# MPLS VPN—Inter-AS—IPv4 BGP Label Distribution

This feature enables you to set up a Virtual Private Network (VPN) service provider network so that the autonomous system boundary routers (ASBRs) exchange IPv4 routes with Multiprotocol Label Switching (MPLS) labels of the provider edge (PE) routers. Route reflectors (RRs) exchange VPNv4 routes by using multihop, multiprotocol, External Border Gateway Protocol (EBGP). This configuration saves the ASBRs from having to store all the VPNv4 routes. Using the route reflectors to store the VPNv4 routes and forward them to the PE routers results in improved scalability.

The MPLS VPN—Inter-AS—IPv4 BGP Label Distribution feature has the following benefits:

- Having the route reflectors store VPNv4 routes results in improved scalability—This configuration scales better than other configurations where the ASBR holds all of the VPNv4 routes and forwards the routes based on VPNv4 labels. With this configuration, route reflectors hold the VPNv4 routes, which simplifies the configuration at the border of the network.
- Enables a non-VPN core network to act as a transit network for VPN traffic—You can transport IPv4 routes with MPLS labels over a non MPLS VPN service provider.
- Eliminates the need for any other label distribution protocol between adjacent LSRs—If two adjacent label switch routers (LSRs) are also BGP peers, BGP can handle the distribution of the MPLS labels. No other label distribution protocol is needed between the two LSRs.
- Includes EBGP multipath support to enable load balancing for IPv4 routes across autonomous system (AS) boundaries.

**Feature History for MPLS VPN—Inter-AS—IPv4 BGP Label Distribution**

| Release | Modification |
|---|---|
| 12.0(21)ST | This feature was introduced. |
| 12.0(22)S | This feature was implemented on the Cisco 12000 series routers (for specific line cards supported, see Table 1) and integrated into Cisco IOS Release 12.0(22)S. |
| 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). |
| 12.2(13)T | This feature was integrated into Cisco IOS Release 12.2(13)T. |

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

| | |
|---|---|
| 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) and implemented on Cisco IOS 12.0(24)S. |
| 12.2(14)S | This feature was integrated into Cisco IOS Release 12.2(14)S and implemented on Cisco 7200 and Cisco 7500 series routers. |
| 12.0(27)S | Support was added for EBGP multipath on the provider edge (PE)-customer edge (CE) links. |
| 12.2(28)SB | This feature was integrated into Cisco IOS Release 12.2(28)SB. |

**Note** Software images for Cisco 12000 series Internet routers have been deferred to Cisco IOS Release 12.0(27)S1.

#### Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

# Contents


- Prerequisites for MPLS VPN—Inter-AS—IPv4 BGP Label Distribution, page 3
- Restrictions for MPLS VPN—Inter-AS—IPv4 BGP Label Distribution, page 3
- Information About MPLS VPN—Inter-AS—IPv4 BGP Label Distribution, page 4
- How to Configure MPLS VPN—Inter-AS—IPv4 BGP Label Distribution, page 6
- Configuration Examples for MPLS VPN—Inter-AS—IPv4 BGP Label Distribution, page 26
- Additional References, page 43
- Command Reference, page 45
- Glossary, page 46

# Prerequisites for MPLS VPN—Inter-AS—IPv4 BGP Label Distribution

The network must be properly configured for MPLS VPN operation before you configure this feature.

Table 1 lists the Cisco 12000 series line card support added by Cisco IOS S releases.

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

| Type | Line Cards | Cisco IOS Release Supported |
|---|---|---|
| Packet Over SONET (POS) | 4-Port OC-3 POS<br>8-Port OC-3 POS<br>16-Port OC-3 POS<br>1-Port OC-12 POS<br>4-Port OC-12 POS<br>1-Port OC-48 POS<br>4-Port OC-3 POS ISE<br>8-Port OC-3 POS ISE<br>16-Port OC-3 POS ISE<br>4-Port OC-12 POS ISE<br>1-Port OC-48 POS ISE | 12.0(22)S, 12.0(23)S, 12.0(27)S |
| Electrical Interface | 6-Port DS3<br>12-Port DS3<br>6-Port E3<br>12-Port E3 | 12.0(22)S, 12.0(23)S, 12.0(27)S |
| Ethernet | 3-Port GbE | 12.0(23)S, 12.0(27)S |
| Asynchronous Transfer Mode (ATM) | 4-Port OC-3 ATM<br>1-Port OC12 ATM<br>4-Port OC-12 ATM<br>8-Port OC-3 ATM | 12.0(22)S, 12.0(23)S, 12.0(27)S<br><br><br>12.0(23)S |
| Channelized Interface | 2-Port CHOC-3<br>6-Port Ch T3 (DS1)<br>1-Port CHOC-12 (DS3)<br>1-Port CHOC-12 (OC-3)<br>4-Port CHOC-12 ISE<br>1-Port CHOC-48 ISE | 12.0(22)S, 12.0(23)S, 12.0(27)S |

# Restrictions for MPLS VPN—Inter-AS—IPv4 BGP Label Distribution

This feature includes the following restrictions:

- For networks configured with EBGP multihop, a labeled switched path (LSP) must be established between nonadjacent routers. (RFC 3107)
- The PE routers must run images that support BGP label distribution. Otherwise, you cannot run EBGP between them.

- Point-to-Point Protocol (PPP) encapsulation on the ASBRs is not supported with this feature.
- The physical interfaces that connect the BGP speakers must support Cisco Express Forwarding (CEF) or distributed CEF and MPLS.

# Information About MPLS VPN—Inter-AS—IPv4 BGP Label Distribution

To configure MPLS VPN—Inter-AS—IPv4 BGP Label Distribution, you need the following information:


- MPLS VPN—Inter-AS—IPv4 BGP Label Distribution Overview, page 4
- BGP Routing Information, page 5
- Types of BGP Messages and MPLS Labels, page 5
- How BGP Sends MPLS Labels with Routes, page 6
- Using Route Maps to Filter Routes, page 6

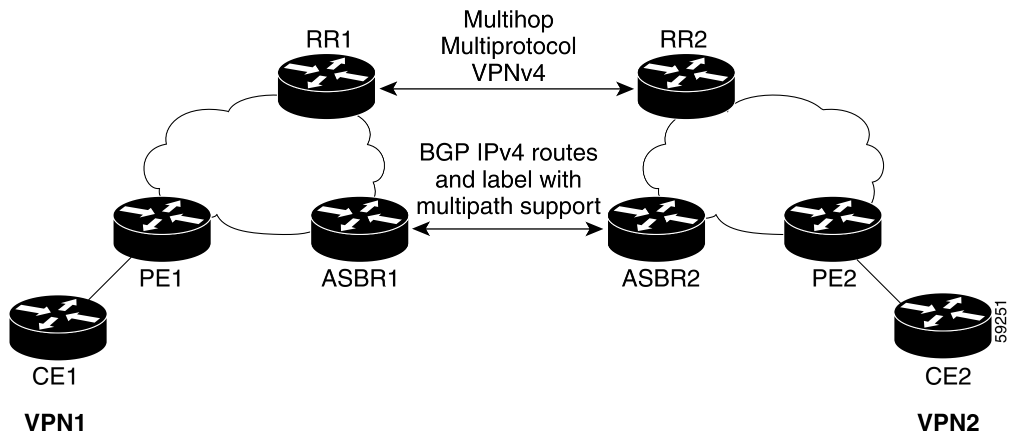

## MPLS VPN—Inter-AS—IPv4 BGP Label Distribution Overview

This feature enables you to set up a VPN service provider network to exchange IPv4 routes with MPLS labels. You can configure the VPN service provider network as follows:

- Route reflectors exchange VPNv4 routes by using multihop, multiprotocol EBGP. This configuration also preserves the next hop information and the VPN labels across the autonomous systems.
- A local PE router (for example, PE1 in Figure 1) needs to know the routes and label information for the remote PE router (PE2). This information can be exchanged between the PE routers and ASBRs in one of two ways:
  - **Internal Gateway Protocol (IGP) and Label Distribution Protocol (LDP):** The ASBR can redistribute the IPv4 routes and MPLS labels it learned from EBGP into IGP and LDP and vice versa.
  - **Internal Border Gateway Protocol (IBGP) IPv4 label distribution:** The ASBR and PE router can use direct IBGP sessions to exchange VPNv4 and IPv4 routes and MPLS labels.

    Alternatively, the route reflector can reflect the IPv4 routes and MPLS labels learned from the ASBR to the PE routers in the VPN. This is accomplished by enabling the ASBR to exchange IPv4 routes and MPLS labels with the route reflector. The route reflector also reflects the VPNv4 routes to the PE routers in the VPN (as mentioned in the first bullet). For example, in VPN1, RR1 reflects to PE1 the VPNv4 routes it learned and IPv4 routes and MPLS labels learned from ASBR1. Using the route reflectors to store the VPNv4 routes and forward them through the PE routers and ASBRs allows for a scalable configuration.
- ASBRs exchange IPv4 routes and MPLS labels for the PE routers by using EBGP. This enables load balancing across CSC boundaries.

*Figure 1 VPNs Using EBGP and IBGP to Distribute Routes and MPLS Labels*




# BGP Routing Information

BGP routing information includes the following items:

- A network number (prefix), which is the IP address of the destination.
- Autonomous system (AS) path, which is a list of the other ASs through which a route passes on its way to the local router. The first AS in the list is closest to the local router; the last AS in the list is farthest from the local router and usually the AS where the route began.
- Path attributes, which provide other information about the AS path, for example, the next hop.

# Types of BGP Messages and MPLS Labels

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

- Open Messages—After a router establishes a TCP connection with a neighboring router, the routers exchange open messages. This message contains the AS number to which the router belongs and the IP address of the router who sent the message.
- Update Messages—When a router has a new, changed, or broken route, it sends an update message to the neighboring router. This message contains the Network Layer Reachability Information (NLRI), which lists the IP addresses of the usable routes. The update message also includes any routes that are no longer usable. The update message also includes path attributes and the lengths of both the usable and unusable paths. Labels for VPNv4 routes are encoded in the update message as specified in RFC 2858. The labels for the IPv4 routes are encoded in the update message as specified in RFC 3107.
- Keepalive Messages—Routers exchange keepalive messages to determine if a neighboring router is still available to exchange routing information. The router sends these messages at regular intervals. (Sixty seconds is the default for Cisco routers.) The keepalive message does not contain routing data; it only contains a message header.
- Notification Messages—When a router detects an error, it sends a notification message.

## How BGP Sends MPLS Labels with Routes

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

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

## Using Route Maps to Filter Routes

When both routers are configured to distribute routes with MPLS labels, all the routes are encoded with the multiprotocol extensions and contain an MPLS label. You can use a route map to control the distribution of MPLS labels between routers. Route maps enable you to specify the following:

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

# How to Configure MPLS VPN—Inter-AS—IPv4 BGP Label Distribution

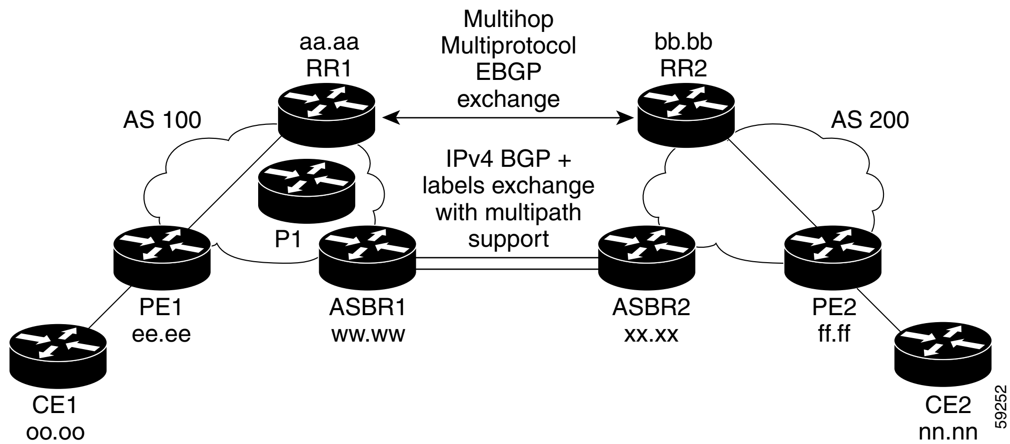
This section contains the following procedures:

- Configuring the ASBRs to Exchange IPv4 Routes and MPLS Labels, page 7
- Configuring the Route Reflectors to Exchange VPNv4 Routes, page 9
- Configuring the Route Reflectors to Reflect Remote Routes in Its AS, page 11
- Creating Route Maps, page 14
- Applying the Route Maps to the ASBRs, page 17
- Verifying the MPLS VPN—Inter-AS—IPv4 BGP Label Distribution Configuration, page 19

Figure 2 shows the following sample configuration:

- The configuration consists of two VPNs.
- The ASBRs exchange the IPv4 routes with MPLS labels.
- The route reflectors exchange the VPNv4 routes using multlihop MPLS EBGP.
- The route reflectors reflect the IPv4 and VPNv4 routes to the other routers in its AS.

*Figure 2 Configuring Two VPN Service Providers to Exchange IPv4 Routes and MPLS Labels*




# Configuring the ASBRs to Exchange IPv4 Routes and MPLS Labels

Perform this task to configure the ASBRs so that they can distribute BGP routes with MPLS labels.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
5. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
6. **maximum paths** *number-paths* (optional for EBGP multipath between the CSC-PE and CSC-CEs)
7. **neighbor** {*ip-address* | *peer-group-name*} **activate**
8. **neighbor** *ip-address* **send-label**
9. **exit-address-family**
10. **end**

## DETAILED STEPS

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

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **router bgp** *as-number*<br><br>**Example:**<br>`Router(config)# router bgp 100` | Configures a BGP routing process and places the router in router configuration mode.<br>• 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.<br>Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535. |
| **Step 4** | **neighbor** {*ip-address* \| *peer-group-name*} **remote-as** *as-number*<br><br>**Example:**<br>`Router(config-router)# neighbor hh.0.0.1 remote-as 200` | Adds an entry to the BGP or multiprotocol BGP neighbor table.<br>• The *ip-address* argument specifies the IP address of the neighbor.<br>• The *peer-group-name* argument specifies the name of a BGP peer group.<br>• The *as-number* argument specifies the autonomous system to which the neighbor belongs. |
| **Step 5** | **address-family ipv4** [**multicast** \| **unicast** \| **vrf** *vrf-name*]<br><br>**Example:**<br>`Router(config-router)# address-family ipv4` | Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv4 address prefixes.<br>• The **multicast** keyword specifies IPv4 multicast address prefixes.<br>• The **unicast** keyword specifies IPv4 unicast address prefixes.<br>• The **vrf** *vrf-name* keyword and argument specifies the name of the VPN routing/forwarding instance (VRF) to associate with subsequent IPv4 address family configuration mode commands. |
| **Step 6** | **maximum-paths** *number-paths*<br><br>**Example:**<br>`Router(config-router)# maximum-paths 2` | (Optional) Controls the maximum number of parallel routes an IP routing protocol can support.<br>The *number-paths* argument specifies the maximum number of parallel routes an IP routing protocol installs in a routing table, in the range from 1 to 6. |
| **Step 7** | **neighbor** {*ip-address* \| *peer-group-name*} **activate**<br><br>**Example:**<br>`Router(config-router-af)# neighbor hh.0.0.1 activate` | Enables the exchange of information with a neighboring router.<br>• The *ip-address* argument specifies the IP address of the neighbor.<br>• The *peer-group-name* argument specifies the name of a BGP peer group. |
| **Step 8** | **neighbor** *ip-address* **send-label**<br><br>**Example:**<br>`Router(config-router-af)# neighbor hh.0.0.1 send-label` | Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router.<br>• The *ip-address* argument specifies the IP address of the neighboring router. |

| | Command or Action | Purpose |
|---|---|---|
| Step 9 | **exit-address-family**<br><br>**Example:**<br>`Router(config-router-af)# exit-address-family` | Exits from the address family submode. |
| Step 10 | **end**<br><br>**Example:**<br>`Router(config-router-af)# end` | (Optional) Exits to privileged EXEC mode. |

## Configuring the Route Reflectors to Exchange VPNv4 Routes

Perform this task to enable the route reflectors to exchange VPNv4 routes by using multihop, multiprotocol EBGP.

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

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
5. **address-family vpnv4** [**unicast**]
6. **neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]
7. **neighbor** {*ip-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *peer-group-name*} **next-hop unchanged**
9. **exit-address-family**
10. **end**

### DETAILED STEPS

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

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **router bgp** *as-number*<br><br>**Example:**<br>`Router(config)# router bgp 100` | Configures a BGP routing process and places the router in router configuration mode.<br>• 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.<br>Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.<br>The AS number identifies RR1 to routers in other autonomous systems. |
| **Step 4** | **neighbor** {*ip-address* \| *peer-group-name*} **remote-as** *as-number*<br><br>**Example:**<br>`Router(config-router)# neighbor bb.bb.bb.bb remote-as 200` | Adds an entry to the BGP or multiprotocol BGP neighbor table.<br>• The *ip-address* argument specifies the IP address of the neighbor.<br>• The *peer-group-name* argument specifies the name of a BGP peer group.<br>• The *as-number* argument specifies the autonomous system to which the neighbor belongs. |
| **Step 5** | **address-family vpnv4** [**unicast**]<br><br>**Example:**<br>`Router(config-router)# address-family vpnv4` | Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard Virtual Private Network Version 4 (VPNv4) address prefixes.<br>• The optional **unicast** keyword specifies VPNv4 unicast address prefixes. |
| **Step 6** | **neighbor** {*ip-address* \| *peer-group-name*} **ebgp-multihop** [*ttl*]<br><br>**Example:**<br>`Router(config-router-af)# neighbor bb.bb.bb.bb ebgp-multihop 255` | Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.<br>• The *ip-address* argument specifies the IP address of the BGP-speaking neighbor.<br>• The *peer-group-name* argument specifies the name of a BGP peer group.<br>• The *ttl* argument specifies the time-to-live in the range from 1 to 255 hops. |
| **Step 7** | **neighbor** {*ip-address* \| *peer-group-name*} **activate**<br><br>**Example:**<br>`Router(config-router-af)# neighbor bb.bb.bb.bb activate` | Enables the exchange of information with a neighboring router.<br>• The *ip-address* argument specifies the IP address of the neighbor.<br>• The *peer-group-name* argument specifies the name of a BGP peer group. |

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

# Configuring the Route Reflectors to Reflect Remote Routes in Its AS

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

This is accomplished by making the ASBR and PE router route reflector clients of the RR. This procedure also explains how to enable the RR to reflect the VPNv4 routes.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **neighbor** {*ip-address* | *peer-group-name*} **activate**
6. **neighbor** *ip-address* **route-reflector-client**
7. **neighbor** *ip-address* **send-label**
8. **exit-address-family**
9. **address-family vpnv4** [**unicast**]
10. **neighbor** {*ip-address* | *peer-group-name*} **activate**

11. **neighbor** *ip-address* **route-reflector-client**
12. **exit-address-family**
13. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **router bgp** *as-number*<br><br>**Example:**<br>`Router(config)# router bgp 100` | Configures a BGP routing process and places the router in router configuration mode.<br>• 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.<br>Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535. |
| **Step 4** | **address-family ipv4** [**multicast** \| **unicast** \| **vrf** *vrf-name*]<br><br>**Example:**<br>`Router(config-router)# address-family ipv4` | Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv4 address prefixes.<br>• The **multicast** keyword specifies IPv4 multicast address prefixes.<br>• The **unicast** keyword specifies IPv4 unicast address prefixes.<br>• The **vrf** *vrf-name* keyword and argument specifies the name of the VPN routing and forwarding instance (VRF) to associate with subsequent IPv4 address family configuration mode commands. |
| **Step 5** | **neighbor** {*ip-address* \| *peer-group-name*} **activate**<br><br>**Example:**<br>`Router(config-router-af)# neighbor ee.ee.ee.ee activate` | Enables the exchange of information with a neighboring router.<br>• The *ip-address* argument specifies the IP address of the neighbor.<br>• The *peer-group-name* argument specifies the name of a BGP peer group. |

| | Command or Action | Purpose |
|---|---|---|
| Step 6 | **neighbor** *ip-address* **route-reflector-client**<br><br>**Example:**<br>`Router(config-router-af)# neighbor ee.ee.ee.ees route-reflector-client` | Configures the router as a BGP route reflector and configures the specified neighbor as its client.<br>• The *ip-address* argument specifies the IP address of the BGP neighbor being identified as a client. |
| Step 7 | **neighbor** *ip-address* **send-label**<br><br>**Example:**<br>`Router(config-router-af)# neighbor ee.ee.ee.ee send-label` | Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router.<br>• The *ip-address* argument specifies the IP address of the neighboring router. |
| Step 8 | **exit-address-family**<br><br>**Example:**<br>`Router(config-router-af)# exit-address-family` | Exits from the address family submode. |
| Step 9 | **address-family vpnv4** [**unicast**]<br><br>**Example:**<br>`Router(config-router)# address-family vpnv4` | Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.<br>• The optional **unicast** keyword specifies VPNv4 unicast address prefixes. |
| Step 10 | **neighbor** {*ip-address* \| *peer-group-name*} **activate**<br><br>**Example:**<br>`Router(config-router-af)# neighbor ee.ee.ee.ee activate` | Enables the exchange of information with a neighboring router.<br>• The *ip-address* argument specifies the IP address of the neighbor.<br>• The *peer-group-name* argument specifies the name of a BGP peer group. |
| Step 11 | **neighbor** *ip-address* **route-reflector-client**<br><br>**Example:**<br>`Router(config-router-af)# neighbor ee.ee.ee.ee route-reflector-client` | Enables the RR to pass IBGP routes to the neighboring router. |
| Step 12 | **exit-address-family**<br><br>**Example:**<br>`Router(config-router-af)# exit-address-family` | Exits from the address family submode. |
| Step 13 | **end**<br><br>**Example:**<br>`Router(config-router-af)# end` | (Optional) Exits to privileged EXEC mode. |

# Creating Route Maps

The following procedures enable the ASBRs to send MPLS labels with the routes specified in the route maps. Further, the ASBRs accept only the routes that are specified in the route map.

- Configuring a Route Map for Arriving Routes, page 14
- Configuring a Route Map for Departing Routes, page 16

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

Route maps work with access lists. You enter the routes into an access list and then specify the access list when you configure the route map.

## Configuring a Route Map for Arriving Routes

This configuration is optional.

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

### SUMMARY STEPS

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

### DETAILED STEPS

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

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **router bgp** *as-number*<br><br>**Example:**<br>`Router(config)# router bgp 100` | Configures a BGP routing process and places the router in router configuration mode.<br>• 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.<br>Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535. |
| **Step 4** | **route-map** *route-map-name* [**permit** \| **deny**] [*sequence-number*]<br><br>**Example:**<br>`Router(config-router)# route-map IN permit 11` | Creates a route map with the name you specify.<br>• The **permit** keyword allows the actions to happen if all conditions are met.<br>• The **deny** keyword prevents any actions from happening if all conditions are met.<br>• The *sequence-number* argument allows you to prioritize route maps. If you have multiple route maps and want to prioritize them, assign each one a number. The route map with the lowest number is implemented first, followed by the route map with the second lowest number, and so on. |
| **Step 5** | **match ip address** {*access-list-number* \| *access-list-name*} [... *access-list-number* \|... *access-list-name*]<br><br>**Example:**<br>`Router(config-route-map)# match ip address 2` | Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, or performs policy routing on packets.<br>• The *access-list-number* argument is a number of a standard or extended access list. It can be an integer from 1 to 199.<br>• The *access-list-name* argument is a name of a standard or extended access list. It can be an integer from 1 to 199. |
| **Step 6** | **match mpls-label**<br><br>**Example:**<br>`Router(config-route-map)# match mpls-label` | Redistributes routes that include MPLS labels if the routes meet the conditions specified in the route map. |
| **Step 7** | **end**<br><br>**Example:**<br>`Router(config-route-map)# end` | (Optional) Exits to privileged EXEC mode. |

## Configuring a Route Map for Departing Routes

This configuration is optional.

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

### SUMMARY STEPS

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

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **router bgp** *as-number*<br><br>**Example:**<br>`Router(config)# router bgp 100` | Configures a BGP routing process and places the router in router configuration mode.<br>• 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.<br>Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535. |

| | Command or Action | Purpose |
|---|---|---|
| Step 4 | **route-map** *route-map-name* [**permit** \| **deny**] [*sequence-number*]<br><br>**Example:**<br>`Router(config-router)# route-map OUT permit 10` | Creates a route map with the name you specify.<br>• The **permit** keyword allows the actions to happen if all conditions are met.<br>• The **deny** keyword prevents the actions from happening if all conditions are met.<br>• The *sequence-number* argument allows you to prioritize route maps. If you have multiple route maps and want to prioritize them, assign each one a number. The route map with the lowest number is implemented first, followed by the route map with the second lowest number, and so on. |
| Step 5 | **match ip address** {*access-list-number* \| *access-list-name*} [... *access-list-number* \|... *access-list-name*]<br><br>**Example:**<br>`Router(config-route-map)# match ip address` *1* | Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, or performs policy routing on packets.<br>• The *access-list-number* argument is a number of a standard or extended access list. It can be an integer from 1 to 199.<br>• The *access-list-name* argument is a name of a standard or extended access list. It can be an integer from 1 to 199. |
| Step 6 | **set mpls-label**<br><br>**Example:**<br>`Router(config-route-map)# set mpls-label` | Enables a route to be distributed with an MPLS label if the route matches the conditions specified in the route map. |
| Step 7 | **end**<br><br>**Example:**<br>`Router(config-route-map)# end` | Exits to privileged EXEC mode. |

# Applying the Route Maps to the ASBRs

This configuration is optional.

Perform this task to enable the ASBRs to use the route maps.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*}
5. **neighbor** *ip-address* **route-map** *route-map-name* **in**
6. **neighbor** *ip-address* **route-map** *route-map-name* **out**

7. **neighbor** *ip-address* **send-label**
8. **exit-address-family**
9. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **router bgp** *as-number*<br><br>**Example:**<br>`Router(config)# router bgp 100` | Configures a BGP routing process and places the router in router configuration mode.<br>• 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.<br>Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535. |
| **Step 4** | **address-family ipv4** [**multicast** \| **unicast** \| **vrf** *vrf-name*]<br><br>**Example:**<br>`Router(config-router)# address-family ipv4` | Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv4 address prefixes.<br>• The **multicast** keyword specifies IPv4 multicast address prefixes.<br>• The **unicast** keyword specifies IPv4 unicast address prefixes.<br>• The **vrf** *vrf-name* keyword and argument specifies the name of the VRF to associate with subsequent IPv4 address family configuration mode commands. |
| **Step 5** | **neighbor** *ip-address* **route-map** *route-map-name* **in**<br><br>**Example:**<br>`Router(config-router-af)# neighbor` *ip-address* `route-map IN in` | Applies a route map to incoming routes.<br>• The *ip-address* argument specifies the router to which the route map is to be applied.<br>• The *route-map-name* argument specifies the name of the route map.<br>• The **in** keyword applies the route map to incoming routes. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | **neighbor** *ip-address* **route-map** *route-map-name* **out**<br><br>**Example:**<br>`Router(config-router-af)# neighbor ww.ww.ww.ww route-map OUT out` | Applies a route map to outgoing routes.<br>• The *ip-address* argument specifies the router to which the route map is to be applied.<br>• The *route-map-name* argument specifies the name of the route map.<br>• The **out** keyword applies the route map to outgoing routes. |
| **Step 7** | **neighbor** *ip-address* **send-label**<br><br>**Example:**<br>`Router(config-router-af)# neighbor ww.ww.ww.ww send-label` | Advertises the ability of the router to send MPLS labels with routes.<br>• The *ip-address* argument specifies the router that is enabled to send MPLS labels with routes. |
| **Step 8** | **exit-address-family**<br><br>**Example:**<br>`Router(config-router-af)#` **`exit-address-family`** | Exits from the address family submode. |
| **Step 9** | **end**<br><br>**Example:**<br>`Router(config-route-af)# end` | (Optional) Exits to privileged EXEC mode. |

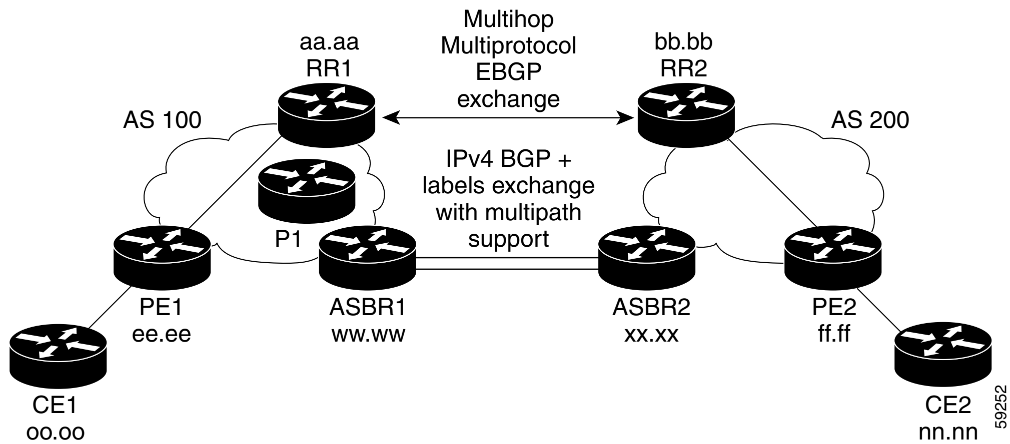
# Verifying the MPLS VPN—Inter-AS—IPv4 BGP Label Distribution Configuration

If you use route reflectors to distribute the VPNv4 routes and use the ASBRs to distribute the IPv4 labels, use the following procedures to help verify the configuration:

- Verifying the Route Reflector Configuration, page 20
- Verifying that CE1 Has Network Reachability Information for CE2, page 21
- Verifying that PE1 Has Network Layer Reachability Information for CE2, page 22
- Verifying that PE2 Has Network Reachability Information for CE2, page 24
- Verifying the ASBR Configuration, page 25

Use Figure 3 as a reference of the configuration.

*Figure 3 Configuring Two VPN Service Providers to Exchange IPv4 Routes and MPLS Labels*




## Verifying the Route Reflector Configuration

Perform this task to verify the route reflector configuration.

### SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**]
3. **disable**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| Step 2 | **show ip bgp vpnv4** {**all** \| **rd** *route-distinguisher* \| **vrf** *vrf-name*} [**summary**] [**labels**]<br><br>**Example:**<br>`Router# show ip bgp vpnv4 all summary`<br><br>**Example:**<br>`Router# show ip bgp vpnv4 all labels` | (Optional) Displays VPN address information from the BGP table.<br>• Use the **show ip bgp vpnv4** command with the **all** and **summary** keywords to verify that a multihop, multiprotocol, EBGP session exists between the route reflectors and that the VPNv4 routes are being exchanged between the route reflectors.<br>The last two lines of the command output show the following information:<br>– Prefixes are being learned from PE1 and then passed to RR2.<br>– Prefixes are being learned from RR2 and then passed to PE1.<br>• Use the **show ip bgp vpnv4** command with the **all** and **labels** keywords to verify that the route reflectors are exchanging VPNv4 label information. |
| Step 3 | **disable**<br><br>**Example:**<br>`Router# disable` | (Optional) Exits to user EXEC mode. |

## Verifying that CE1 Has Network Reachability Information for CE2

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

### SUMMARY STEPS

1. **enable**
2. **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | [protocol [process-id]] | [**list** access-list-number | access-list-name]
3. **disable**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] \| [protocol [process-id]] \| [**list** access-list-number \| access-list-name]<br><br>**Example:**<br>`Router# show ip route nn.nn.nn.nn`<br><br>**Example:**<br>`Router# show ip route` | Displays the current state of the routing table.<br>• Use the **show ip route** command with the *ip-address* argument to verify that CE1 has a route to CE2.<br>• Use the **show ip route** command to verify the routes learned by CE1. Make sure that the route for CE2 is listed. |
| **Step 3** | **disable**<br><br>**Example:**<br>`Router# disable` | (Optional) Exits to user EXEC mode. |

## Verifying that PE1 Has Network Layer Reachability Information for CE2

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

### SUMMARY STEPS

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

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| Step 2 | **show ip route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*] [*tag*] [*output-modifiers*]] [**list** *number* [*output-modifiers*]] [**profile**] [**static** [*output-modifiers*]] [**summary** [*output-modifiers*]] [**supernets-only** [*output-modifiers*]] [**traffic-engineering** [*output-modifiers*]]<br><br>**Example:**<br>`Router# show ip route vrf vpn1 nn.nn.nn.nn` | (Optional) Displays the IP routing table associated with a VRF.<br>• Use the **show ip route vrf** command to verify that router PE1 learns routes from router CE2 (nn.nn.nn.nn). |
| Step 3 | **show ip bgp vpnv4** {**all** \| **rd** *route-distinguisher* \| **vrf** *vrf-name*} [*ip-prefix/length* [**longer-prefixes**] [output-modifiers]] [network-address [mask] [**longer-prefixes**] [output-modifiers]] [**cidr-only**] [community] [**community-list**] [**dampened-paths**] [**filter-list**] [**flap-statistics**] [**inconsistent-as**] [**neighbors**] [**paths** [*line*]] [**peer-group**] [**quote-regexp**] [**regexp**] [**summary**] [**tags**]<br><br>**Example:**<br>`Router# show ip bgp vpnv4 vrf vpn1 nn.nn.nn.nn`<br><br>**Example:**<br>`Router# show ip bgp vpn4 all nn.nn.nn.nn` | (Optional) Displays VPN address information from the BGP table.<br>• Use the **show ip bgp vpnv4** command with the **vrf** or **all** keyword to verify that router PE2 is the BGP next-hop to router CE2. |
| Step 4 | **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]<br><br>**Example:**<br>`Router# show ip cef vrf vpn1 nn.nn.nn.nn` | (Optional) Displays entries in the forwarding information base (FIB) or displays a summary of the FIB.<br>• Use the **show ip cef** command to verify that the Cisco Express Forwarding (CEF) entries are correct. |
| Step 5 | **show mpls forwarding-table** [{*network* {*mask* \| *length*} \| **labels** *label* [- *label*] \| **interface** *interface* \| **next-hop** *address* \| **lsp-tunnel** [*tunnel-id*]}] [**detail**]<br><br>**Example:**<br>`Router# show mpls forwarding-table` | (Optional) Displays the contents of the MPLS forwarding information base (LFIB).<br>• Use the **show mpls forwarding-table** command to verify the IGP label for the BGP next hop router (AS boundary). |
| Step 6 | **show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**]<br><br>**Example:**<br>`Router# show ip bgp ff.ff.ff.ff` | (Optional) Displays entries in the BGP routing table.<br>• Use the **show ip bgp** command to verify the label for the remote egress PE router (PE2). |

| | Command or Action | Purpose |
|---|---|---|
| **Step 7** | **show ip bgp vpnv4** {**all** \| **rd** *route-distinguisher* \| **vrf** *vrf-name*} [**summary**] [**labels**]<br><br>**Example:**<br>`Router# show ip bgp vpnv4 all labels` | (Optional) Displays VPN address information from the BGP table.<br>• Use the **show ip bgp vpnv4 all summary** command to verify the VPN label of CE2, as advertised by PE2. |
| **Step 8** | **disable**<br><br>**Example:**<br>`Router# disable` | (Optional) Exits to user EXEC mode. |

## Verifying that PE2 Has Network Reachability Information for CE2

Perform this task to ensure that PE2 can access CE2.

### SUMMARY STEPS

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

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **show ip route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*] [*tag*] [*output-modifiers*]] [**list** *number* [*output-modifiers*]] [**profile**] [**static** [*output-modifiers*]] [**summary** [*output-modifiers*]] [**supernets-only** [*output-modifiers*]] [**traffic-engineering** [*output-modifiers*]]<br><br>**Example:**<br>`Router# show ip route vrf vpn1 nn.nn.nn.nn` | (Optional) Displays the IP routing table associated with a VRF.<br>• Use the **show ip route vrf** command to check the VPN routing and forwarding table for CE2. The output provides next hop information. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **show mpls forwarding-table** [**vrf** *vpn-name*] [{*network* {*mask* \| *length*} \| **labels** *label* [-*label*] \| **interface** *interface* \| **next-hop** *address* \| **lsp-tunnel** [*tunnel-id*]}] [**detail**]<br><br>**Example:**<br>`Router# show mpls forwarding-table vrf vpn1 nn.nn.nn.nn` | (Optional) Displays the contents of the LFIB.<br>• Use the **show mpls forwarding-table** command with the **vrf** keyword to check the VPN routing and forwarding table for CE2. The output provides the label for CE2 and the outgoing interface. |
| **Step 4** | **show ip bgp vpnv4** {**all** \| **rd** *route-distinguisher* \| **vrf** *vrf-name*} [**summary**] [**labels**]<br><br>**Example:**<br>`Router# show ip bgp vpnv4 all labels` | (Optional) Displays VPN address information from the BGP table.<br>• Use the **show ip bgp vpnv4** command with the **all** and **labels** keywords to check the VPN label for CE2 in the multiprotocol BGP table. |
| **Step 5** | **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]<br><br>**Example:**<br>Router# **show ip cef <*vrf-name*> nn.nn.nn.nn** | (Optional) Displays entries in the FIB or displays a summary of the FIB.<br>• Use the **show ip cef c**ommand to check the CEF entry for CE2. The command output shows the local label for CE2 and the outgoing interface. |
| **Step 6** | **disable**<br><br>**Example:**<br>`Router# disable` | (Optional) Exits to user EXEC mode. |

## Verifying the ASBR Configuration

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

### SUMMARY STEPS

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

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**]<br><br>**Example:**<br>`Router# show ip bgp ff.ff.ff.ff`<br><br>**Example:**<br>`Router# show ip bgp bb.bb.bb.bb` | (Optional) Displays entries in the BGP routing table.<br>• Use the **show ip bgp** command to check that—<br>– ASBR1 receives an MPLS label for PE2 from ASBR2.<br>– ASBR1 received from ASBR2 IPv4 routes for RR2 without labels. If the command output does not display MPLS label information, the route was received without an MPLS label.<br>– ASBR2 distributes an MPLS label for PE2 to ASBR1.<br>– ASBR2 does not distribute a label for RR2 to ASBR1. |
| **Step 3** | **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]<br><br>**Example:**<br>`Router# show ip cef ff.ff.ff.ff`<br><br>**Example:**<br>`Router# show ip cef bb.bb.bb.bb` | (Optional) Displays entries in the FIB or displays a summary of the FIB.<br>• Use the **show ip cef** command from ASBR1 and ASBR2 to check that—<br>– The CEF entry for PE2 is correct.<br>– The CEF entry for RR2 is correct. |
| **Step 4** | **disable**<br><br>**Example:**<br>`Router# disable` | (Optional) Exits to user EXEC mode. |

# Configuration Examples for MPLS VPN—Inter-AS—IPv4 BGP Label Distribution

Configuration examples for MPLS VPN—Inter-AS—IPv4 BGP Label Distribution feature include the following:

- Configuring Inter-AS Using BGP to Distribute Routes and MPLS Labels Over an MPLS VPN Service Provider Example, page 27
- Configuring Inter-AS Using BGP to Distribute Routes and MPLS Labels Over a Non MPLS VPN Service Provider Example, page 33

# Configuring Inter-AS Using BGP to Distribute Routes and MPLS Labels Over an MPLS VPN Service Provider Example

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

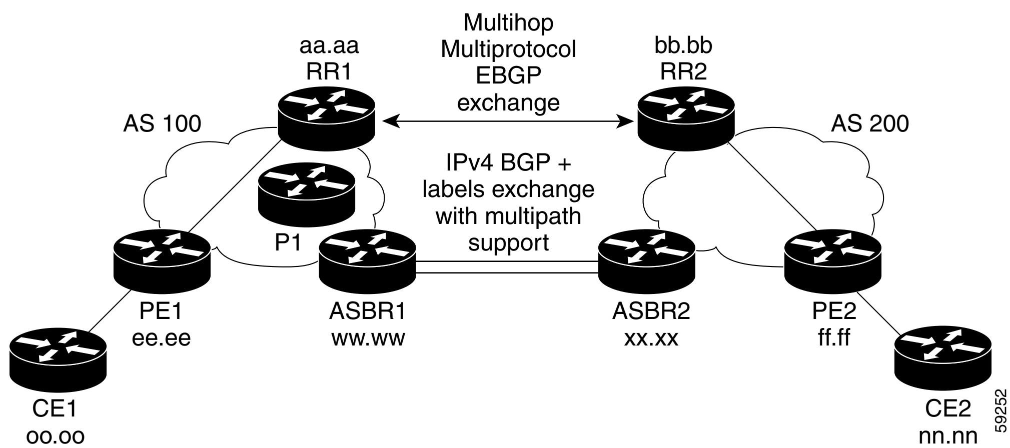
- Route Reflector 1 Configuration Example (MPLS VPN Service Provider), page 28
- ASBR1 Configuration Example (MPLS VPN Service Provider), page 29
- Route Reflector 2 Configuration Example (MPLS VPN Service Provider), page 30
- ASBR2 Configuration Example (MPLS VPN Service Provider), page 31

Figure 4 shows two MPLS VPN service providers. The service provider distributes the VPNv4 routes between the route reflectors. They distribute the IPv4 routes with MPLS labels between the ASBRs.

The configuration example shows the two techniques you can use to distribute the VPNv4 routes and the IPv4 routes with MPLS labels of the remote RRs and PEs to the local RRs and PEs:

- AS 100 uses the RRs to distribute the VPNv4 routes learned from the remote RRs. The RRs also distribute the remote PE address and label learned from ASBR1 using IPv4 + labels.
- In AS 200, the IPv4 routes that ASBR2 learned are redistributed into IGP.

***Figure 4 Distributing IPv4 Routes and MPLS Labels Between MPLS VPN Service Providers***



## Route Reflector 1 Configuration Example (MPLS VPN Service Provider)

The configuration example for RR1 specifies the following:

- RR1 exchanges VPNv4 routes with RR2 using multiprotocol, multihop EBGP.
- The VPNv4 next hop information and the VPN label are preserved across the autonomous systems.
- RR1 reflects to PE1:
  - The VPNv4 routes learned from RR2
  - The IPv4 routes and MPLS labels learned from ASBR1

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address aa.aa.aa.aa 255.255.255.255
 no ip directed-broadcast
!
interface Serial1/2
 ip address dd.0.0.2 255.0.0.0
 no ip directed-broadcast
 clockrate 124061
!
router ospf 10
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 network aa.aa.aa.aa 0.0.0.0 area 100
 network dd.0.0.0 0.255.255.255 area 100
!
router bgp 100
 bgp cluster-id 1
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor ee.ee.ee.ee remote-as 100
 neighbor ee.ee.ee.ee update-source Loopback0
 neighbor ww.ww.ww.ww remote-as 100
 neighbor ww.ww.ww.ww update-source Loopback0
 neighbor bb.bb.bb.bb remote-as 200
 neighbor bb.bb.bb.bb ebgp-multihop 255
 neighbor bb.bb.bb.bb update-source Loopback0
 no auto-summary
 !
address-family ipv4
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee route-reflector-client                !IPv4+labels session to PE1
 neighbor ee.ee.ee.ee send-label
 neighbor ww.ww.ww.ww activate
 neighbor ww.ww.ww.ww route-reflector-client                !IPv4+labels session to ASBR1
 neighbor ww.ww.ww.ww send-label
 no neighbor bb.bb.bb.bb activate
 no auto-summary
 no synchronization
 exit-address-family
 !
address-family vpnv4
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee route-reflector-client                !VPNv4 session with PE1
 neighbor ee.ee.ee.ee send-community extended
 neighbor bb.bb.bb.bb activate
 neighbor bb.bb.bb.bb next-hop-unchanged                    !MH-VPNv4 session with RR2
 neighbor bb.bb.bb.bb send-community extended                !with next hop unchanged
 exit-address-family
```

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

## ASBR1 Configuration Example (MPLS VPN Service Provider)

ASBR1 exchanges IPv4 routes and MPLS labels with ASBR2.

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

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

```
ip subnet-zero
mpls label protocol tdp
!
interface Loopback0
 ip address ww.ww.ww.ww 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Ethernet0/2
 ip address hh.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
interface Ethernet0/3
 ip address dd.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 mpls label protocol ldp
 mpls ip
!
router ospf 10
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Ethernet0/2
 network ww.ww.ww.ww 0.0.0.0 area 100
 network dd.0.0.0 0.255.255.255 area 100

router bgp 100
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor aa.aa.aa.aa remote-as 100
 neighbor aa.aa.aa.aa update-source Loopback0
 neighbor hh.0.0.1 remote-as 200
 no auto-summary
 !
!
```

```
address-family ipv4                        ! Redistributing IGP into BGP
 redistribute ospf 10                      ! so that PE1 & RR1 loopbacks
 neighbor aa.aa.aa.aa activate             ! get into the BGP table
 neighbor aa.aa.aa.aa send-label
 neighbor hh.0.0.1 activate
 neighbor hh.0.0.1 advertisement-interval 5
 neighbor hh.0.0.1 send-label
 neighbor hh.0.0.1 route-map IN in         ! accepting routes in route map IN.
 neighbor hh.0.0.1 route-map OUT out       ! distributing routes in route map OUT.
 neighbor kk.0.0.1 activate
 neighbor kk.0.0.1 advertisement-interval 5
 neighbor kk.0.0.1 send-label
 neighbor kk.0.0.1 route-map IN in         ! accepting routes in route map IN.
 neighbor kk.0.0.1 route-map OUT out       ! distributing routes in route map OUT.
 no auto-summary
 no synchronization
 exit-address-family
!
ip default-gateway 3.3.0.1
ip classless
!
access-list 1 permit ee.ee.ee.ee log                 !Setting up the access lists
access-list 2 permit ff.ff.ff.ff log
access-list 3 permit aa.aa.aa.aa log
access-list 4 permit bb.bb.bb.bb log

route-map IN permit 10                               !Setting up the route maps
 match ip address 2
 match mpls-label
!
route-map IN permit 11
 match ip address 4
!
route-map OUT permit 12
 match ip address 3
!
route-map OUT permit 13
 match ip address 1
 set mpls-label
!
end
```

## Route Reflector 2 Configuration Example (MPLS VPN Service Provider)

RR2 exchanges VPNv4 routes with RR1 through multihop, multiprotocol EBGP. This configuration also specifies that the next hop information and the VPN label are preserved across the autonomous systems.

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address bb.bb.bb.bb 255.255.255.255
 no ip directed-broadcast
!
interface Serial1/1
 ip address ii.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
router ospf 20
 log-adjacency-changes
 network bb.bb.bb.bb 0.0.0.0 area 200
```

```
 network ii.0.0.0 0.255.255.255 area 200
!
router bgp 200
 bgp cluster-id 1
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor aa.aa.aa.aa remote-as 100
 neighbor aa.aa.aa.aa ebgp-multihop 255
 neighbor aa.aa.aa.aa update-source Loopback0
 neighbor ff.ff.ff.ff remote-as 200
 neighbor ff.ff.ff.ff update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor aa.aa.aa.aa activate
 neighbor aa.aa.aa.aa next-hop-unchanged              !Multihop VPNv4 session with RR1
 neighbor aa.aa.aa.aa send-community extended              !with next-hop-unchanged
 neighbor ff.ff.ff.ff activate
 neighbor ff.ff.ff.ff route-reflector-client          !VPNv4 session with PE2
 neighbor ff.ff.ff.ff send-community extended
 exit-address-family
!
ip default-gateway 3.3.0.1
no ip classless
!
end
```

## ASBR2 Configuration Example (MPLS VPN Service Provider)

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

```
ip subnet-zero
ip cef
!
mpls label protocol tdp
!
interface Loopback0
 ip address xx.xx.xx.xx 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet1/0
 ip address hh.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
interface Ethernet1/2
 ip address jj.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 mpls label protocol tdp
 mpls ip
 !
router ospf 20
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 redistribute bgp 200 subnets           ! Redistributing the routes learned from
 passive-interface Ethernet1/0               ! ASBR1(EBGP+labels session) into IGP
 network xx.xx.xx.xx 0.0.0.0 area 200         ! so that PE2 will learn them
 network jj..0.0 0.255.255.255 area 200
```

```
 !
router bgp 200
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor bb.bb.bb.bb remote-as 200
 neighbor bb.bb.bb.bb update-source Loopback0
 neighbor hh.0.0.2 remote-as 100
 no auto-summary
 !
address-family ipv4
 redistribute ospf 20                          ! Redistributing IGP into BGP
 neighbor hh.0.0.2 activate                    ! so that PE2 & RR2 loopbacks
 neighbor hh.0.0.2 advertisement-interval 5    ! will get into the BGP-4 table.
 neighbor hh.0.0.2 route-map IN in
 neighbor hh.0.0.2 route-map OUT out
 neighbor hh.0.0.2 send-label
 neighbor kk.0.0.2 activate
 neighbor kk.0.0.2 advertisement-interval 5
 neighbor kk.0.0.2 route-map IN in
 neighbor kk.0.0.2 route-map OUT out
 neighbor kk.0.0.2 send-label
 no auto-summary
 no synchronization
 exit-address-family
!
address-family vpnv4
 neighbor bb.bb.bb.bb activate
 neighbor bb.bb.bb.bb send-community extended
 exit-address-family
 !
ip default-gateway 3.3.0.1
ip classless
!
access-list 1 permit ff.ff.ff.ff log          !Setting up the access lists
access-list 2 permit ee.ee.ee.ee log
access-list 3 permit bb.bb.bb.bb log
access-list 4 permit aa.aa.aa.aa log

route-map IN permit 11                        !Setting up the route maps
 match ip address 2
 match mpls-label
!
route-map IN permit 12
 match ip address 4
!
route-map OUT permit 10
 match ip address 1
 set mpls-label
!
route-map OUT permit 13
 match ip address 3
end
```

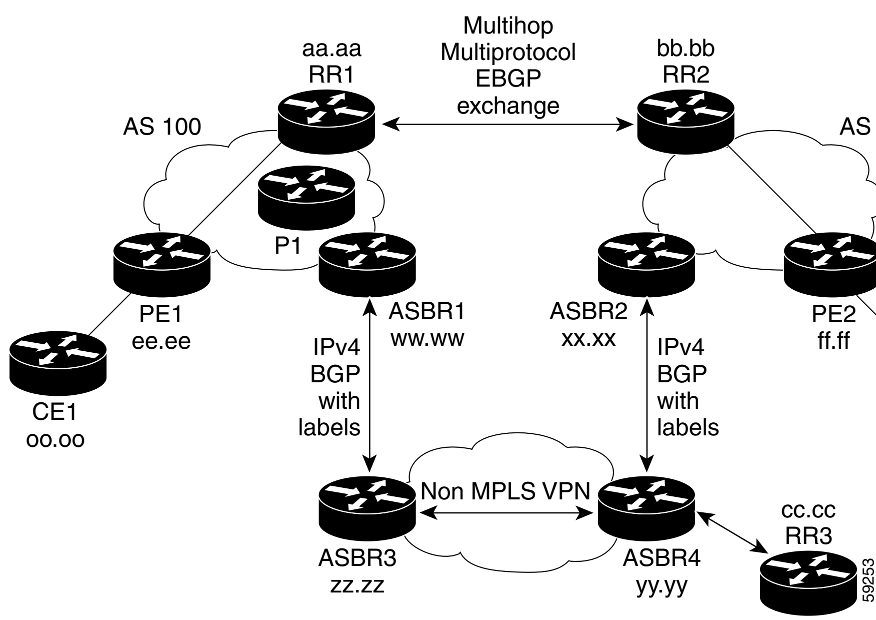
## Configuring Inter-AS Using BGP to Distribute Routes and MPLS Labels Over a Non MPLS VPN Service Provider Example

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


- Route Reflector 1 Configuration Example (Non MPLS VPN Service Provider), page 34
- ASBR1 Configuration Example (Non MPLS VPN Service Provider), page 35
- Route Reflector 2 Configuration Example (Non MPLS VPN Service Provider), page 36
- ASBR2 Configuration Example (Non MPLS VPN Service Provider), page 37
- ASBR3 Configuration Example (Non MPLS VPN Service Provider), page 38
- Route Reflector 3 Configuration Example (Non MPLS VPN Service Provider), page 40
- ASBR4 Configuration Example (Non MPLS VPN Service Provider), page 41


Figure 5 shows two MPLS VPN service providers that are connected through a non MPLS VPN service provider. The autonomous system in the middle of the network is configured as a backbone autonomous system that uses Label Distribution Protocol (LDP) or Tag Distribution Protocol (TDP) to distribute MPLS labels. You can also use traffic engineering tunnels instead of TDP or LDP to build the LSP across the non MPLS VPN service provider.

***Figure 5 Distributing Routes and MPLS Labels Over a Non MPLS VPN Service Provider***



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

The configuration example for RR1 specifies the following:

- RR1 exchanges VPNv4 routes with RR2 using multiprotocol, multihop EBGP.
- The VPNv4 next hop information and the VPN label are preserved across the autonomous systems.
- RR1 reflects to PE1:
  - The VPNv4 routes learned from RR2
  - The IPv4 routes and MPLS labels learned from ASBR1

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address aa.aa.aa.aa 255.255.255.255
 no ip directed-broadcast
!
interface Serial1/2
 ip address dd.0.0.2 255.0.0.0
 no ip directed-broadcast
 clockrate 124061
!
router ospf 10
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 network aa.aa.aa.aa 0.0.0.0 area 100
 network dd.dd.0.0.0 0.255.255.255 area 100
!
router bgp 100
 bgp cluster-id 1
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor ee.ee.ee.ee remote-as 100
 neighbor ee.ee.ee.ee update-source Loopback0
 neighbor ww.ww.ww.ww remote-as 100
 neighbor ww.ww.ww.ww update-source Loopback0
 neighbor bb.bb.bb.bb remote-as 200
 neighbor bb.bb.bb.bb ebgp-multihop 255
 neighbor bb.bb.bb.bb update-source Loopback0
 no auto-summary
 !
address-family ipv4
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee route-reflector-client                !IPv4+labels session to PE1
 neighbor ee.ee.ee.ee send-label
 neighbor ww.ww.ww.ww activate
 neighbor ww.ww.ww.ww route-reflector-client                !IPv4+labels session to ASBR1
 neighbor ww.ww.ww.ww send-label
 no neighbor bb.bb.bb.bb activate
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee route-reflector-client               !VPNv4 session with PE1
 neighbor ee.ee.ee.ee send-community extended
 neighbor bb.bb.bb.bb activate
 neighbor bb.bb.bb.bb next-hop-unchanged                    !MH-VPNv4 session with RR2
 neighbor bb.bb.bb.bb send-community extended                 with next-hop-unchanged
 exit-address-family
```

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

## ASBR1 Configuration Example (Non MPLS VPN Service Provider)

ASBR1 exchanges IPv4 routes and MPLS labels with ASBR2.

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

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

```
ip subnet-zero
ip cef distributed
mpls label protocol tdp
!
interface Loopback0
 ip address ww.ww.ww.ww 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Serial3/0/0
 ip address kk.0.0.2 255.0.0.0
 no ip directed-broadcast
 ip route-cache distributed
!
interface Ethernet0/3
 ip address dd.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 mpls label protocol ldp
 mpls ip
!
router ospf 10
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Serial3/0/0
 network ww.ww.ww.ww 0.0.0.0 area 100
 network dd.0.0.0 0.255.255.255 area 100

router bgp 100
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor aa.aa.aa.aa remote-as 100
 neighbor aa.aa.aa.aa update-source Loopback0
 neighbor kk.0.0.1 remote-as 200
 no auto-summary
!
 address-family ipv4
```

```
 redistribute ospf 10                      ! Redistributing IGP into BGP
 neighbor aa.aa.aa.aa activate             ! so that PE1 & RR1 loopbacks
 neighbor aa.aa.aa.aa send-label           ! get into BGP table
 neighbor kk.0.0.1 activate
 neighbor kk.0.0.1 advertisement-interval 5
 neighbor kk.0.0.1 send-label
 neighbor kk.0.0.1 route-map IN in    ! Accepting routes specified in route map IN
 neighbor kk.0.0.1 route-map OUT out  ! Distributing routes specified in route map OUT
 no auto-summary
 no synchronization
 exit-address-family
!
ip default-gateway 3.3.0.1
ip classless
!
access-list 1 permit ee.ee.ee.ee log
access-list 2 permit ff.ff.ff.ff log
access-list 3 permit aa.aa.aa.aa log
access-list 4 permit bb.bb.bb.bb log
!
route-map IN permit 10
 match ip address 2
 match mpls-label
!
route-map IN permit 11
 match ip address 4
!
route-map OUT permit 12
 match ip address 3
!
route-map OUT permit 13
 match ip address 1
 set mpls-label
!
end
```

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

RR2 exchanges VPNv4 routes with RR1 using multihop, multiprotocol EBGP. This configuration also specifies that the next hop information and the VPN label are preserved across the autonomous systems.

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address bb.bb.bb.bb 255.255.255.255
 no ip directed-broadcast
!
interface Serial1/1
 ip address ii.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
router ospf 20
 log-adjacency-changes
 network bb.bb.bb.bb 0.0.0.0 area 200
 network ii.0.0.0 0.255.255.255 area 200
!
router bgp 200
 bgp cluster-id 1
 bgp log-neighbor-changes
 timers bgp 10 30
```

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

## ASBR2 Configuration Example (Non MPLS VPN Service Provider)

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

```
ip subnet-zero
ip cef
!
mpls label protocol tdp
!
interface Loopback0
 ip address xx.xx.xx.xx 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0/1
 ip address qq.0.0.2 255.0.0.0
 no ip directed-broadcast
!
interface Ethernet1/2
 ip address jj.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 mpls label protocol tdp
 mpls ip
 !
router ospf 20
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 redistribute bgp 200 subnets            !redistributing the routes learned from
 passive-interface Ethernet0/1                !ASBR2 (EBGP+labels session) into IGP
 network xx.xx.xx.xx 0.0.0.0 area 200         !so that PE2 will learn them
 network jj.0.0.0 0.255.255.255 area 200
 !
router bgp 200
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor bb.bb.bb.bb remote-as 200
 neighbor bb.bb.bb.bb update-source Loopback0
 neighbor qq.0.0.1 remote-as 100
```

```
 no auto-summary
!
address-family ipv4           ! Redistributing IGP into BGP
redistribute ospf 20                        ! so that PE2 & RR2 loopbacks
 neighbor qq.0.0.1 activate                 !  will get into the BGP-4 table
 neighbor qq.0.0.1 advertisement-interval 5
 neighbor qq.0.0.1 route-map IN in
 neighbor qq.0.0.1 route-map OUT out
 neighbor qq.0.0.1 send-label
 no auto-summary
 no synchronization
 exit-address-family
 !
address-family vpnv4
 neighbor bb.bb.bb.bb activate
 neighbor bb.bb.bb.bb send-community extended
 exit-address-family
 !
ip default-gateway 3.3.0.1
ip classless
!
access-list 1 permit ff.ff.ff.ff log
access-list 2 permit ee.ee.ee.ee log
access-list 3 permit bb.bb.bb.bb log
access-list 4 permit aa.aa.aa.aa log
!
route-map IN permit 11
 match ip address 2
 match mpls-label
!
route-map IN permit 12
 match ip address 4
!
route-map OUT permit 10
 match ip address 1
 set mpls-label
!
route-map OUT permit 13
 match ip address 3
!
end
```

## ASBR3 Configuration Example (Non MPLS VPN Service Provider)

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

**Note** Do not redistribute EBGP routes learned into IBG if you are using IBGP to distribute the routes and labels. This is not a supported configuration.

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address yy.yy.yy.yy 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
```

```
interface Hssi4/0
 ip address mm.0.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 mpls ip
 hssi internal-clock
 !
interface Serial5/0
 ip address kk.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 load-interval 30
 clockrate 124061
!
router ospf 30
log-adjacency-changes
auto-cost reference-bandwidth 1000
redistribute connected subnets
network yy.yy.yy.yy 0.0.0.0 area 300
network mm.0.0.0 0.255.255.255 area 300
!
router bgp 300
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor cc.cc.cc.cc remote-as 300
 neighbor cc.cc.cc.cc update-source Loopback0
 neighbor kk.0.0.2 remote-as 100
 no auto-summary
 !
 address-family ipv4
 neighbor cc.cc.cc.cc activate            ! IBGP+labels session with RR3
 neighbor cc.cc.cc.cc send-label
 neighbor kk.0.0.2 activate               ! EBGP+labels session with ASBR1
 neighbor kk.0.0.2 advertisement-interval 5
 neighbor kk.0.0.2 send-label
neighbor kk.0.0.2 route-map IN in
neighbor kk.0.0.2 route-map OUT out
no auto-summary
 no synchronization
 exit-address-family
!
ip classless
!
access-list 1 permit ee.ee.ee.ee log
access-list 2 permit ff.ff.ff.ff log
access-list 3 permit aa.aa.aa.aa log
access-list 4 permit bb.bb.bb.bb log
!
route-map IN permit 10
 match ip address 1
  match mpls-label
!
route-map IN permit 11
   match ip address 3
!
route-map OUT permit 12
 match ip address 2
  set mpls-label
!
route-map OUT permit 13
   match ip address 4
!
```

```
ip default-gateway 3.3.0.1
ip classless
!
end
```

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

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

```
ip subnet-zero
mpls label protocol tdp
mpls traffic-eng auto-bw timers
no mpls ip
!
interface Loopback0
 ip address cc.cc.cc.cc 255.255.255.255
 no ip directed-broadcast
!
interface POS0/2
 ip address pp.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip route-cache cef
 no ip route-cache
 no ip mroute-cache
 crc 16
 clock source internal
!
router ospf 30
 log-adjacency-changes
 network cc.cc.cc.cc 0.0.0.0 area 300
 network pp.0.0.0 0.255.255.255 area 300
!
router bgp 300
 bgp log-neighbor-changes
 neighbor zz.zz.zz.zz remote-as 300
 neighbor zz.zz.zz.zz update-source Loopback0
 neighbor yy.yy.yy.yy remote-as 300
 neighbor yy.yy.yy.yy update-source Loopback0
 no auto-summary
 !
address-family ipv4
 neighbor zz.zz.zz.zz activate
 neighbor zz.zz.zz.zz route-reflector-client
 neighbor zz.zz.zz.zz send-label              ! IBGP+labels session with ASBR3
 neighbor yy.yy.yy.yy activate
 neighbor yy.yy.yy.yy route-reflector-client
 neighbor yy.yy.yy.yy send-label              ! IBGP+labels session with ASBR4
 no auto-summary
 no synchronization
 exit-address-family
!
ip default-gateway 3.3.0.1
ip classless
!
end
```

## ASBR4 Configuration Example (Non MPLS VPN Service Provider)

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

**Note** Do not redistribute EBGP routes learned into IBG if you are using IBGP to distribute the routes and labels. This is not a supported configuration.

```
ip subnet-zero
ip cef distributed
!
interface Loopback0
 ip address zz.zz.zz.zz 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Ethernet0/2
 ip address qq.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
interface POS1/1/0
 ip address pp.0.0.2 255.0.0.0
 no ip directed-broadcast
 ip route-cache distributed
 !
interface Hssi2/1/1
 ip address mm.0.0.2 255.0.0.0
 no ip directed-broadcast
 ip route-cache distributed
 no ip mroute-cache
 mpls label protocol tdp
 mpls ip
 hssi internal-clock
!
router ospf 30
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
passive-interface Ethernet0/2
 network zz.zz.zz.zz 0.0.0.0 area 300
 network pp.0.0.0 0.255.255.255 area 300
 network mm.0.0.0 0.255.255.255 area 300
 !
router bgp 300
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor cc.cc.cc.cc remote-as 300
 neighbor cc.cc.cc.cc update-source Loopback0
 neighbor qq.0.0.2 remote-as 200
 no auto-summary
 !
 address-family ipv4
 neighbor cc.cc.cc.cc activate
 neighbor cc.cc.cc.cc send-label
 neighbor qq.0.0.2 activate
 neighbor qq.0.0.2 advertisement-interval 5
 neighbor qq.0.0.2 send-label
 neighbor qq.0.0.2 route-map IN in
 neighbor qq.0.0.2 route-map OUT out
```

```
no auto-summary
 no synchronization
 exit-address-family
!
ip classless
!
access-list 1 permit ff.ff.ff.ff log
access-list 2 permit ee.ee.ee.ee log
access-list 3 permit bb.bb.bb.bb log
access-list 4 permit aa.aa.aa.aa log
!
route-map IN permit 10
 match ip address 1
  match mpls-label
!
route-map IN permit 11
   match ip address 3
!
route-map OUT permit 12
 match ip address 2
  set mpls-label
!
route-map OUT permit 13
   match ip address 4
!
ip default-gateway 3.3.0.1
ip classless
!
end
```

# Additional References

The following sections provide references related to MPLS VPN—Inter-AS IPv4 BGP Label Distribution.

## Related Documents

| Related Topic | Document Title |
|---|---|
| MPLS VPN Interautonomous systems configuration tasks | *MPLS VPN—Interautonomous System Support* |
| VPN configuration tasks | *MPLS Virtual Private Networks (VPNs)* |
| An explanation of how BGP works and how you can use it to participate in routing with other networks that run BGP | *Using the Border Gateway Protocol for Interdomain Routing* |
| BGP configuration tasks | "Configuring BGP" chapter in the *Cisco IOS IP Configuration Guide*, Release 12.2 |
| An explanation of the purpose of BGP and the BGP route selection process, and how to use BGP attributes in route selection | "Border Gateway Protocol" chapter in the *Internetworking Technology Overview* |
| MPLS configuration tasks | "Configuring Multiprotocol Label Switching" chapter in the *Cisco IOS Switching Services Configuration Guide*, Release 12.2 |
| Commands to configure and monitor BGP | "BGP Commands" chapter in the *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols*, Release 12.2 |
| Explicit null labels | MPLS VPN—Explicit Null Label Support with BGP IPv4 Label Session |

## Standards

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

## MIBs

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

## RFCs

| RFCs | Title |
|---|---|
| RFC 1700 | *Assigned Numbers* |
| RFC 1966 | *BGP Route Reflection: An Alternative to Full Mesh IBGP* |
| RFC 2842 | *Capabilities Advertisement with BGP-4* |
| RFC 2858 | *Multiprotocol Extensions for BGP-4* |
| RFC 3107 | *Carrying Label Information in BGP-4* |

## Technical Assistance

| Description | Link |
|---|---|
| Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content. | http://www.cisco.com/public/support/tac/home.shtml |

# Command Reference

None.

# Glossary

**AS**—autonomous system. A collection of networks that share the same routing protocol and that are under the same system administration.

**ASBR**—autonomous system boundary router. A router that connects and exchanges information between two or more autonomous systems.

**BGP**—Border Gateway Protocol. The exterior border gateway protocol used to exchange routing information between routers in separate autonomous systems. BGP uses Transmission Control Protocol (TCP). Because TCP is a reliable protocol, BGP does not experience problems with dropped or fragmented data packets.

**CE router**—customer edge router. The customer router that connects to the provider edge (PE) router.

**EBGP**—External Border Gateway Protocol. A BGP session between routers in different autonomous systems (ASs). When a pair of routers in different ASs are more than one IP hop away from each other, an EBGP session between those two routers is called multihop EBGP.

**IBGP**—Internal Border Gateway Protocol. A BGP session between routers within the same autonomous system.

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

**LDP**—Label Distribution Protocol. A standard protocol between MPLS-enabled routers to negotiate the labels (addresses) used to forward packets. This protocol is not supported in Cisco IOS Release 12.0. The Cisco proprietary version of this protocol is the Tag Distribution Protocol (TDP).

**LER**—label edge router. The edge router that performs label imposition and disposition.

**LSR**—label switch router. The role of an LSR is to forward packets in an MPLS network by looking only at the fixed-length label.

**NLRI**—Network Layer Reachability Information. BGP sends routing update messages containing NLRI, which describes the route. In this context, an NLRI is a prefix. A BGP update message carries one or more NLRI prefixes and the attributes of a route for the NLRI prefixes. The route attributes include a BGP next hop gateway address, community values, and other information.

**P router**—provider router. The core router in the service provider network that connects to provider edge (PE) routers. In a packet-switched star topology, a router that is part of the backbone and that serves as the single pipe through which all traffic from peripheral networks must pass on its way to other peripheral networks.

**PE router**—provider edge router. The label edge router (LER) in the service provider network that connects to the customer edge (CE) router.

**RR**—route reflector. A router that advertises or reflects IBGP learned routes to other IBGP peers without requiring a full network mesh.

**VPN**—Virtual Private Network. A group of sites that, as a result of a set of administrative policies, can communicate with each other over a shared backbone.

**VPNv4 addresses**—When multiple VPNs use the same address space, the VPN addresses are made unique by adding a route distinguisher to the front of the address.

**Note** Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.