.3)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:0x0:0:0 RT:100:1 0x0:0:0
      Originator:10.0.0.2, Cluster list:10.0.0.3
  22
    10.1.1.12 from 10.1.1.12 (10.22.22.12)
      Origin IGP, metric 0, localpref 100, valid, external, multipath, best
      Extended Community:RT:100:1
```

Additional References

Related Documents

Related Topic	Document Title
MPLS	*MPLS: Layer 3 VPNs: Inter-AS and CSC Configuration Guide*, *MPLS VPN Carrier Supporting Carrier with BGP*
BGP	*Cisco IOS IP Routing: BGP Configuration Guide*, *Configuring BGP*

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 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 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 Load Sharing MPLS VPN Traffic

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 Load Sharing MPLS VPN Traffic

Feature Name	Releases	Feature Configuration Information
MPLS VPN--Load Balancing Support for Inter-AS and CSC VPNs	12.0(29)S 12.4(20)T	This 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.
BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN	12.2(4)T 12.2(14)S 12.0(24)S	This feature allows multihomed autonomous systems and PE routers to be configured to distribute traffic across both external BGP (eBGP) and internal BGP (iBGP) paths.
iBGP Multipath Load Sharing	12.2(2)T 12.2(14)S	This feature enables the BGP speaking router to select multiple iBGP paths as the best paths to a destination.
eBGP Multipath	12.0(27)S	This 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.