

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

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

- Prerequisites for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels, on page 1
- Restrictions for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels, on page 3
- Information About MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels, on page 3
- How to Configure MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels, on page 6
- Configuration Examples for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels, on page 20
- Additional References, on page 33
- Feature Information for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels, on page 35

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

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

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

Table 1: Cisco 12000 Series Line Card Support in Cisco IOS S Releases

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

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

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

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

MPLS VPN Inter-AS Introduction

An autonomous system is a single network or group of networks that is controlled by a common system administration group and that uses a single, clearly defined routing protocol.

As VPNs grow, their requirements expand. In some cases, VPNs need to reside on different autonomous systems in different geographic areas. Also, some VPNs need to extend across multiple service providers (overlapping VPNs). Regardless of the complexity and location of the VPNs, the connection between autonomous systems must be seamless to the customer.

Benefits of MPLS VPN Inter-AS

An MultiprotocolLabel Switching (MPLS) VPN Inter-AS provides the following benefits:

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

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

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

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

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

- Saves the ASBRs from having to store all the VPN-IPv4 routes. Using the route reflectors to store the VPN-IPv4 routes and forward them to the PE routers results in improved scalability compared wtih configurations where the ASBR holds all of the VPN-IPv4 routes and forwards the routes based on VPN-IPv4 labels.
- Simplifies the configuration at the border of the network by having the route reflectors hold the VPN-IPv4 routes.
- Enables a non-VPN core network to act as a transit network for VPN traffic. You can transport IPv4 routes with MPLS labels over a non-MPLS VPN service provider.
- Eliminates the need for any other label distribution protocol between adjacent LSRs. If two adjacent label switch routers (LSRs) are also BGP peers, BGP can handle the distribution of the MPLS labels. No other label distribution protocol is needed between the two LSRs.

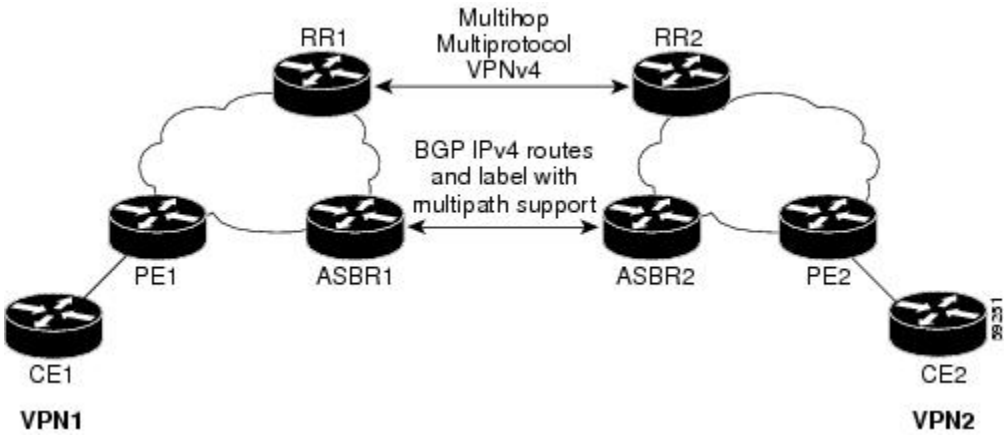
How the Inter-AS Works When ASBRs Exchange IPv4 Routes with MPLS Labels

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

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

Alternatively, the route reflector can reflect the IPv4 routes and MPLS labels learned from the ASBR to the PE routers in the VPN. This is accomplished by the ASBR exchanging IPv4 routes and MPLS labels with the route reflector. The route reflector also reflects the VPN-IPv4 routes to the PE routers in the VPN. For example,

in VPN1 of the figure below, RR1 reflects to PE1 the VPN-IPv4 routes it learned and IPv4 routes and MPLS labels learned from ASBR1. Using the route reflectors to store the VPN-IPv4 routes and forward them through the PE routers and ASBRs allows for a scalable configuration.



BGP Routing Information

BGP routing information includes the following items:

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

Types of BGP Messages and MPLS Labels

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

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

How BGP Sends MPLS Labels with Routes

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

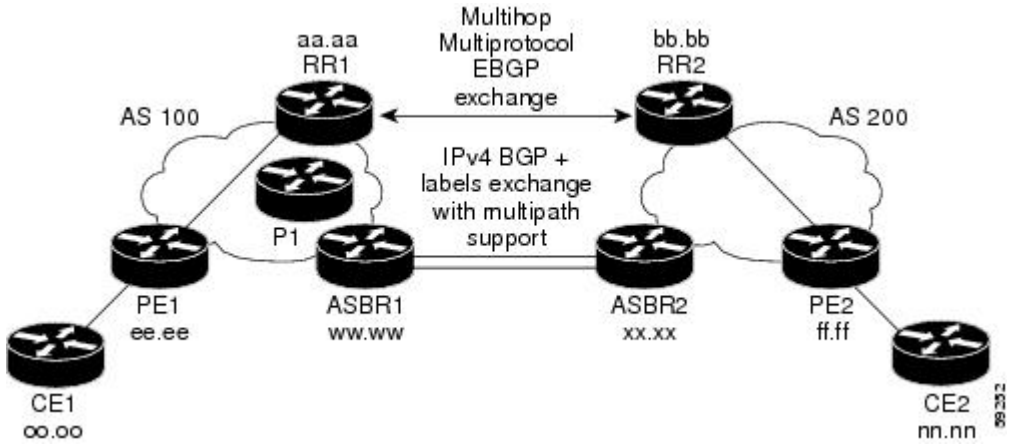
When you issue the **neighbor send-label** command on both BPG routers, the routers advertise to each other that they can then send MPLS labels with the routes. If the routers successfully negotiate their ability to send MPLS labels, the routers add MPLS labels to all outgoing BGP updates.

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

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

The figure below shows the following sample configuration:

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



Configuring the ASBRs to Exchange IPv4 Routes and MPLS Labels

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*

4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
5. **address-family ipv4** [**multicast** | **unicast** | **mdt** | **vrf** *vrf-name*]
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **neighbor***ip-address* **send-label**
8. **exit-address-family**
9. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and places the router in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor hh.0.0.1 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 5	**address-family ipv4** [**multicast** | **unicast** | **mdt** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4`	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv4 address prefixes. • The **multicast** keyword specifies IPv4 multicast address prefixes.

	Command or Action	Purpose
		• The **unicast** keyword specifies IPv4 unicast address prefixes. • The **mdt** keyword specifies an IPv4 multicast distribution tree (MDT) address family session. • The **vrf** *vrf-name* keyword and argument specify the name of the VPN routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor hh.0.0.1 activate`	Enables the exchange of information with a neighboring router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**neighbor***ip-address* **send-label** **Example:** `Router(config-router-af)# neighbor hh.0.0.1 send-label`	Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighboring router.
Step 8	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 9	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Configuring the Route Reflectors to Exchange VPN-IPv4 Routes

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

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*

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

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and places the router in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535. The autonomous system number identifies RR1 to routers in other autonomous systems.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor bb.bb.bb.bb remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]	Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.

	Command or Action	Purpose
	Example: `Router(config-router)# neighbor bb.bb.bb.bb ebgp-multihop 255`	• The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *ttl* argument specifies the time-to-live in the range from 1 to 255 hops.
Step 6	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP sessions, that use standard VPNv4 address prefixes. • The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor bb.bb.bb.bb activate`	Enables the exchange of information with a neighboring router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **next-hop unchanged** **Example:** `Router(config-router-af)# neighbor ip-address next-hop unchanged`	Enables an eBGP multihop peer to propagate the next hop unchanged. • The *ip-address* argument specifies the IP address of the next hop. • The *peer-group-name* argument specifies the name of a BGP peer group that is the next hop.
Step 9	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 10	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and places the router in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**address-family ipv4** [**multicast**	**unicast**	**vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4`	Enters address family configuration mode for configuring routing sessions, such as BGP sessions, that use standard IPv4 address prefixes.

	Command or Action	Purpose
		• The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**neighbor** {*ip-address* | *peer-group-name* **activate** **Example:** `Router(config-router-af)# neighbor ee.ee.ee.ee activate`	Enables the exchange of information with a neighboring router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 6	**neighbor** *ip-address* **route-reflector-client** **Example:** `Router(config-router-af)# neighbor ee.ee.ee.ees route-reflector-client`	Configures the router as a BGP route reflector and configures the specified neighbor as its client. • The *ip-address* argument specifies the IP address of the BGP neighbor being configured as a client.
Step 7	**neighbor** *ip-address* **send-label** **Example:** `Router(config-router-af)# neighbor ee.ee.ee.ee send-label`	Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighboring router.
Step 8	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 9	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP sessions, that use standard VPNv4 address prefixes. • The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 10	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor ee.ee.ee.ee activate`	Enables the exchange of information with a neighboring router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.

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

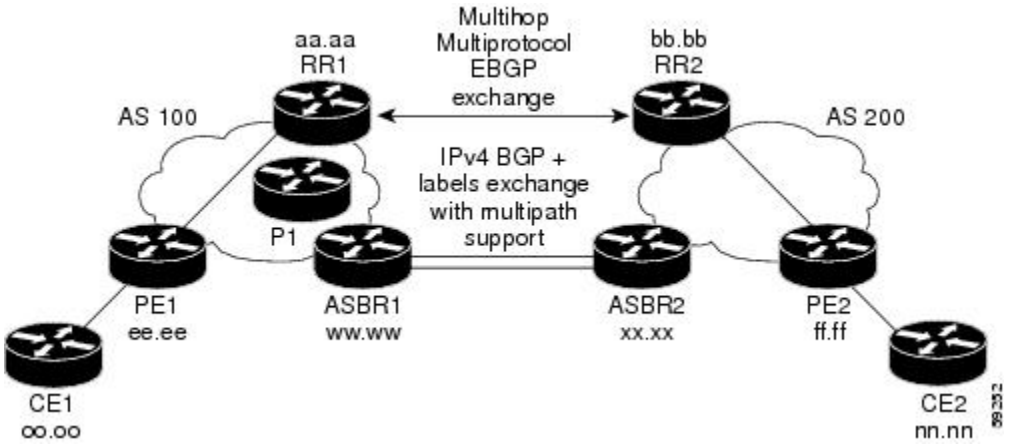
Step 12	**exit-address-family**
	Example:
	`Router(config-router-af)#`
	`exit-address-family`
		Exits address family configuration mode.

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

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

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

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



Verifying the Route Reflector Configuration

Perform this task to verify the route reflector configuration.

SUMMARY STEPS

1. **enable**

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

DETAILED STEPS

Procedure

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

Verifying that CE1 Can Communicate with CE2

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

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

Verifying that PE1 Can Communicate with CE2

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*] [*tag*] [*output-modifiers*]] [**list** *number* [*output-modifiers*]] [**profile**] [**static** [[]] [**summary***output-modifiers*]] [**supernets-only** [*output-modifiers*]] [**traffic-engineering** [*output-modifiers*]] **Example:** `Router# show ip route vrf vpn1 nn.nn.nn.nn`	(Optional) Displays the IP routing table associated with a VRF. • Use this command to verify that router PE1 learns routes from router CE2 (nn.nn.nn.nn).

Step 3	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [*ip-prefix* | *length* [**longer-prefixes**] [*output-modifiers*]]] [**network-address** *mask*]] **longer-prefixes** [*output-modifiers*]] [**cidr-only**] [**community**] [**community-list**] [**dampened-paths**] [**filter-list**] [**flap-statistics**] [**inconsistent-as**] [**neighbors**] [**paths** [*line*]] [**peer-group**] [**quote-regexp**] [**regexp**] [**summary**] [**tags**]
	Example:
	`Router# show ip bgp vpnv4 vrf vpn1 nn.nn.nn.nn`
	Example:
	`Router# show ip bgp vpnv4 all nn.nn.nn.nn`
		(Optional) Displays VPN address information from the BGP table.
		• Use the **vrf** or **all** keyword to verify that router PE2 is the BGP next-hop to router CE2.

Step 4	**show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**] **Example:** `Router# show ip cef vrf vpn1 nn.nn.nn.nn`	(Optional) Displays entries in the Forwarding Information Base (FIB) or displays a summary of the FIB. • Use this command to verify that the Cisco Express Forwarding entries are correct.
Step 5	**show mpls forwarding-table** [{*network* {*mask* | *length*} | **labels** *label* [*-label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:** `Router# show mpls forwarding-table`	(Optional) Displays the contents of the MPLS LFIB. • Use this command to verify the IGP label for the BGP next hop router (autonomous system boundary).
Step 6	**show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**]	(Optional) Displays entries in the BGP routing table.

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

Verifying that PE2 Can Communicate with CE2

Perform this task to ensure that PE2 can access CE2.

SUMMARY STEPS

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

DETAILED STEPS

Procedure

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

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

Verifying the ASBR Configuration

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

Verifying the ASBR Configuration

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

Procedure

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

Step 2	**show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**]
	Example:
	`Router# show ip bgp ff.ff.ff.ff`
		(Optional) Displays entries in the BGP routing table.
		• Use this command to check that:
		  • ASBR1 receives an MPLS label for PE2 from ASBR2.
		  • ASBR1 receives IPv4 routes for RR2 without labels from ASBR2.
		  • ASBR2 distributes an MPLS label for PE2 to ASBR1.
		  • ASBR2 does not distribute a label for RR2 to ASBR1.

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

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

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

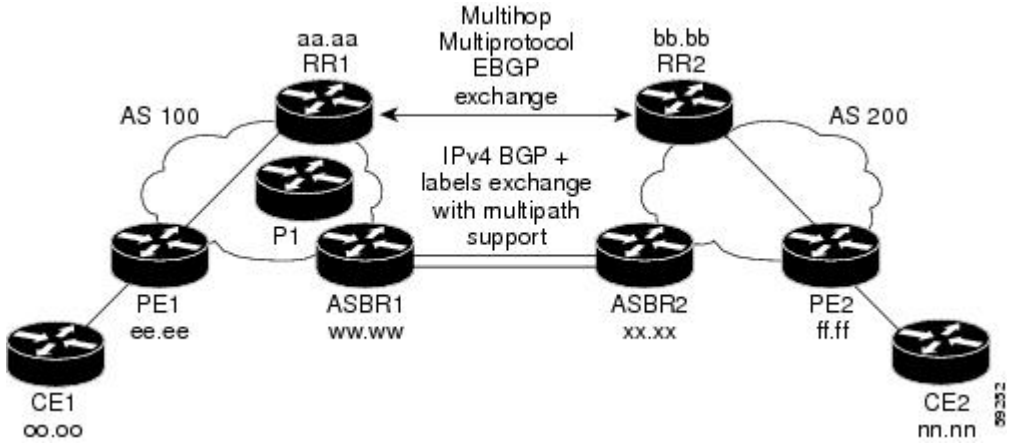
Configuring MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels over an MPLS VPN Service Provider Examples

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

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

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

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



Route Reflector 1 Configuration Example (MPLS VPN Service Provider)

The configuration example for RR1 specifies the following:

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

```
ip subnet-zero
```

```
ip cef
!
interface Loopback0
 ip address aa.aa.aa.aa 255.255.255.255
!
interface Ethernet0/3
 ip address dd.0.0.2 255.0.0.0
!
router ospf 10
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 network aa.aa.aa.aa 0.0.0.0 area 100
 network dd.0.0.0 0.255.255.255 area 100
!
router bgp 100
 bgp cluster-id 1
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor ee.ee.ee.ee remote-as 100
 neighbor ee.ee.ee.ee update-source Loopback0
 neighbor ww.ww.ww.ww remote-as 100
 neighbor ww.ww.ww.ww update-source Loopback0
 neighbor bb.bb.bb.bb remote-as 200
 neighbor bb.bb.bb.bb ebgp-multihop 255
 neighbor bb.bb.bb.bb update-source Loopback0
 no auto-summary
 !
address-family ipv4
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee route-reflector-client               !IPv4+labels session to PE1
 neighbor ee.ee.ee.ee send-label
 neighbor ww.ww.ww.ww activate
 neighbor ww.ww.ww.ww route-reflector-client               !IPv4+labels session to ASBR1
 neighbor ww.ww.ww.ww send-label
 no neighbor bb.bb.bb.bb activate
 no auto-summary
 no synchronization
 exit-address-family
 !
address-family vpnv4
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee route-reflector-client               !VPNv4 session with PE1
 neighbor ee.ee.ee.ee send-community extended
 neighbor bb.bb.bb.bb activate
 neighbor bb.bb.bb.bb next-hop-unchanged                   !MH-VPNv4 session with RR2
 neighbor bb.bb.bb.bb send-community extended                !with next hop unchanged

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

ASBR1 Configuration Example (MPLS VPN Service Provider)

ASBR1 exchanges IPv4 routes and MPLS labels with ASBR2.

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

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

```
ip subnet-zero
mpls label protocol ldp
!
interface Loopback0
 ip address ww.ww.ww.ww 255.255.255.255
!
interface Ethernet0/2
 ip address hh.0.0.2 255.0.0.0
!
interface Ethernet0/3
 ip address dd.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
!
router ospf 10
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Ethernet0/2
 network ww.ww.ww.ww 0.0.0.0 area 100
 network dd.0.0.0 0.255.255.255 area 100

router bgp 100
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor aa.aa.aa.aa remote-as 100
 neighbor aa.aa.aa.aa update-source Loopback0
 neighbor hh.0.0.1 remote-as 200
 no auto-summary
 !
!
address-family ipv4                      ! Redistributing IGP into BGP
 redistribute ospf 10                    ! so that PE1 & RR1 loopbacks
 neighbor aa.aa.aa.aa activate           ! get into the BGP table
 neighbor aa.aa.aa.aa send-label
 neighbor hh.0.0.1 activate
 neighbor hh.0.0.1 advertisement-interval 5
 neighbor hh.0.0.1 send-label
 neighbor hh.0.0.1 route-map IN in       ! accepting routes in route map IN.
 neighbor hh.0.0.1 route-map OUT out     ! distributing routes in route map OUT.
 neighbor kk.0.0.1 activate
 neighbor kk.0.0.1 advertisement-interval 5
 neighbor kk.0.0.1 send-label
 neighbor kk.0.0.1 route-map IN in       ! accepting routes in route map IN.
 neighbor kk.0.0.1 route-map OUT out     ! distributing routes in route map OUT.
 no auto-summary
 no synchronization
 exit-address-family
!
ip default-gateway 3.3.0.1
ip classless
!
access-list 1 permit ee.ee.ee.ee log              !Setting up the access lists
access-list 2 permit ff.ff.ff.ff log
```

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

Route Reflector 2 Configuration Example (MPLS VPN Service Provider)

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

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address bb.bb.bb.bb 255.255.255.255
!
interface Serial1/1
 ip address ii.0.0.2 255.0.0.0
!
router ospf 20
 log-adjacency-changes
 network bb.bb.bb.bb 0.0.0.0 area 200
 network ii.0.0.0 0.255.255.255 area 200
!
router bgp 200
 bgp cluster-id 1
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor aa.aa.aa.aa remote-as 100
 neighbor aa.aa.aa.aa ebgp-multihop 255
 neighbor aa.aa.aa.aa update-source Loopback0
 neighbor ff.ff.ff.ff remote-as 200
 neighbor ff.ff.ff.ff update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor aa.aa.aa.aa activate
 neighbor aa.aa.aa.aa next-hop-unchanged              !Multihop VPNv4 session with RR1
 neighbor aa.aa.aa.aa send-community extended              !with next-hop-unchanged
 neighbor ff.ff.ff.ff activate
 neighbor ff.ff.ff.ff route-reflector-client          !VPNv4 session with PE2
 neighbor ff.ff.ff.ff send-community extended
 exit-address-family
!
ip default-gateway 3.3.0.1
no ip classless
!
end
```

ASBR2 Configuration Example (MPLS VPN Service Provider)

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

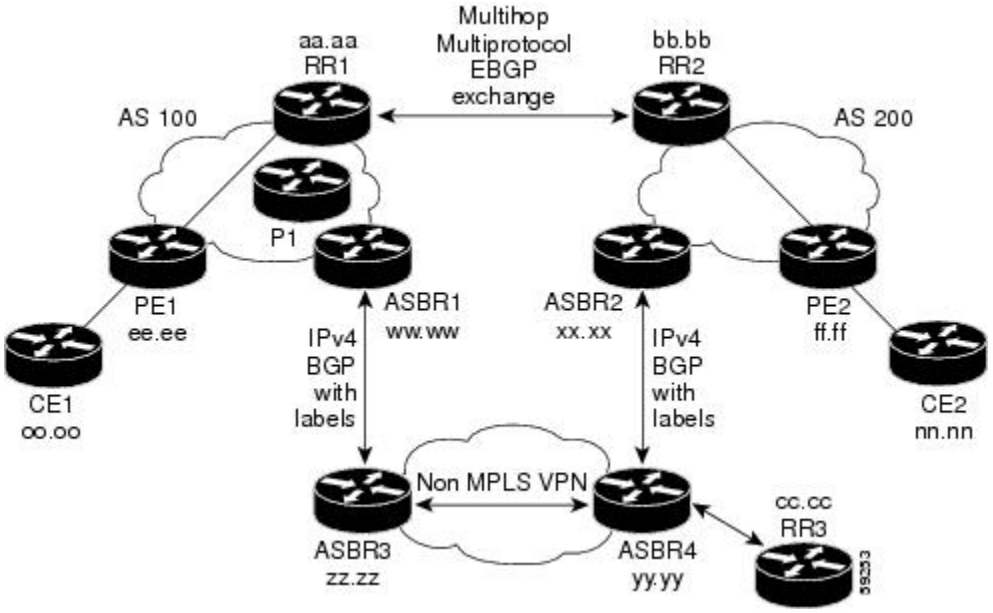
```
ip subnet-zero
ip cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address xx.xx.xx.xx 255.255.255.255
!
interface Ethernet1/0
 ip address hh.0.0.1 255.0.0.0
!
interface Ethernet1/2
 ip address jj.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
 !
router ospf 20
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 redistribute bgp 200 subnets           ! Redistributing the routes learned from
 passive-interface Ethernet1/0              ! ASBR1(eBGP+labels session) into IGP
 network xx.xx.xx.xx 0.0.0.0 area 200        ! so that PE2 will learn them
 network jj..0.0 0.255.255.255 area 200
 !
router bgp 200
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor bb.bb.bb.bb remote-as 200
 neighbor bb.bb.bb.bb update-source Loopback0
 neighbor hh.0.0.2 remote-as 100
 no auto-summary
 !
address-family ipv4
 redistribute ospf 20                         ! Redistributing IGP into BGP
 neighbor hh.0.0.2 activate                   ! so that PE2 & RR2 loopbacks
 neighbor hh.0.0.2 advertisement-interval 5   ! will get into the BGP-4 table.
 neighbor hh.0.0.2 route-map IN in
 neighbor hh.0.0.2 route-map OUT out
 neighbor hh.0.0.2 send-label
 neighbor kk.0.0.2 activate
 neighbor kk.0.0.2 advertisement-interval 5
 neighbor kk.0.0.2 route-map IN in
 neighbor kk.0.0.2 route-map OUT out
 neighbor kk.0.0.2 send-label
 no auto-summary
 no synchronization
 exit-address-family
!
address-family vpnv4
 neighbor bb.bb.bb.bb activate
 neighbor bb.bb.bb.bb send-community extended
 exit-address-family
 !
ip default-gateway 3.3.0.1
ip classless
!
```

```
access-list 1 permit ff.ff.ff.ff log          !Setting up the access lists
access-list 2 permit ee.ee.ee.ee log
access-list 3 permit bb.bb.bb.bb log
access-list 4 permit aa.aa.aa.aa log
route-map IN permit 11                       !Setting up the route maps
 match ip address 2
 match mpls-label
!
route-map IN permit 12
 match ip address 4
!
route-map OUT permit 10
 match ip address 1
 set mpls-label
!
route-map OUT permit 13
 match ip address 3
end
```

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

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

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



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

The configuration example for RR1 specifies the following:

- RR1 exchanges VPN-IPv4 routes with RR2 using multiprotocol, multihop eBGP.

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

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address aa.aa.aa.aa 255.255.255.255
!
interface Serial1/2
 ip address dd.0.0.2 255.0.0.0
 clockrate 124061
!
router ospf 10
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 network aa.aa.aa.aa 0.0.0.0 area 100
 network dd.0.0.0 0.255.255.255 area 100
!
router bgp 100
 bgp cluster-id 1
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor ee.ee.ee.ee remote-as 100
 neighbor ee.ee.ee.ee update-source Loopback0
 neighbor ww.ww.ww.ww remote-as 100
 neighbor ww.ww.ww.ww update-source Loopback0
 neighbor bb.bb.bb.bb remote-as 200
 neighbor bb.bb.bb.bb ebgp-multihop 255
 neighbor bb.bb.bb.bb update-source Loopback0
 no auto-summary
 !
address-family ipv4
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee route-reflector-client               !IPv4+labels session to PE1
 neighbor ee.ee.ee.ee send-label
 neighbor ww.ww.ww.ww activate
 neighbor ww.ww.ww.ww route-reflector-client               !IPv4+labels session to ASBR1
 neighbor ww.ww.ww.ww send-label
 no neighbor bb.bb.bb.bb activate
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee route-reflector-client              !VPNv4 session with PE1
 neighbor ee.ee.ee.ee send-community extended
 neighbor bb.bb.bb.bb activate
 neighbor bb.bb.bb.bb next-hop-unchanged                   !MH-VPNv4 session with RR2
 neighbor bb.bb.bb.bb send-community extended                with next-hop-unchanged
 exit-address-family
!
ip default-gateway 3.3.0.1
no ip classless
!
snmp-server engineID local 00000009020000D0584B25C0
snmp-server community public RO
```

```
snmp-server community write RW
no snmp-server ifindex persist
snmp-server packetsize 2048
!
end
```

ASBR1 Configuration Example (Non-MPLS VPN Service Provider)

ASBR1 exchanges IPv4 routes and MPLS labels with ASBR2.

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

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

```
ip subnet-zero
ip cef distributed
mpls label protocol ldp
!
interface Loopback0
 ip address ww.ww.ww.ww 255.255.255.255
!
interface Serial3/0/0
 ip address kk.0.0.2 255.0.0.0
 ip route-cache distributed
!
interface Ethernet0/3
 ip address dd.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
!
router ospf 10
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Serial3/0/0
 network ww.ww.ww.ww 0.0.0.0 area 100
 network dd.0.0.0 0.255.255.255 area 100

router bgp 100
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor aa.aa.aa.aa remote-as 100
 neighbor aa.aa.aa.aa update-source Loopback0
 neighbor kk.0.0.1 remote-as 200
 no auto-summary
!
 address-family ipv4
 redistribute ospf 10                      ! Redistributing IGP into BGP
 neighbor aa.aa.aa.aa activate             ! so that PE1 & RR1 loopbacks
 neighbor aa.aa.aa.aa send-label           ! get into BGP table
 neighbor kk.0.0.1 activate
 neighbor kk.0.0.1 advertisement-interval 5
 neighbor kk.0.0.1 send-label
 neighbor kk.0.0.1 route-map IN in    ! Accepting routes specified in route map IN
 neighbor kk.0.0.1 route-map OUT out  ! Distributing routes specified in route map OUT
 no auto-summary
 no synchronization
 exit-address-family
```

```
!
ip default-gateway 3.3.0.1
ip classless
!
access-list 1 permit ee.ee.ee.ee log
access-list 2 permit ff.ff.ff.ff log
access-list 3 permit aa.aa.aa.aa log
access-list 4 permit bb.bb.bb.bb log
!
route-map IN permit 10
 match ip address 2
 match mpls-label
!
route-map IN permit 11
 match ip address 4
!
route-map OUT permit 12
 match ip address 3
!
route-map OUT permit 13
 match ip address 1
 set mpls-label
!
end
```

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

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

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address bb.bb.bb.bb 255.255.255.255
!
interface Serial1/1
 ip address ii.0.0.2 255.0.0.0
!
router ospf 20
 log-adjacency-changes
 network bb.bb.bb.bb 0.0.0.0 area 200
 network ii.0.0.0 0.255.255.255 area 200
!
router bgp 200
 bgp cluster-id 1
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor aa.aa.aa.aa remote-as 100
 neighbor aa.aa.aa.aa ebgp-multihop 255
 neighbor aa.aa.aa.aa update-source Loopback0
 neighbor ff.ff.ff.ff remote-as 200
 neighbor ff.ff.ff.ff update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor aa.aa.aa.aa activate
 neighbor aa.aa.aa.aa next-hop-unchanged              !MH vpnv4 session with RR1
 neighbor aa.aa.aa.aa send-community extended              !with next-hop-unchanged
 neighbor ff.ff.ff.ff activate
 neighbor ff.ff.ff.ff route-reflector-client          !vpnv4 session with PE2
 neighbor ff.ff.ff.ff send-community extended
 exit-address-family
```

```
!
ip default-gateway 3.3.0.1
no ip classless
!
end
```

ASBR2 Configuration Example (Non-MPLS VPN Service Provider)

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

```
ip subnet-zero
ip cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address xx.xx.xx.xx 255.255.255.255
!
interface Ethernet0/1
 ip address qq.0.0.2 255.0.0.0
!
interface Ethernet1/2
 ip address jj.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
 !
router ospf 20
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 redistribute bgp 200 subnets            !redistributing the routes learned from
 passive-interface Ethernet0/1                !ASBR2 (eBGP+labels session) into IGP
 network xx.xx.xx.xx 0.0.0.0 area 200         !so that PE2 will learn them
 network jj.0.0.0 0.255.255.255 area 200
 !
router bgp 200
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor bb.bb.bb.bb remote-as 200
 neighbor bb.bb.bb.bb update-source Loopback0
 neighbor qq.0.0.1 remote-as 100
 no auto-summary
!
address-family ipv4                          ! Redistributing IGP into BGP
                                                                redistribute ospf 20
                    ! so that PE2 & RR2 loopbacks
 neighbor qq.0.0.1 activate                  !  will get into the BGP-4 table
 neighbor qq.0.0.1 advertisement-interval 5
 neighbor qq.0.0.1 route-map IN in
 neighbor qq.0.0.1 route-map OUT out
 neighbor qq.0.0.1 send-label
 no auto-summary
 no synchronization
 exit-address-family
 !
address-family vpnv4
 neighbor bb.bb.bb.bb activate
 neighbor bb.bb.bb.bb send-community extended
 exit-address-family
 !
```

```
ip default-gateway 3.3.0.1
ip classless
!
access-list 1 permit ff.ff.ff.ff log
access-list 2 permit ee.ee.ee.ee log
access-list 3 permit bb.bb.bb.bb log
access-list 4 permit aa.aa.aa.aa log
!
route-map IN permit 11
 match ip address 2
 match mpls-label
!
route-map IN permit 12
 match ip address 4
!
route-map OUT permit 10
 match ip address 1
 set mpls-label
!
route-map OUT permit 13
 match ip address 3
!
end
```

ASBR3 Configuration Example (Non-MPLS VPN Service Provider)

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



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

```
ip subnet-zero
ip cef
!
interface Loopback0
 ip address yy.yy.yy.yy 255.255.255.255
interface Hssi4/0
 ip address mm.0.0.0.1 255.0.0.0
 mpls ip
 hssi internal-clock
 !
interface Serial5/0
 ip address kk.0.0.1 255.0.0.0
 load-interval 30
 clockrate 124061
!
router ospf 30
log-adjacency-changes
auto-cost reference-bandwidth 1000
redistribute connected subnets
network yy.yy.yy.yy 0.0.0.0 area 300
network mm.0.0.0 0.255.255.255 area 300
!
router bgp 300
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor cc.cc.cc.cc remote-as 300
 neighbor cc.cc.cc.cc update-source Loopback0
```

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

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

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

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

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

ASBR4 Configuration Example (Non-MPLS VPN Service Provider)

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



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

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

```
 network mm.0.0.0 0.255.255.255 area 300
 !
router bgp 300
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor cc.cc.cc.cc remote-as 300
 neighbor cc.cc.cc.cc update-source Loopback0
 neighbor qq.0.0.2 remote-as 200
 no auto-summary
 !
 address-family ipv4
 neighbor cc.cc.cc.cc activate
 neighbor cc.cc.cc.cc send-label
 neighbor qq.0.0.2 activate
 neighbor qq.0.0.2 advertisement-interval 5
 neighbor qq.0.0.2 send-label
 neighbor qq.0.0.2 route-map IN in
 neighbor qq.0.0.2 route-map OUT out
 no auto-summary
 no synchronization
 exit-address-family
!
ip classless
!
access-list 1 permit ff.ff.ff.ff log
access-list 2 permit ee.ee.ee.ee log
access-list 3 permit bb.bb.bb.bb log
access-list 4 permit aa.aa.aa.aa log
!
route-map IN permit 10
 match ip address 1
  match mpls-label
!
route-map IN permit 11
   match ip address 3
!
route-map OUT permit 12
 match ip address 2
  set mpls-label
!
route-map OUT permit 13
   match ip address 4
!
ip default-gateway 3.3.0.1
ip classless
!
end
```

Additional References

Related Documents

Related Topic	Document Title
MPLS	MPLS Product Literature

Standards

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

MIBs

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

RFCs

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

Technical Assistance

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

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

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

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

Table 2: Feature Information for MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels

Feature Name	Releases	Feature Configuration Information
MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels	12.0(21)ST 12.0(22)S 12.0(23)S 12.2(13)T 12.0(24)S 12.2(14)S 12.0(27)S 12.0(29)S Cisco IOS XE Release 2.5	This module explains how to configure an MPLS VPN Inter-AS network so that the ASBRs exchange IPv4 routes with MPLS labels of the provider edge (PE) routers. Route reflectors (RRs) exchange VPN-IPv4 routes by using multihop, multiprotocol, external Border Gateway Protocol (eBGP). In Cisco IOS XE Release 2.5, this feature was implemented on the Cisco ASR 1000 Series Routers. This feature uses no new or modified commands.