MPLS VPN—Interautonomous System Support

This document is still under development and subject to change.

Feature History

Release	Modification
12.1(5)T	This document was introduced.
12.0(16)ST	This feature was integrated into Cisco IOS Release 12.0(16)ST. Support for the Cisco 12000 Series Four-Port OC-3c/STM-1c ATM Line Card (4-Port OC-3 ATM) and the Cisco 12000 Series Four-Port OC-3c/STM-1c POS/SDH Line Card (4-Port OC-3 POS) was added.
12.0(17)ST	This feature was integrated into Cisco IOS Release 12.0(17)ST. Support for the Cisco 12000 series was added (See Table 1 for the Cisco 12000 series line cards supported).
12.0(22)S	This feature was integrated into Cisco IOS Release 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).
12.0(23)S	This feature was integrated into Cisco IOS Release 12.0(23)S. Support was added for the Cisco 12000 Series Eight-Port OC-3c/STM-1c ATM Line Card (8-Port OC-3 ATM) and the Cisco 12000 Series Three-Port Gigabit Ethernet Line Card (3-Port GbE).
12.0(24)S	This feature was integrated into Cisco IOS Release 12.0(24)S. Support was added for the Cisco 12000 Series One-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).
12.0(29)S	Support was added for EBGP sessions between loopbacks of directly connected MPLS-enabled routers to provide for loadsharing between neighbors.

CISCO SYSTEMS

EARLY FIELD TEST DRAFT - CISCO CONFIDENTIAL

This feature module explains how to provide Multiprotocol Label Switching Virtual Private Network (MPLS VPN) services that can span multiple autonomous systems (ASs) and VPN service providers. This document includes the following sections:

Feature Overview

The MPLS VPN—Interautonomous System Support feature allows an MPLS VPN to span service providers and autonomous systems.

As VPNs grow, their requirements expand. In some cases, VPNs need to reside on different autonomous systems in different geographic areas. (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.) 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 seamless integration of autonomous systems and service providers. Separate autonomous systems from different service providers can communicate by exchanging IPv4 Network Layer Reachability Information (NLRI) in the form of VPN-IPv4 addresses. The autonomous systems' border edge routers use Exterior Border Gateway Protocol (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.

An MPLS VPN with interautonomous system support allows a service provider to provide to customers scalable Layer 3 VPN services, such as web hosting, application hosting, interactive learning, electronic commerce, and telephony service. A VPN service provider supplies a secure, IP-based network that shares resources on one or more physical networks.

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 section "Routing Between Autonomous Systems" for more information.

Interautonomous system configurations supported in an MPLS VPN can include:

EARLY FIELD TEST DRAFT - CISCO CONFIDENTIAL

- 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.

Benefits

The MPLS VPN—Interautonomous System Support feature 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. Previous MPLS VPN could only traverse a single BGP autonomous system service provider backbone. This feature allows multiple autonomous systems to form a continuous (and seamless) network between customer sites of a service provider.

Allows a VPN to Exist in Different Areas

A service provider can create a VPN in different geographic areas. Having all VPN traffic flow through one point (between the areas) allows for better rate control of network traffic between the areas.

Allows Confederations to Optimize IBGP Meshing

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.

Routing Between Autonomous Systems

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 Autonomous Systems
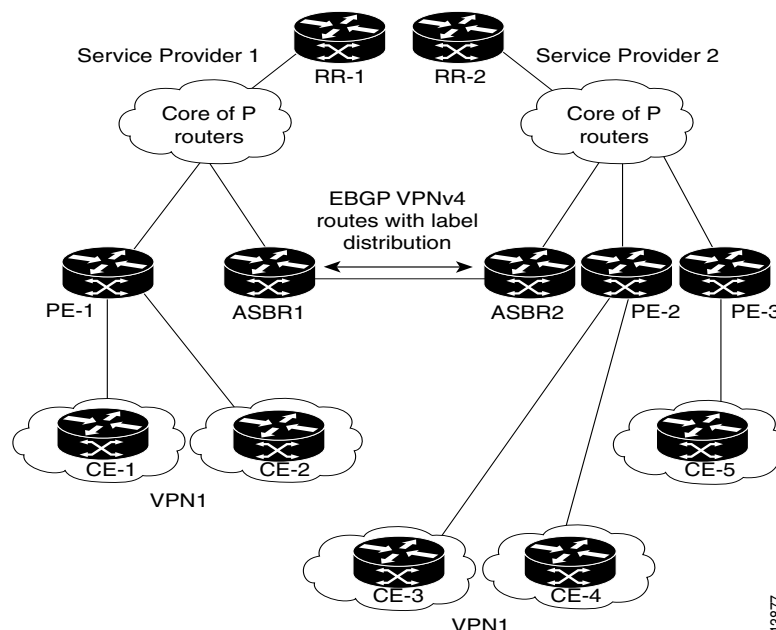
This configuration uses the following process to transmit information:

Step 1   The provider edge router (PE-1) assigns a label for a route before distributing that route. The PE router uses the multiprotocol extensions of a Border Gateway Protocol (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.

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

Step 3   The 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.)

Step 4   The 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 on.

- 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.

Exchanging VPN Routing Information

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 includes:
    - 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 to make 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.
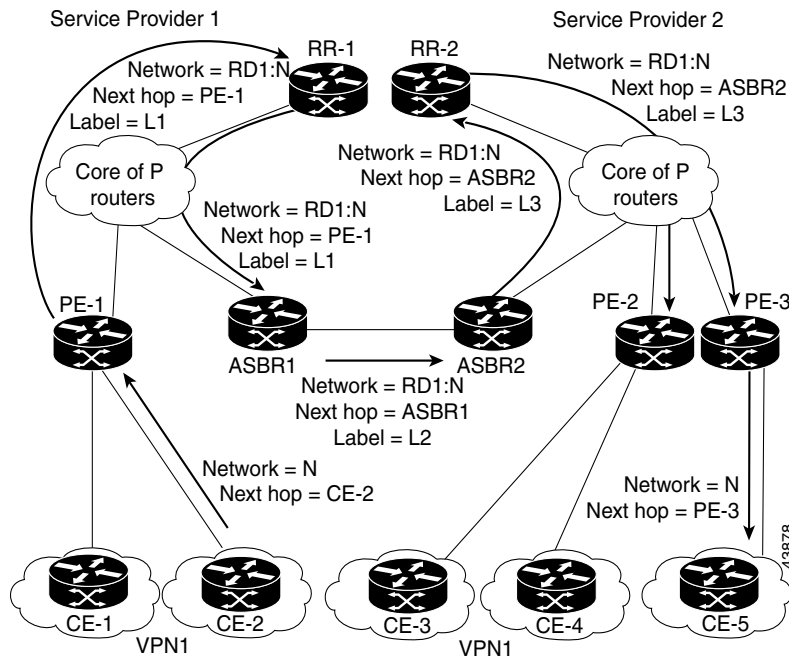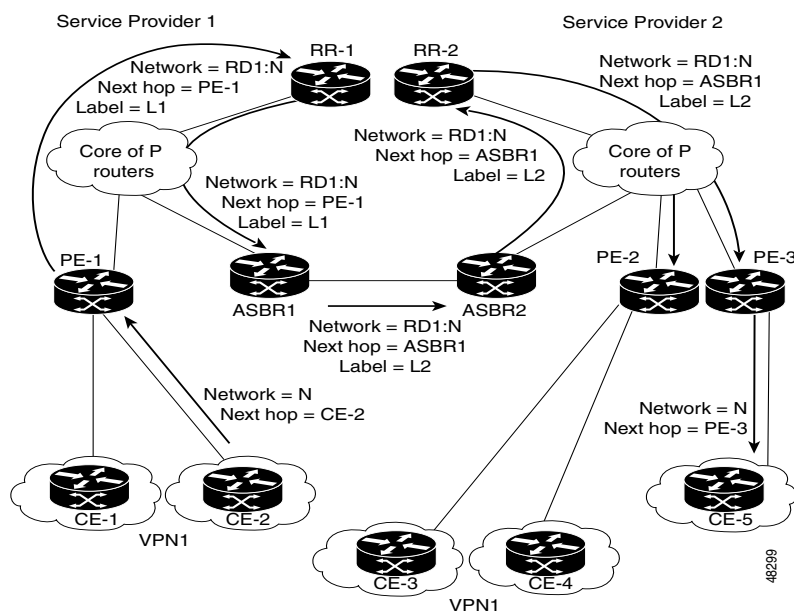
EARLY FIELD TEST DRAFT - CISCO CONFIDENTIAL

Figure 2   *Exchanging Routes and Labels Between Autonomous Systems in an Interprovider VPN Network*

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 Between Autonomous Systems in an Interprovider VPN Network with the redistributed connected Command*

Packet Forwarding

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   Forwarding Packets Between Autonomous Systems in an Interprovider VPN Network
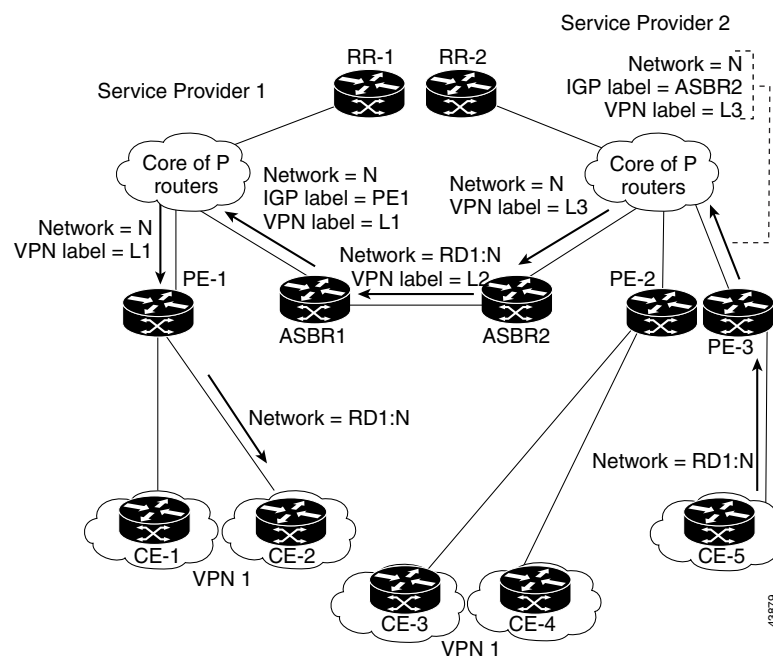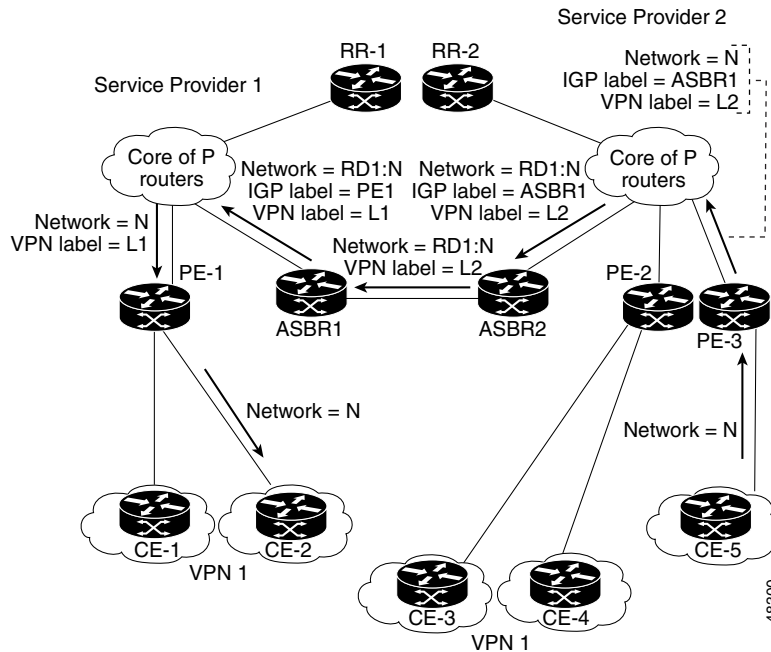

Figure 5 illustrates shows the same packet forwarding method, except the EBGP router (ASBR1) forwards the packet without reassigning it a new label.

Figure 5   Forwarding Packets Between Autonomous Systems in an Interprovider VPN Network Without a New Label Assignment

Routing Between Subautonomous Systems in a Confederation

A VPN can span service providers running in separate autonomous systems or between multiple subautonomous systems that have been grouped together to form a confederation.

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.

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 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.

EARLY FIELD TEST DRAFT - CISCO CONFIDENTIAL

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
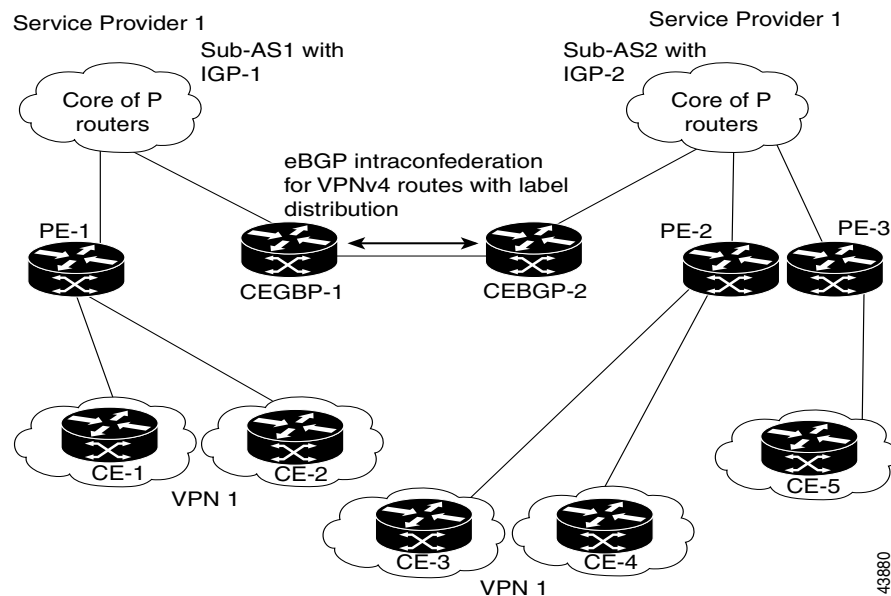
In this confederation configuration:

- 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 an 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.

Restrictions

A VPN-IPv4 EBGP session must be configured between directly connected ASBRs. Multihop VPN-IPv4 EBGP is not supported.

Related Features and Technologies

The MPLS VPN—Interautonomous System Support feature is used with the VPN capabilities of MPLS. MPLS VPNs were introduced in Cisco IOS Release 12.0(5)T.

Related Documents

- *MPLS Virtual Private Networks (VPNs)*
- *MPLS Virtual Private Network Enhancements*
- *Cisco IOS Switching Services Configuration Guide (Release 12.2), Multiprotocol Label Switching*

Supported Platforms

The following router platforms are supported at the service provider edge:

- Cisco 7200 series
- Cisco 7500 series
- Cisco 10000 series edge services routers (ESRs)
- Cisco 10720 Internet Routers
- Cisco 12000 series

Table 1 lists the Cisco 12000 series line card support added for Cisco IOS Releases.

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	12.0(16)ST
	8-Port OC-3 POS 16-Port OC-3 POS 4-Port OC-12 POS 1-Port OC-48 POS	12.0(17)ST
	4-Port OC-3 POS ISE 8-Port OC-3 POS ISE 16 x OC-3 POS ISE 4 Port OC-12 POS ISE 1-Port OC-48 POS ISE	12.0(22)S
Electrical Interface	6- Port DS3 12- Port DS3	12.0(21ST
	6-Port E3 12-Port E3	12.0(22)S

EARLY FIELD TEST DRAFT - CISCO CONFIDENTIAL

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

Type	Line Cards	Cisco IOS Release Added
Ethernet	3-Port GbE 1-Port 10-GbE Modular GbE/FE	12.0(23)S 12.0(24)S
Asynchronous Transfer Mode (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

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To obtain updated information about platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features that releases have in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

- RFC 1771, *A Border Gateway Protocol 4*
- RFC 1965, *Autonomous System Confederation for BGP*
- RFC 1164, *Application of the Border Gateway Protocol in the Internet*
- RFC 2842, *Capabilities Advertisement with BGP-4*
- RFC 2283, *Multiprotocol Extensions for BGP-4*
- RFC 2547, *BGP/MPLS VPNs*

Standards

- Internet draft draft-ramachandra-bgp-ext-communities-08.txt, *BGP Extended Community Attribute*

Prerequisites

The network must be properly configured for MPLS VPN operation before you configure interautonomous systems. Refer to the following documents for MPLS VPN network configuration details:

- *MPLS Virtual Private Networks (VPNs)*
- *MPLS Virtual Private Network Enhancements*
- *Cisco IOS Switching Services Configuration Guide*, Release 12.2

EARLY FIELD TEST DRAFT - CISCO CONFIDENTIAL

Configuration Tasks

To configure the exchange of VPN-IPv4 addresses between two or more autonomous systems or subautonomous systems in a confederation, perform the following tasks:

- Configuring EBGP Routing for the Exchange of VPN Routes Between Autonomous Systems (required)

- Configuring EBGP Routing for the Exchange of VPN Routes Between Subautonomous Systems in a Confederation (required)

- Displaying VPN-IPv4 LFIB Entries (optional)

Before You Begin

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 this section build from those configuration tasks.

Perform (as appropriate to the existing network configuration) the following tasks as described in the *Cisco IOS Switching Services Configuration Guide* (the Configuring Multiprotocol Label Switching chapter).

- Define VPN routing instances

- Configure BGP routing sessions in the service provider (P) network

- Configure PE to PE routing sessions in the service provider (P) network

- Configure BGP PE to CE routing sessions

Configuring EBGP Routing for the Exchange of VPN Routes Between Autonomous Systems

Perform one of the following tasks to configure an EBGP border edge router in an autonomous system to exchange VPN 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 7 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 7   *Configuration for Peering with Directly-Connected Interface Between ASBRs*

Note When External Border Gateway Protocol (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** *peer-group-name* **remote-as** *as-number*
7. **neighbor** *peer-group-name* **activate**
8. **exit-address-family**

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`	Creates an EBGP routing process and assigns it an AS number. The autonomous system number is passed along to identify the router to EBGP routers in another autonomous system.

EARLY FIELD TEST DRAFT - CISCO CONFIDENTIAL

	Command or Action	Purpose

Step 4	`no bgp default route-target filter`
	
	Example:
	`Router(config)# no bgp default route-target filter`
		Disables BGP route-target filtering. All received BGP VPN-IPv4 routes are accepted by the router.

Step 5	`address-family vpnv4`[`unicast`] **Example:** `Router(config-router)# address-family vpnv4`	Configures a routing session to carry VPN-IPv4 addresses across the VPN backbone. Each address has been made globally unique by the addition of an 8-byte route distinguisher (RD). Unicast is optional; use it if you need to specify a unicast prefix.
Step 6	`neighbor` *peer-group-name* `remote-as` *as-number* **Example:** `Router(config-router-af)# neighbor Lowell remote-as 2`	Enters the address family submode and specifies a neighboring EBGP peer group. This EBGP peer group is identified to the specified autonomous system.

Step 7	`neighbor` *peer-group-name* `activate`
	
	Example:
	`Router(config-router-af)# neighbor Lowell activate`
		Activates the advertisement of the VPN-IPv4 address family to a neighboring EBGP router.

Step 8	`exit-address-family`
	
	Example:
	`Router(config-router-af)# exit-address-family`
		Exits from the address family submode of the global configuration mode.

Configuring Peering of the Loopback Interface of Directly-Connected ASBRs

This functionality is provided with the release of the Cisco IOS feature MPLS VPN—Loadbalancing Support for Inter-As and CSC VPNs.

This section describes the tasks you need to do to configure peering of loopback interfaces of directly-connected autonomous system boundary routers (ASBRs). The tasks include the following:

Figure 8 shows the loopback configuration for directly-connected ASBR1 and ASBR2 routers. This configuration is used as the example in the tasks that follow.

Figure 8   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 addresses.

> **Note**   Loopback addresses need to be configured for each directly-connected ASBR. That is, configure a loopback address for ASBR1 and for ASBR2 in our example (see Figure 8).

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 loopback0`	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

EARLY FIELD TEST DRAFT - CISCO CONFIDENTIAL

	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)# end`
		Exits to privileged EXEC mode.

Examples

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

```
configure terminal
interface loopback0
    ip address 10.10.10.10 255.255.255.255.255
```

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

```
configure terminal
interface loopback0
    ip address 10.20.20.20 255.255.255.255.255
```

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 Ethernet1/0 Y.Y.Y.Y
Router(config)# ip route X.X.X.X 255.255.255.255 Ethernet0/0 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 Ethernet1/0 168.192.0.1`
		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 that network.
		
		• The *interface-type 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 not 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 via 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 e1/0 168.192.0.1
ip route 10.20.20.20 255.255.255.255 e0/0 168.192.2.1
```

The following example shows the configuration of a /32 static route from theASBR2 router to the loopback address of the ASBR1 router:

```
configure terminal
ip route vrf vpn1 10.10.10.10 255.255.255.255 e1/0 168.192.0.2
ip route vrf vpn1 10.10.10.10 255.255.255.255 e0/0 168.192.2.2
```

Configuring Forwarding on Connecting Loopback Interfaces

Perform this task to configure forwarding on the connecting loopback interfaces.

This task is required for sessions between loopbacks. In the "Configuring /32 Static Routes to the EBGP Neighbor Loopback" 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. **mpls bgp forwarding**
5. **exit**
6. Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0)
7. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		
		• Enter your password if prompted.

Step 2	`configure terminal`
	
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	`interface` *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 type of interface to be configured. • The *slot* argument is the slot number. Refer to the appropriate hardware manual for slot and port information. • The */port* argument is the port number. Refer to the appropriate hardware manual for slot and port information.

Step 4	`mpls bgp forwarding`
	
	Example:
	`Router(config-if)# mpls bgp forwarding`
		Configures BGP to enable MPLS forwarding on connecting interfaces.

	Command or Action	Purpose

Step 5	`exit`
	
	Example:
	`Router(config-if)# exit`
		Exits to global configuration mode.

Step 6	Repeat Steps 3 and 4 for another connecting interface (Ethernet 0/0).	—

Step 7	`end`
	
	Example:
	`Router(config)# end`
		Exits to privileged EXEC mode.

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 168.192.0.2 255.255.255.255
    mpls bgp forwarding
    exit
interface ethernet 0/0
    ip address 168.192.2.0 255.255.255.255
    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 vrf forwarding vpn1
    ip address 168.192.0.1 255.255.255.255
    mpls bgp forwarding
    exit
interface ethernet 0/0
    ip vrf forwarding vpn1
    ip address 168.192.2.1 255.255.255.255
    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**] **vrf** *vrf-name*

9. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**

10. **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]

11. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		
		• Enter your password if prompted.

Step 2	`configure terminal`
	
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	`router bgp as-number`
	
	Example:
	`Router(config)# router bgp 200`
		Configures the BGP routing process.
		
		• The *as-number* indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along.

Step 4	`no bgp default route-target filter`
	
	Example:
	`Router(config)# no bgp default route-target filter`
		Disables BGP route-target filtering. All received BGP VPN-IPv4 routes are accepted by the router.

Step 5	`neighbor {ip-address	peer-group-name} remote-as as-number`  **Example:** `Router(config-router)# neighbor 10.20.20.20 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group. • The *as-number* argument is the autonomous system to which the neighbor belongs.
Step 6	`neighbor {ip-address	peer-group-name} disable-connected-check`  **Example:** `Router(config-router)# neighbor 10.20.20.20 disable-connected-check`	Allows peering between loopbacks. • The *ip-address* argument is the IP address of the neighbor. • The *peer-group-name* argument is the name of a BGP peer group.

	Command or Action	Purpose
Step 7	`neighbor {ip-address	ipv6-address	peer-group-name} update-source interface-type interface-number`  **Example:** `Router(config-router)# neighbor 10.20.20.20 update-source Loopback0`	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 the address family submode for configuring routing protocols such as BGP, Routing Information Protocol (RIP), and static routing.
		
		• The **vpnv4** keyword configures sessions that carry customer Virtual Private Network (VPN)-IPv4 prefixes, each of which has been made globally unique by adding an 8-byte route distinguisher.
		
		• The **unicast** keyword specifies unicast prefixes.

Step 9	`neighbor {ip-address	peer-group-name	ipv6-address} activate`  **Example:** `Router(config-router-af)# neighbor 10.20.20.20 activate`	Enables the exchange of information with a BGP neighbor. • The *ip-address* argument is the IP address of the neighboring router. • The *peer-group-name* argument name of 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.

EARLY FIELD TEST DRAFT - CISCO CONFIDENTIAL

	Command or Action	Purpose

Step 10	`neighbor {ip-address | peer-group-name} send-community [both | standard | extended]`
	
	Example:
	`Router(config-router-af)# neighbor 10.20.20.20 send-community extended`
		Specifies that a communities attribute should be sent to a BGP neighbor.
		
		• The *ip-address* argument is the IP address of the neighboring router.
		
		• The *peer-group-name* argument name of BGP peer group.
		
		• The **both** keyword specifies that both standard and extended communities will be sent.
		
		• The **standard** keyword specifies that only standard communities will be sent.
		
		• The **extended** keyword specifies that only extended communities will be sent.

Step 11	`end`
	
	Example:
	`Router(config)# end`
		Exits to privileged EXEC mode.

Examples

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

```
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 Loopback0
!
 address-family vpnv4
 neighbor 10.20.20.20 activate
 neighbor 10.20.20.20 send-community extended
```

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

```
router bgp 200
 bgp log-neighbor-changes
 neighbor 10.10.10.10 remote-as 100
 neighbor 10.10.10.10 disable-connected-check
 neighbor 10.10.10.10 update-source Loopback0
!
 address-family vpnv4
 neighbor 10.10.10.10 activate
 neighbor 10.10.10.10 send-community extended
```

Verifying That Load Balancing Occurs Between Loopbacks

To verify that load-balancing can occur between loopbacks, ensure that the MPLS Label Forwarding Information Base (LFIB) entry for the neighbor route lists the available paths and interfaces.

SUMMARY STEPS

1. **enable**

2. **show mpls forwarding-table**

3. **disable**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		
		• Enter your password if prompted.

Step 2	`show mpls forwarding-table` [*network* {*mask* | *length*} | `labels` *label* [`-` *label*] | `interface` *interface* | `next-hop` *address* | `lsp-tunnel` [*tunnel-id*]] [`vrf` *vrf-name*] [`detail`] **Example:** `Router# show mpls forwarding-table`	Displays the contents of the MPLS label forwarding information base (LFIB).

Step 3	`disable`
	
	Example:
	`Router# disable`
		Exits to user EXEC mode.

Examples

TBD

Configuring EBGP Routing for the Exchange of VPN Routes Between Subautonomous Systems in a Confederation

To configure EBGP border edge router in a confederation to exchange VPN routes with another subautonomous system, use the following commands starting in EXEC mode.

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.

EARLY FIELD TEST DRAFT - CISCO CONFIDENTIAL

	Command	Purpose
Step 1	Router# **configure terminal**	Enters the global configuration mode.
Step 2	Router(config)# **router bgp** *sub-autonomous-system*	Creates an EBGP routing process and assigns it an AS number. The subautonomous system number is passed along to identify the router to EBGP routers in other subautonomous systems.
Step 3	Router(config)# **bgp confederation identifier** *as-number*	Defines an EBGP confederation by specifying a confederation identifier associated with each subautonomous system. The subautonomous systems appear as a single autonomous system.
Step 4	Router(config)# **bgp confederation peers** *sub-autonomous-systems*	Specifies the subautonomous systems that belong to the confederation (identifying neighbors from other subautonomous systems within the confederation as special EBGP peers).
Step 5	Router(config)# **no bgp default route-target filter**	Disables BGP route-target community filtering. All received BGP VPN-IPv4 routes are accepted by the router.
Step 6	Router(config-router)# **address-family vpnv4**[**unicast**]	Configures a routing session to carry VPN-IPv4 addresses across the VPN backbone. Each address has been made globally unique by the addition of an 8-byte route distinguisher (RD). Unicast is optional; use it if you need to specify a unicast prefix.
Step 7	Router(config-router-af)# **neighbor** *peer-group-name* **remote-as** *as-number*	Enters the address family submode and specifies a neighboring EBGP peer group. This EBGP peer group is identified to the specified subautonomous system.
Step 8	Router(config-router-af)# **neighbor** *peer-group-name* **next-hop-self**	Advertises the router as the next hop for the specified neighbor. If you specify a next-hop-self address as part of the router configuration, you do not need to use the **redistribute connected** command.
Step 9	Router(config-router-af)# **neighbor** *peer-group-name* **activate**	Activates the advertisement of the VPN-IPv4 address family to a neighboring PE router in the specified subautonomous system.
Step 10	Router(config-router-af)# **exit-address-family**	Exits from the address family submode of the global configuration mode.

Displaying VPN-IPv4 LFIB Entries

To display the VPN-IPv4 label forwarding information base (LFIB) entries at the border edge routers in the autonomous systems, use the following commands starting in EXEC mode:

	Command	Purpose
Step 1	Router# **show ip bgp vpnv4 all** [*tags*]	Displays information about all VPN-IPv4 labels.
Step 2	Router# **show tag-switching forwarding-table**	Displays the contents of the LFIB (such as VPN-IPv4 prefix/length and BGP next hop destination for the route).

The following is an example of how the VPN-IPv4 LFIB entries appear when you use the **show tag-switching forwarding-table** privileged EXEC command:

```
Router# show tag-switching 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
```

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

Configuration Examples

This section provides the following configuration examples:

- Configuring EBGP Routing to Exchange VPN Routes Between Autonomous Systems Examples
- Configuring EBGP Routing to Exchange VPN Routes Between Autonomous Systems in a Confederation Examples

Configuring EBGP Routing to Exchange VPN Routes Between Autonomous Systems Examples

The network topology in Figure 9 shows two autonomous systems, which are configured as follows:

- Autonomous system 1 (AS1) includes PE1, P1, EBGP1. The IGP is OSPF.
- Autonomous system 2 (AS2) includes PE2, P2, EBGP2. The IGP is ISIS.
- CE1 and CE2 belongs to the same VPN, which is called VPN1.
- The P routers are route reflectors.
- EBGP1 is configured with the **redistribute connected subnets** command.
- EBGP2 is configured with the **neighbor next-hop-self** command.

Figure 9   Configuring Two Autonomous Systems
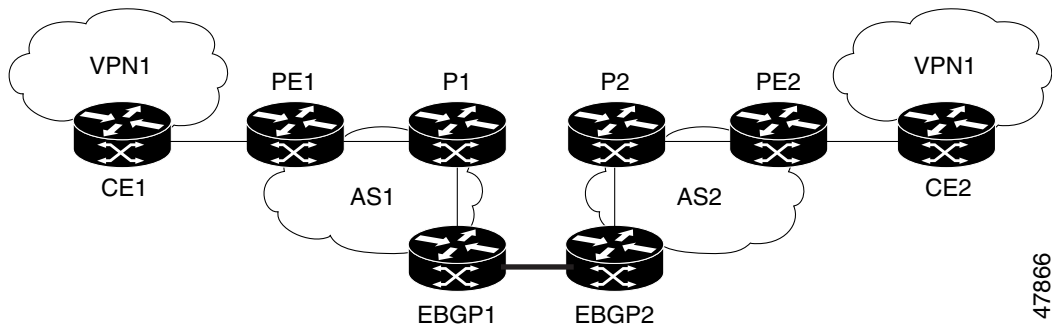
Configuration for Autonomous System 1, CE1 Example

```
CE1: Burlington
!
interface Loopback1
 ip address 1.0.0.6 255.255.255.255
```

EARLY FIELD TEST DRAFT - CISCO CONFIDENTIAL

```
!
interface Serial1/3
 description Veritas
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
!
interface Serial1/3.1 point-to-point
 description Veritas
 ip address 1.6.2.1 255.255.255.252
 frame-relay interface-dlci 22
!
router ospf 1
 network 1.0.0.0 0.255.255.255 area 0
```

Configuration for Autonomous System 1, PE1 Example

```
PE1: Veritas
!
ip cef
!
ip vrf V1
 rd 1:105
 route-target export 1:100
 route-target import 1:100
!
interface Serial0/0
 description Burlington
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 2000000
!
interface Serial0/0.3 point-to-point
 description Burlington
 ip vrf forwarding V1
 ip address 1.6.2.2 255.255.255.252
 frame-relay interface-dlci 22
!
interface Ethernet0/1
 description Vermont
 ip address 100.2.2.5 255.255.255.0
 tag-switching ip
!
router ospf 1
 log-adjacency-changes
 network 100.0.0.0 0.255.255.255 area 0
!
router ospf 10 vrf V1
 log-adjacency-changes
 redistribute bgp 1 metric 100 subnets
 network 1.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 neighbor R peer-group
 neighbor R remote-as 1
 neighbor R update-source Loopback0
 neighbor 100.0.0.2 peer-group R
 no auto-summary
 !
```

EARLY FIELD TEST DRAFT - CISCO CONFIDENTIAL

```
address-family ipv4 vrf V1
 redistribute ospf 10
 no auto-summary
 no synchronization
 exit-address-family
!
address-family vpnv4
 neighbor R activate
 neighbor R send-community extended
 neighbor 100.0.0.2 peer-group R
 no auto-summary
 exit-address-family
```

Configuration for Autonomous System 1, P1 Example

```
P1: Vermont
!
ip cef
!
interface Loopback0
 ip address 100.0.0.2 255.255.255.255
!
interface Ethernet0/1
 description Ogunquit
 ip address 100.2.1.1 255.255.255.0
 tag-switching ip
!
interface FastEthernet2/0
 description Veritas
 ip address 100.2.2.1 255.255.255.0
 duplex auto
 speed auto
 tag-switching ip
!
router ospf 1
 log-adjacency-changes
 network 100.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor R peer-group
 neighbor R remote-as 1
 neighbor R update-source Loopback0
 neighbor R route-reflector-client
 neighbor 100.0.0.4 peer-group R
 neighbor 100.0.0.5 peer-group R
 !
 address-family vpnv4
 neighbor R activate
 neighbor R route-reflector-client
 neighbor R send-community extended
 neighbor 100.0.0.4 peer-group R
 neighbor 100.0.0.5 peer-group R
 exit-address-family
```

Configuration for Autonomous System 1, EBGP1 Example

```
EBGP1: Ogunquit
!
ip cef
```

EARLY FIELD TEST DRAFT - CISCO CONFIDENTIAL

```
!
interface Loopback0
 ip address 100.0.0.4 255.255.255.255
!
EBGP1: Ogunquit
!
ip cef
!
interface Loopback0
 ip address 100.0.0.4 255.255.255.255
!
interface Ethernet0/1
 description Vermont
 ip address 100.2.1.40 255.255.255.0
 tag-switching ip
!
interface ATM1/0
 description Lowell
 no ip address
 no atm scrambling cell-payload
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 description Lowell
 ip address 12.0.0.1 255.255.255.252
 pvc 1/100
!
router ospf 1
 log-adjacency-changes
 redistribute connected subnets
 network 100.0.0.0 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
 neighbor R update-source Loopback0
 neighbor 12.0.0.2 remote-as 2
 neighbor 100.0.0.2 peer-group R
 no auto-summary
 !
 address-family vpnv4
  neighbor R activate
  neighbor R send-community extended
  neighbor 12.0.0.2 activate
  neighbor 12.0.0.2 send-community extended
  neighbor 100.0.0.2 peer-group R
  no auto-summary
  exit-address-family
```

Configuration for Autonomous System 2, EBGP2 Example

```
EBGP2: Lowell
!
ip cef
!
ip vrf V1
 rd 2:103
 route-target export 1:100
 route-target import 1:100
```

EARLY FIELD TEST DRAFT - CISCO CONFIDENTIAL

```
!
interface Loopback0
 ip address 200.0.0.3 255.255.255.255
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
 ip address 1.0.0.3 255.255.255.255
!
interface Serial0/0
 description Littleton
 no ip address
 encapsulation frame-relay
 load-interval 30
 no fair-queue
 clockrate 2000000
!
interface Serial0/0.2 point-to-point
 description Littleton
 ip unnumbered Loopback0
 ip router isis
 tag-switching ip
 frame-relay interface-dlci 23
!
interface ATM1/0
 description Ogunquit
 no ip address
 atm clock INTERNAL
 no atm scrambling cell-payload
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 description Ogunquit
 ip address 12.0.0.2 255.255.255.252
 pvc 1/100
!
router isis
 net 49.0002.0000.0000.0003.00
!
router bgp 2
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 neighbor 12.0.0.1 remote-as 1
 neighbor 200.0.0.8 remote-as 2
 neighbor 200.0.0.8 update-source Loopback0
 neighbor 200.0.0.8 next-hop-self
!
 address-family ipv4 vrf V1
  redistribute connected
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor 12.0.0.1 activate
  neighbor 12.0.0.1 send-community extended
  neighbor 200.0.0.8 activate
  neighbor 200.0.0.8 next-hop-self
  neighbor 200.0.0.8 send-community extended
  exit-address-family
```

Configuration for Autonomous System 2, P2 Example

```
P2: Littleton
!
ip cef
!
ip vrf V1
 rd 2:108
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 200.0.0.8 255.255.255.255
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
 ip address 1.0.0.8 255.255.255.255
!
interface FastEthernet0/0
 description Pax
 ip address 200.9.1.2 255.255.255.0
 ip router isis
 tag-switching ip
!
interface Serial5/0
 description Lowell
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
!
interface Serial5/0.1 point-to-point
 description Lowell
 ip unnumbered Loopback0
 ip router isis
 tag-switching ip
 frame-relay interface-dlci 23
!
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
 neighbor R update-source Loopback0
 neighbor R route-reflector-client
 neighbor 200.0.0.3 peer-group R
 neighbor 200.0.0.9 peer-group R
 !
 address-family ipv4 vrf V1
  redistribute connected
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor R activate
  neighbor R route-reflector-client
  neighbor R send-community extended
  neighbor 200.0.0.3 peer-group R
  neighbor 200.0.0.9 peer-group R
```

```
        exit-address-family
```

Configuration for Autonomous System 2, PE2 Example

```
PE2: Pax
!
ip cef
!
ip vrf V1
 rd 2:109
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 200.0.0.9 255.255.255.255
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
 ip address 1.0.0.9 255.255.255.255
!
interface Serial0/0
 description Bethel
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
 no fair-queue
 clockrate 2000000
!
interface Serial0/0.1 point-to-point
 description Bethel
 ip vrf forwarding V1
 ip unnumbered Loopback1
 frame-relay interface-dlci 24
!
interface FastEthernet0/1
 description Littleton
 ip address 200.9.1.1 255.255.255.0
 ip router isis
 tag-switching ip
!
router ospf 10 vrf V1
 log-adjacency-changes
 redistribute bgp 2 subnets
 network 1.0.0.0 0.255.255.255 area 0
!
router isis
 net 49.0002.0000.0000.0009.00
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor 200.0.0.8 remote-as 2
 neighbor 200.0.0.8 update-source Loopback0
 !
 address-family ipv4 vrf V1
  redistribute connected
  redistribute ospf 10
  no auto-summary
  no synchronization
  exit-address-family
 !
```

```
address-family vpnv4
 neighbor 200.0.0.8 activate
 neighbor 200.0.0.8 send-community extended
 exit-address-family v
```
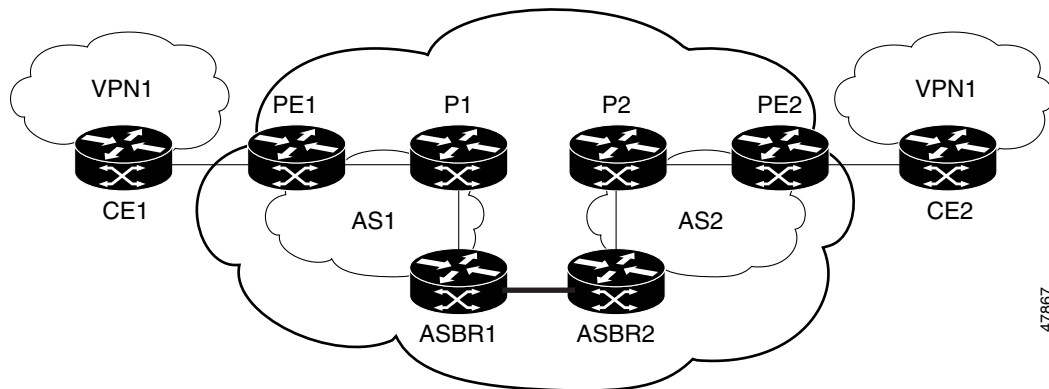
Configuration for Autonomous System 2, CE2 Example

```
CE2: Bethel
!
interface Loopback0
 ip address 1.0.0.11 255.255.255.255
!
interface Serial0
 description Pax
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 2000000
!
interface Serial0.1 point-to-point
 description Pax
 ip unnumbered Loopback0
 frame-relay interface-dlci 24
!
router ospf 1
 network 1.0.0.0 0.255.255.255 area 0
```

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

The network topology in Figure 10 shows a single Internet service provider (ISP), which is partitioning the backbone with confederations. The AS number of the provider is 100. The two autonomous systems run their own IGPs and are configured as follows:

- Autonomous system 1 (AS1) includes PE1, P1, EBGP1. The IGP is OSPF.

- Autonomous system 2 (AS2) includes PE2, P2, EBGP2. The IGP is ISIS.

- CE1 and CE2 belongs to the same VPN, which is called VPN1.

- The P routers are route reflectors.

- EBGP1 is configured with the **redistribute connected subnets** command.

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

Figure 10   Configuring Two Autonomous Systems in a Confederation

Configuration for Autonomous System 1, CE1 Example

```
CE1: Burlington
!
interface Loopback1
 ip address 1.0.0.6 255.255.255.255
!
interface Serial1/3
 description Veritas
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
!
interface Serial1/3.1 point-to-point
 description Veritas
 ip address 1.6.2.1 255.255.255.252
 frame-relay interface-dlci 22
!
router ospf 1
 network 1.0.0.0 0.255.255.255 area 0
```

Configuration for Autonomous System 1, PE1 Example

```
PE1: Veritas
!
ip cef
!
ip vrf V1
 rd 1:105
 route-target export 1:100
 route-target import 1:100
!
interface Serial0/0
 description Burlington
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 2000000
!
interface Serial0/0.3 point-to-point
 description Burlington
```

```
  ip vrf forwarding V1
  ip address 1.6.2.2 255.255.255.252
  frame-relay interface-dlci 22
!
interface Ethernet0/1
 description Vermont
 ip address 100.2.2.5 255.255.255.0
 tag-switching ip
!
router ospf 1
 log-adjacency-changes
 network 100.0.0.0 0.255.255.255 area 0
!
router ospf 10 vrf V1
 log-adjacency-changes
 redistribute bgp 1 metric 100 subnets
 network 1.0.0.0 0.255.255.255 area 0
!
router bgp 1
 no synchronization
 bgp confederation identifier 100
 bgp confederation identifier 100
 neighbor R peer-group
 neighbor R remote-as 1
 neighbor R update-source Loopback0
 neighbor 100.0.0.2 peer-group R
 no auto-summary
 !
 address-family ipv4 vrf V1
  redistribute ospf 10
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor R activate
  neighbor R send-community extended
  neighbor 100.0.0.2 peer-group R
  no auto-summary
  exit-address-family
```

Configuration for Autonomous System 1, P1 Example

```
P1: Vermont
!
ip cef
!
interface Loopback0
 ip address 100.0.0.2 255.255.255.255
!
interface Ethernet0/1
 description Ogunquit
 ip address 100.2.1.1 255.255.255.0
 tag-switching ip
!
interface FastEthernet2/0
 description Veritas
 ip address 100.2.2.1 255.255.255.0
 duplex auto
 speed auto
 tag-switching ip
!
```

EARLY FIELD TEST DRAFT - CISCO CONFIDENTIAL

```
router ospf 1
 log-adjacency-changes
 network 100.0.0.0 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
 neighbor R update-source Loopback0
 neighbor R route-reflector-client
 neighbor 100.0.0.4 peer-group R
 neighbor 100.0.0.5 peer-group R
 !
 address-family vpnv4
  neighbor R activate
  neighbor R route-reflector-client
  neighbor R send-community extended
  neighbor 100.0.0.4 peer-group R
  neighbor 100.0.0.5 peer-group R
  exit-address-family
```

Configuration for Autonomous System 1, EBGP1 Example

```
EBGP1: Ogunquit
!
ip cef
!
interface Loopback0
 ip address 100.0.0.4 255.255.255.255
!
interface Ethernet0/1
 description Vermont
 ip address 100.2.1.40 255.255.255.0
 tag-switching ip
!
interface ATM1/0
 description Lowell
 no ip address
 no atm scrambling cell-payload
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 description Lowell
 ip address 12.0.0.1 255.255.255.252
 pvc 1/100
!
router ospf 1
 log-adjacency-changes
 redistribute connected subnets
 network 100.0.0.0 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 1
 neighbor R peer-group
 neighbor R remote-as 1
 neighbor R update-source Loopback0
 neighbor 12.0.0.2 remote-as 2
 neighbor 12.0.0.2 next-hop-self
 neighbor 100.0.0.2 peer-group R
 no auto-summary
 !
 address-family vpnv4
  neighbor R activate
  neighbor R send-community extended
  neighbor 12.0.0.2 activate
  neighbor 12.0.0.2 next-hop-self
  neighbor 12.0.0.2 send-community extended
  neighbor 100.0.0.2 peer-group R
  no auto-summary
  exit-address-family
```

Configuration for Autonomous System 2, EBGP2 Example

```
EBGP2: Lowell
!
ip cef
!
ip vrf V1
 rd 2:103
```

```
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 200.0.0.3 255.255.255.255
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
 ip address 1.0.0.3 255.255.255.255
!
interface Serial0/0
 description Littleton
 no ip address
 encapsulation frame-relay
 load-interval 30
 no fair-queue
 clockrate 2000000
!
interface Serial0/0.2 point-to-point
 description Littleton
 ip unnumbered Loopback0
 ip router isis
 tag-switching ip
 frame-relay interface-dlci 23
!
interface ATM1/0
 description Ogunquit
 no ip address
 atm clock INTERNAL
 no atm scrambling cell-payload
 no atm ilmi-keepalive
!
interface ATM1/0.1 point-to-point
 description Ogunquit
 ip address 12.0.0.2 255.255.255.252
 pvc 1/100
!
router isis
 net 49.0002.0000.0000.0003.00
!
router bgp 2
 no synchronization
 no bgp default route-target filter
 bgp log-neighbor-changes
 bgp confederation identifier 100
 bgp confederation peers 1
 neighbor 12.0.0.1 remote-as 1
 neighbor 12.0.0.1 next-hop-self
 neighbor 200.0.0.8 remote-as 2
 neighbor 200.0.0.8 update-source Loopback0
 neighbor 200.0.0.8 next-hop-self
 !
 address-family ipv4 vrf V1
  redistribute connected
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family vpnv4
  neighbor 12.0.0.1 activate
  neighbor 12.0.0.1 next-hop-self
  neighbor 12.0.0.1 send-community extended
  neighbor 200.0.0.8 activate
```

```
    neighbor 200.0.0.8 next-hop-self
    neighbor 200.0.0.8 send-community extended
    exit-address-family
```

Configuration for Autonomous System 2, P2 Example

```
P2: Littleton
!
ip cef
!
ip vrf V1
 rd 2:108
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 200.0.0.8 255.255.255.255
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
 ip address 1.0.0.8 255.255.255.255
!
interface FastEthernet0/0
 description Pax
 ip address 200.9.1.2 255.255.255.0
 ip router isis
 tag-switching ip
!
interface Serial5/0
 description Lowell
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
!
interface Serial5/0.1 point-to-point
 description Lowell
 ip unnumbered Loopback0
 ip router isis
 tag-switching ip
 frame-relay interface-dlci 23
!
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
 neighbor R update-source Loopback0
 neighbor R route-reflector-client
 neighbor 200.0.0.3 peer-group R
 neighbor 200.0.0.9 peer-group R
 !
 address-family ipv4 vrf V1
  redistribute connected
  no auto-summary
  no synchronization
  exit-address-family
 !
```

```
 address-family vpnv4
  neighbor R activate
  neighbor R route-reflector-client
  neighbor R send-community extended
  neighbor 200.0.0.3 peer-group R
  neighbor 200.0.0.9 peer-group R
  exit-address-family
```

Configuration for Autonomous System 2, PE2 Example

```
PE2: Pax
!
ip cef
!
ip vrf V1
 rd 2:109
 route-target export 1:100
 route-target import 1:100
!
interface Loopback0
 ip address 200.0.0.9 255.255.255.255
 ip router isis
!
interface Loopback1
 ip vrf forwarding V1
 ip address 1.0.0.9 255.255.255.255
!
interface Serial0/0
 description Bethel
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
 no fair-queue
 clockrate 2000000
!
interface Serial0/0.1 point-to-point
 description Bethel
 ip vrf forwarding V1
 ip unnumbered Loopback1
 frame-relay interface-dlci 24
!
interface FastEthernet0/1
 description Littleton
 ip address 200.9.1.1 255.255.255.0
 ip router isis
 tag-switching ip
!
router ospf 10 vrf V1
 log-adjacency-changes
 redistribute bgp 2 subnets
 network 1.0.0.0 0.255.255.255 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 200.0.0.8 remote-as 2
 neighbor 200.0.0.8 update-source Loopback0
 !
```

```
address-family ipv4 vrf V1
 redistribute connected
 redistribute ospf 10
 no auto-summary
 no synchronization
 exit-address-family
!
 address-family vpnv4
 neighbor 200.0.0.8 activate
 neighbor 200.0.0.8 send-community extended
 exit-address-family
```

Configuration for Autonomous System 2, CE2 Example

```
CE2: Bethel
!
interface Loopback0
 ip address 1.0.0.11 255.255.255.255
!
interface Serial0
 description Pax
 no ip address
 encapsulation frame-relay
 no fair-queue
 clockrate 2000000
!
interface Serial0.1 point-to-point
 description Pax
 ip unnumbered Loopback0
 frame-relay interface-dlci 24
!
router ospf 1
 network 1.0.0.0 0.255.255.255 area 0
```

Command Reference

This section documents the following new command related to interautonomous system MPLS VPN operation:

- **bgp default route-target filter**

- **mpls bgp forwarding**

All other commands used with this feature are described in the following Cisco IOS documentation:

- *MPLS Virtual Private Networks (VPNs)*

- *MPLS Virtual Private Network Enhancements*

- *Cisco IOS Switching Services Configuration Guide* (Release 12.3), Multiprotocol Label Switching

bgp default route-target filter

To enable automatic Border Gateway Protocol (BGP) route-target community filtering, use the **bgp default route-target filter** command in router configuration mode. To disable this feature, 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.

Defaults   This command is enabled by default.

Command Modes   Router configuration

Command History

Release	Modification
12.1(5)T	This command was introduced.
12.0(16)ST	This command was integrated into the Cisco IOS 12.0(16)ST release.
12.0(22)S	This command was integrated into the Cisco IOS 12.0(22)S release.
12.0(23)S	This command was integrated into the Cisco IOS 12.0(23)S release.
12.0(24)S	This command was integrated into the Cisco IOS 12.0(24)S release.

Usage Guidelines   You use this command to control the distribution of 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 Confederation Exterior Border Gateway Protocol (CEBGP) border edge router.

If you configure the router for BGP route-target community filtering, all received 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 VRFs. This is the desired behavior for a router configured as a 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
```

EARLY FIELD TEST DRAFT - CISCO CONFIDENTIAL

Related Commands	Command	Description
	show tag-switching forwarding-table	Displays the contents of the LFIB.

mpls bgp forwarding

To enable an interface to receive Multiprotocol Label Switching (MPLS) packets when the signalling 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 MPLS forwarding by BGP on an interface, use the **no** form of this command.

> **mpls bgp forwarding**

> **no mpls bgp forwarding**

Syntax Description   This command has no arguments or keywords.

Defaults   MPLS forwarding by BGP is not enabled.

Command Modes   Interface configuration

Command History

Release	Modification
12.0(29)S	This command was introduced.

Usage Guidelines   Us 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 non-loopback 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)# bgp mpls forwarding
```

Related Commands

Command	Description
ip vrf forwarding	Associates a Virtual Private network (VPN) routing/forwarding instance (VRF) with an interface or subinterface.

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 table —VPN routing/forwarding table. 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. A VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

EARLY FIELD TEST DRAFT - CISCO CONFIDENTIAL