MPLS VPN—Interautonomous System Support

First Published: October 3, 2000
Last Updated: June 29, 2007

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" section on page 62.

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.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

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" section on page 13 build from those configuration tasks.

Perform (as appropriate to the existing network configuration) the following tasks as described in the section "Configuring MPLS Virtual Private Networks"of the *Cisco IOS Multiprotocol Label Switching Configuration Guide,* Release 12.4.

- 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 Table 1.

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 1-Port OC-12 POS 8-Port OC-3 POS 16-Port OC-3 POS 4-Port OC-12 POS 1-Port OC-48 POS 4-Port OC-3 POS ISE 8-Port OC-3 POS ISE 16-Port OC-3 POS ISE 4 Port OC-12 POS ISE 1-Port OC-48 POS ISE	12.0(16)ST 12.0(17)ST 12.0(22)S
Electrical Interface	6-Port DS3 12-Port DS3 6-Port E3 12-Port E3	12.0(21ST 12.0(22)S
Ethernet	3-Port GbE 1-Port 10-GbE Modular GbE/FE	12.0(23)S 12.0(24)S

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

Type	Line Cards	Cisco IOS Release Added
ATM	4-Port OC-3 ATM 1-Port OC12 ATM 4-Port OC-12 ATM 8-Port OC-3 ATM	12.0(16)ST 12.0(17)ST 12.0(23)S
Channelized Interface	2-Port CHOC-3 6-Port Ch T3 (DS1) 1-Port CHOC-12 (DS3) 1-Port CHOC-12 (OC-3) 4-Port CHOC-12 ISE 1-Port CHOC-48 ISE	12.0(22)S

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

Before configuring this feature, you should understand the following concepts:

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 Exchanging VPN-IPv4 Addresses

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 Exchanging VPN-IPv4 Addresses" 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 Exchanging VPN-IPv4 Addresses

This section contains the following topics:

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

Figure 1 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

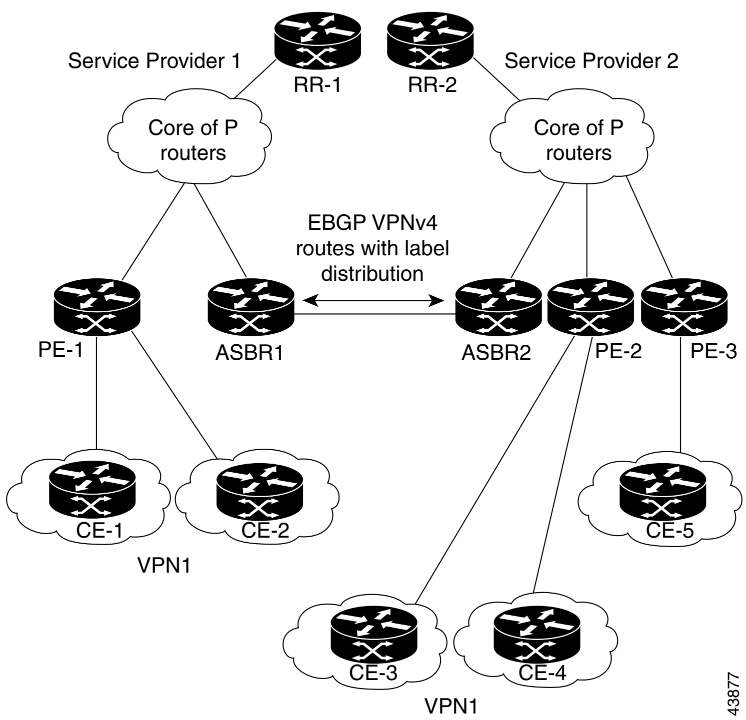


Table 2 describes the process to transmit information in an Inter-As configuration with ASBRs exchanging VPN-IPv4 addresses.

Table 2 Information Transmission Process in an Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

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 Exchanging VPN-IPv4 Addresses

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.

Figure 2 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

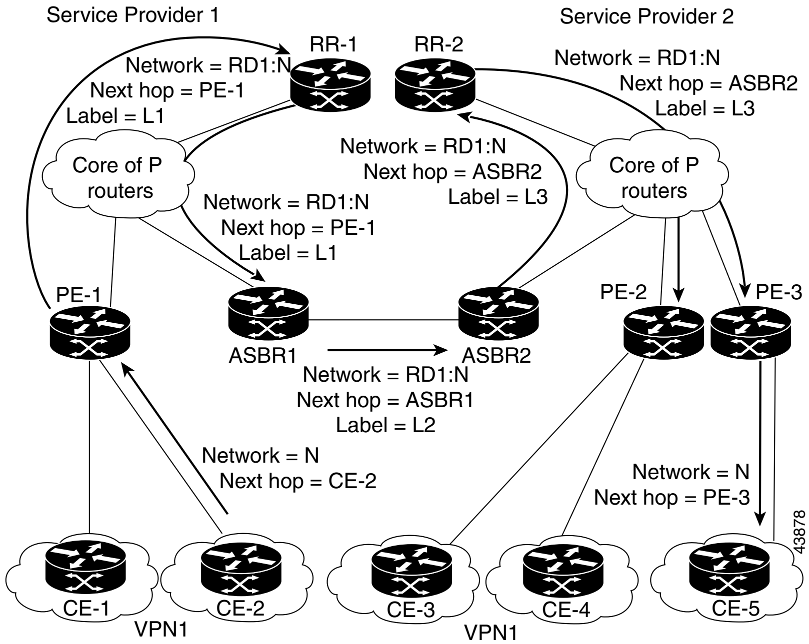
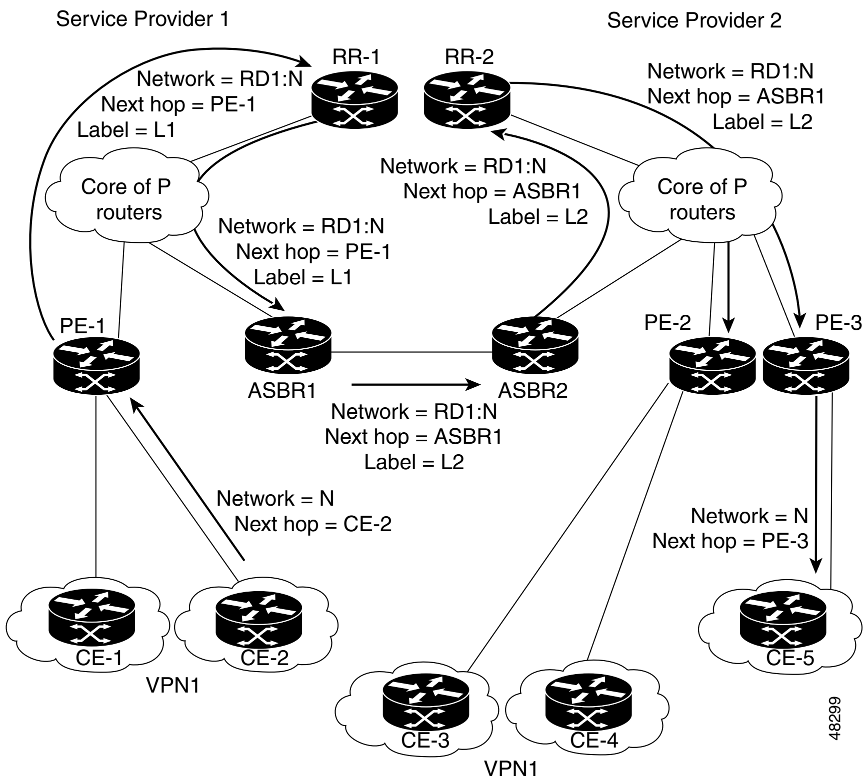


Figure 3 illustrates the exchange of VPN route and label information between autonomous systems. The only difference is that ASBR2 is configured with the **redistribute connected** command, which propagates the host routes to all PEs. The **redistribute connected** command is necessary because ASBR2 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 Exchanging VPN-IPv4 Addresses

Figure 4 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***

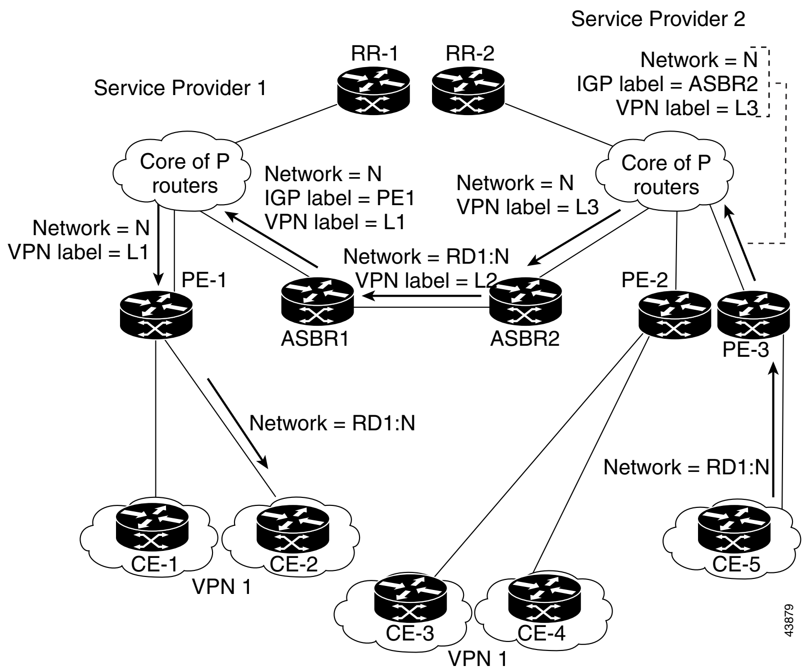
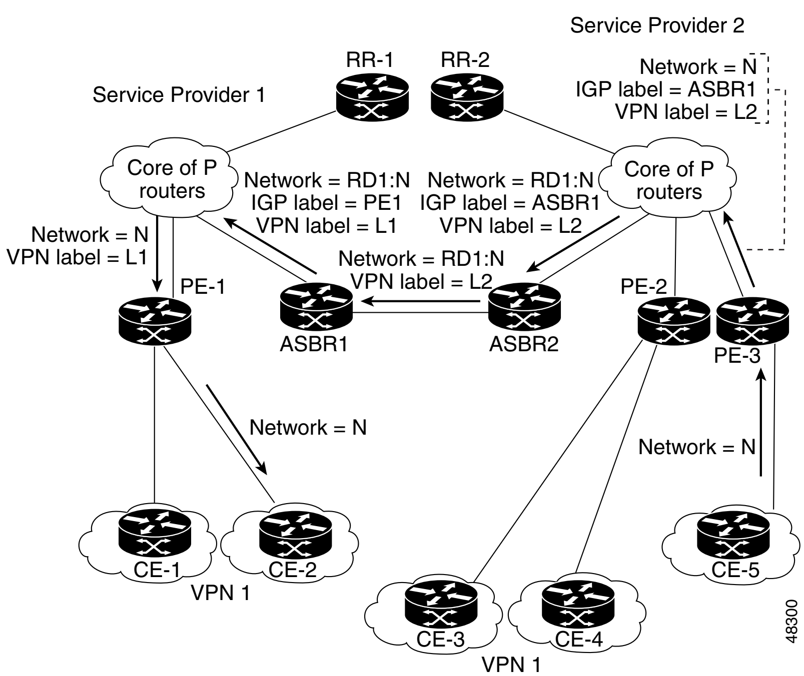


Figure 5 shows the same packet forwarding method, except the eBGP router (ASBR1) forwards the packet without reassigning it a new label.

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 Exchanging VPN-IPv4 Addresses

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.

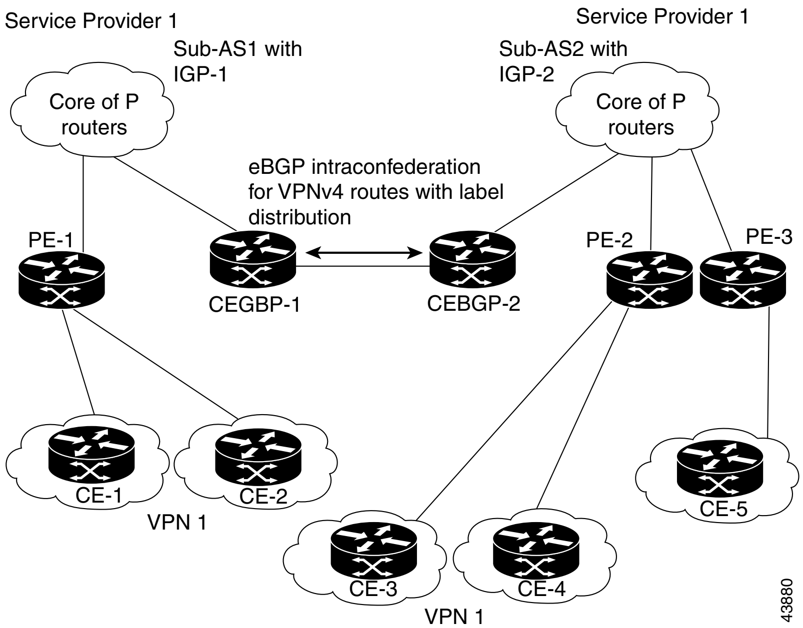
Note Figure 2 and Figure 3 illustrate how two autonomous systems exchange routes and forward packets. Subautonomous systems in a confederation use a similar method of exchanging routes and forwarding packets.

Figure 6 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 Exchanging VPN-IPv4 Addresses

Before the MPLS VPN—Multipath Support for Inter-AS VPNs 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:

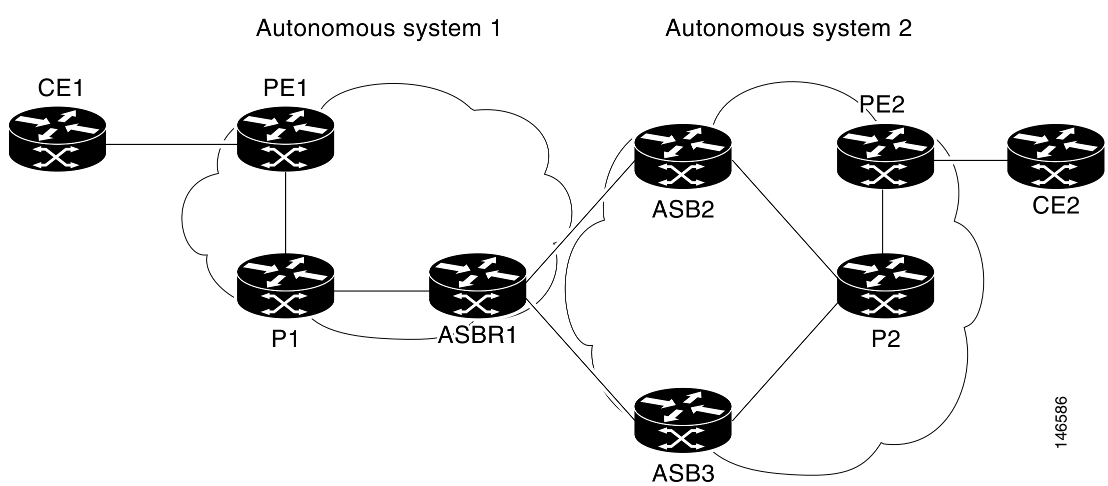
- Cisco IOS Release 12.0S-based software: 8 paths
- Cisco IOS Release 12.3T-based software: 16 paths
- Cisco IOS Release 12.2S-based software: 32 paths

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.

Figure 7 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, page 13 (required)
- Configuring eBGP Routing to Exchange MPLS VPN Routes Between Subautonomous Systems in a Confederation, page 24 (required)
- Verifying Inter-AS for ASBRs Exchanging MPLS VPN-IPv4 Addresses, page 26 (optional)
- Configuring eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs Exchanging VPN-IPv4 Routes, page 28 (optional)
- Verifying eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs, page 33 (optional)

Configuring an eBGP ASBR to Exchange MPLS VPN-IPv4 Addresses

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

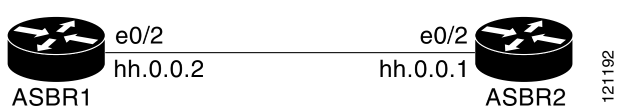
- Configuring Peering with Directly Connected Interfaces Between ASBRs, page 13 (optional)
- Configuring Peering of the Loopback Interface of Directly Connected ASBRs, page 15 (optional)

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.

Figure 8 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.
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.

	Command or Action	Purpose
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

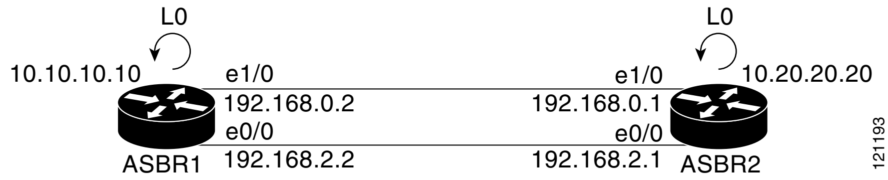
This functionality is provided with the release of the MPLS VPN—Interautonomous System Support feature on Cisco IOS Release 12.0(29)S and later releases. An eBGP session configured between loopbacks of directly connected ASBRs allows load sharing between loopback addresses.

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

- Configuring Loopback Interface Addresses for Directly Connected ASBRs, page 16 (required)
- Configuring /32 Static Routes to the eBGP Neighbor Loopback, page 17 (required)
- Configuring Forwarding on the Directly Connected Interfaces, page 19 (required)
- Configuring an eBGP Session Between the Loopbacks, page 21 (required)

Figure 9 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 Figure 9).

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.

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 /32 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 0/0 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 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 1/0 192.168.0.1`	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 /32 Static Routes to the eBGP Neighbor Loopback" 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.
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.

	Command or Action	Purpose
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.

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.

	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 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 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.

	Command or Action	Purpose
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.
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.
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.

	Command or Action	Purpose
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.
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:

```
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:

```
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:

```
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:

```
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:

```
Router# exit
Router>
```

Configuring eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs Exchanging VPN-IPv4 Routes

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.

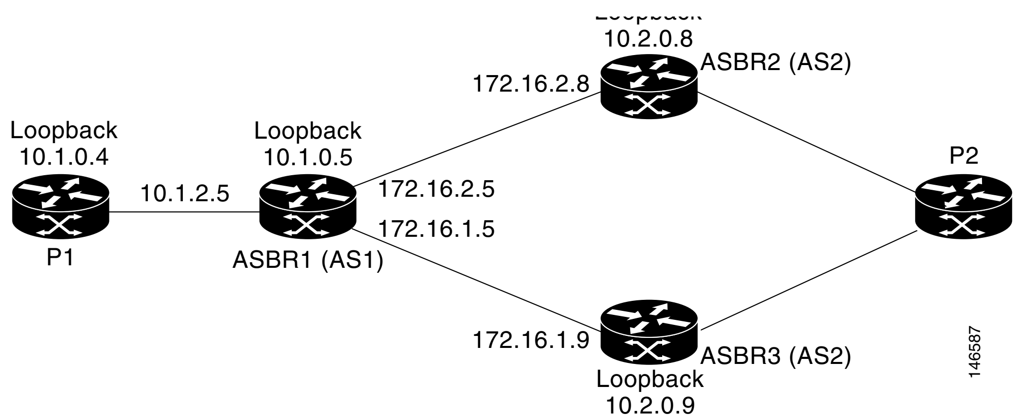
Restrictions for eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs Exchanging VPN-IPv4 Routes

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.

Figure 10 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 Figure 10, 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 Figure 10 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" section on page 33.

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.
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.

	Command or Action	Purpose
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-type interface-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.	—
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.

	Command or Action	Purpose
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 Exchanging VPN-IPv4 Addresses" section on 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

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 Figure 10 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 all** [*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:

```
Router> enable
Router#
```

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

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

```
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:

```
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:

```
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:

```
Router# exit
Router>
```

Configuration Examples for MPLS VPN—Interautonomous System Support

This section provides the following configuration examples:

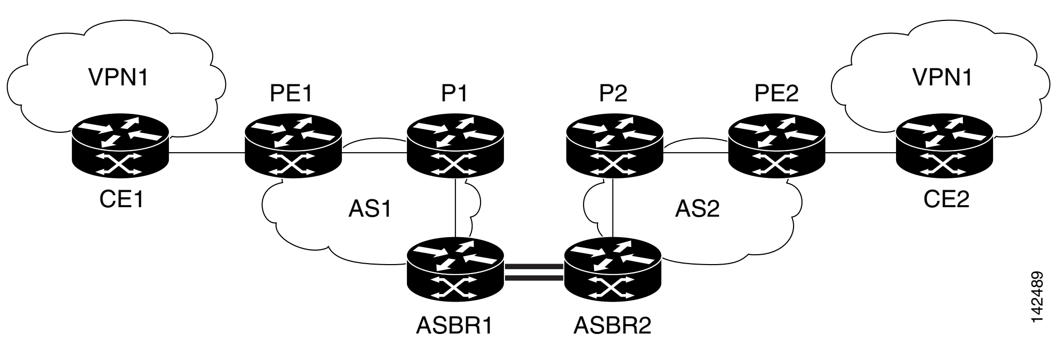
- Configuring Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses: Example, page 35
- Configuring Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses in a Confederation: Example, page 42
- Configuring eBGP Multipath Load Sharing for MPLS VPN Inter-AS ASBRs Exchanging VPN-IPv4 Routes: Example, page 48

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

The network topology in Figure 11 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 Figure 11):

```
!
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 Figure 11):

```
!
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 Figure 11):

```
!
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 Figure 11):

```
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 Figure 11):

```
!
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 Figure 11):

```
!
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 Figure 11):

```
!
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 Figure 11):

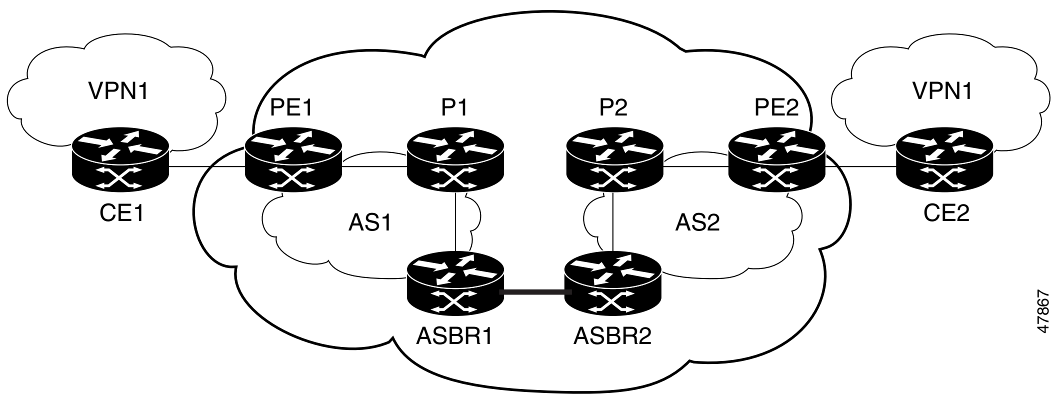
```
!
hostname CE2
!
interface Loopback 0
 ip address 192.168.0.2 255.255.255.255
!
interface Serial 2/0
 description Link to PE2
 ip address 192.168.2.1 255.255.255.0
 no fair-queue
 serial restart-delay 0
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.255.255 area 0
!
end
```

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

The network topology in Figure 12 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

The following example shows how to configure CE1 in VPN1 in an Inter-AS confederation (see Figure 12):

```
!
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 Figure 12):

```
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 Figure 12):

```
!
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 Figure 12):

```
!
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 Figure 12):

```
!
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 Figure 12):

```
!
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 Figure 12):

```
!
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 Figure 12):

```
!
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 Exchanging VPN-IPv4 Routes: Example

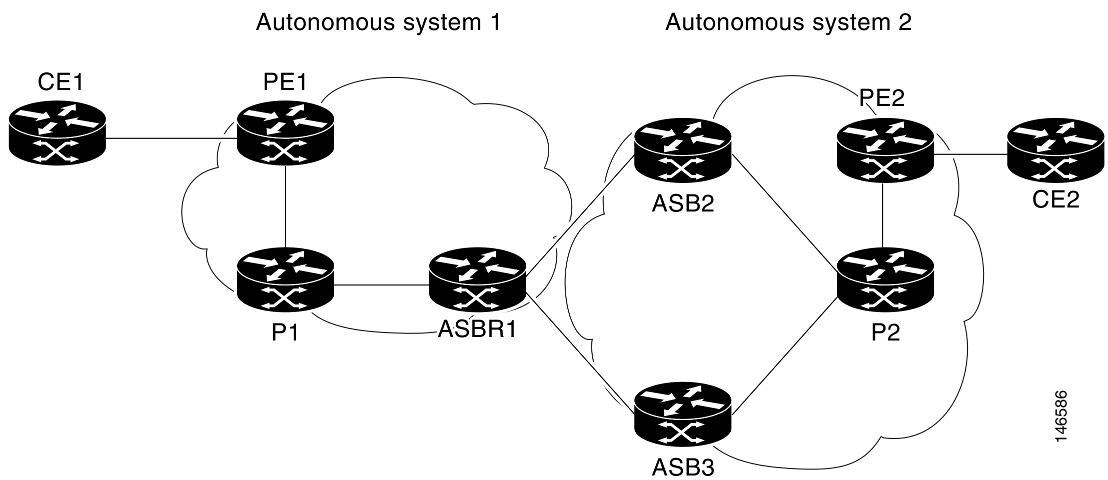
This section includes examples that show how to configure eBGP multipath load sharing for MPLS VPN Inter-AS ASBRS that exchange VPN-IPv4 routes. These configurations support the MPLS VPN—Multipath Support for Inter-AS VPNs feature.

The network topology in Figure 13 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—Multipath Support for Inter-AS VPNs feature (see Figure 13):

```
!
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—Multipath Support for Inter-AS VPNs feature (see Figure 13):

```
!
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—Multipath Support for Inter-AS VPNs feature (see Figure 13):

```
!
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—Multipath Support for Inter-AS VPNs feature (see Figure 13):

```
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—Multipath Support for Inter-AS VPNs feature (see Figure 13):

```
!
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—Multipath Support for Inter-AS VPNs feature (see Figure 13):

```
!
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—Multipath Support for Inter-AS VPNs feature (see Figure 13):

```
!
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—Multipath Support for Inter-AS VPNs feature (see Figure 13):

```
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—Multipath Support for Inter-AS VPNs feature (see Figure 13):

```
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

The following sections provide references related to configuring MPLS VPN—Inter-AS support.

Related Documents

Related Topic	Document Title
Configuration tasks for basic MPLS VPNs	*MPLS Virtual Private Networks (VPNs)*
Configuration tasks for enhanced VPN traffic management including BGP commands	*MPLS Virtual Private Network Enhancements*
Configuration tasks for MPLS VPN Inter-AS system exchanging IPv4 routes and MPLS labels	*MPLS VPN—Inter-AS—IPv4 BGP Label Distribution*
Configuration tasks for MPLS VPN Carrier Supporting Carrier	*MPLS VPN—Carrier Supporting Carrier*
Information about monitoring MPLS VPNs with MIBs	*MPLS VPN—SNMP MIB Support*

Related Topic	Document Title
Configuration tasks for assigning an ID number to a VPN	*MPLS VPN ID*
Configuration tasks for MPLS and MPLS applications	*Cisco IOS Multiprotocol Label Switching Configuration Guide,* Release 12.4

Standards

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

MIBs

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

RFCs

RFC	Title
RFC 1164	*Application of the Border Gateway Protocol in the Internet*
RFC 1700	*Assigned Numbers*
RFC 1771	*A Border Gateway Protocol 4*
RFC 1965	*Autonomous System Confederation for BGP*
RFC 1966	*BGP Route Reflection: An Alternative to Full Mesh iBGP*
RFC 2547	*BGP/MPLS VPNs*
RFC 2842	*Capabilities Advertisement with BGP-4*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 3107	*Carrying Label Information in BGP-4*

Technical Assistance

Description	Link
The Cisco Support 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

Command Reference

This section documents only commands that are new or modified.

- **bgp default route-target filter**
- **mpls bgp forwarding**

bgp default route-target filter

To enable automatic Border Gateway Protocol (BGP) default route-target community filtering, use the **bgp default route-target filter** command in router configuration mode. To disable automatic BGP route-target community filtering, use the **no** form of this command.

bgp default route-target filter

no bgp default route-target filter

Syntax Description

This command has no arguments or keywords.

Command Default

Automatic BGP default route-target community filtering is enabled.

Command Modes

Router configuration (config-router)

Command History

Release	Modification
12.1(5)T	This command was introduced.
12.0(16)ST	This command was integrated into Cisco IOS Release 12.0(16)ST.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

Use the **bgp default route-target filter** command to control the distribution of Virtual Private Network (VPN) routing information through the list of VPN route-target communities.

When you use the **no** form of this command, 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 autonomous system border edge router or as a customer edge (CE) BGP border edge router.

If you configure the router for BGP route-target community filtering, all received exterior BGP (eBGP) VPN-IPv4 routes are discarded when those routes do not contain a route-target community value that matches the import list of any configured VPN routing and forwarding (VRFs) instances. This is the desired behavior for a router configured as a provider edge (PE) router.

Note: This command is automatically disabled if a PE router is configured as a client of a common VPN-IPv4 route reflector in the autonomous system.

Examples

In the following example, BGP route-target filtering is disabled for autonomous system 120:

```
Router(config)# router bgp 120
Router(config-router)# no bgp default route-target filter
```

Related Commands

Command	Description
show mpls forwarding-table	Displays the contents of the LFIB.

mpls bgp forwarding

To enable an interface to receive Multiprotocol Label Switching (MPLS) packets when the signaling of MPLS labels is through the use of the Border Gateway Protocol (BGP), use the **mpls bgp forwarding** command in interface configuration mode. To disable an interface from receiving MPLS packets when the signaling of MPLS labels is through the use of the BGP, use the **no** form of this command.

mpls bgp forwarding

no mpls bgp forwarding

Syntax Description

This command has no arguments or keywords.

Command Default

MPLS forwarding by BGP is not enabled.

Command Modes

Interface configuration (config-if)

Command History

Release	Modification
12.0(29)S	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

Use the **mpls bgp forwarding** command when you want to enable MPLS forwarding on directly connected loopback interfaces. This command is automatically generated by BGP for directly connected nonloopback neighbors.

Examples

The following example shows how to configure BGP to enable MPLS forwarding on a directly connected loopback interface, Ethernet 0/0:

```
Router(config)# interface ethernet 0/0
Router(config-if)# mpls forwarding
```

Related Commands

Command	Description
ip vrf forwarding	Associates a VRF with an interface or subinterface.

Feature Information for MPLS VPN—Interautonomous System Support

Table 3 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

For information on a feature in this technology that is not documented here, see the "MPLS Layer 3 VPN Features Roadmap" module.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 3 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release. Unless noted otherwise, subsequent releases of that Cisco IOS software release also support that feature.

Table 3 Feature Information for MPLS 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 Table 1 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 Table 1 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 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.

Table 3 Feature Information for MPLS VPN—Interautonomous System Support (continued)

Feature Name	Releases	Feature Information
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.
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.