MPLS: Layer 3 VPNs Configuration Guide, Cisco IOS Release 15M&T

First Published: 2012-11-21

Last Modified: 2013-03-15

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

CONTENTS

CHAPTER 6 **Configuring Route Maps to Control the Distribution of MPLS Labels Between Routers in an MPLS VPN** 109

CHAPTER 7 **Assigning an ID Number to an MPLS VPN** 121

CHAPTER 11

CHAPTER 12

CHAPTER 1

MPLS Virtual Private Networks

An MPLS Virtual Private Network (VPN) consists of a set of sites that are interconnected by means of a Multiprotocol Label Switching (MPLS) provider core network. At each customer site, one or more customer edge (CE) devices attach to one or more provider edge (PE) devices. This module explains how to create an MPLS VPN.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for MPLS Virtual Private Networks

- Make sure that you have installed Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP), and Cisco Express Forwarding in your network.

- All devices in the core, including the provider edge (PE) devices, must be able to support Cisco Express Forwarding and MPLS forwarding. See the “Assessing the Needs of the MPLS Virtual Private Network Customers” section.

- Cisco Express Forwarding must be enabled on all devices in the core, including the PE devices. For information about how to determine if Cisco Express Forwarding is enabled, see the “Configuring Basic Cisco Express Forwarding” module in the *Cisco Express Forwarding Configuration Guide*.

Restrictions for MPLS Virtual Private Networks

When static routes are configured in a Multiprotocol Label Switching (MPLS) or MPLS virtual private network (VPN) environment, some variations of the **ip route** and **ip route vrf** commands are not supported. These variations of the commands are not supported in software releases that support the Tag Forwarding Information Base (TFIB). The TFIB cannot resolve prefixes when the recursive route over which the prefixes travel disappears and then reappears. However, the command variations are supported in releases that support the MPLS Forwarding Infrastructure (MFI). For details about the supported releases, see the *Multiprotocol Label Switching Command Reference*. Use the following guidelines when configuring static routes.

Supported Static Routes in an MPLS Environment

The following **ip route** command is supported when you configure static routes in an MPLS environment:

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

The following **ip route** commands are supported when you configure static routes in an MPLS environment and configure load sharing with static nonrecursive routes and a specific outbound interface:

- **ip route** *destination-prefix mask* **interface1 next-hop1**
- **ip route** *destination-prefix mask* **interface2 next-hop2**

Unsupported Static Routes in an MPLS Environment That Uses the TFIB

The following **ip route** command is not supported when you configure static routes in an MPLS environment:

- **ip route** *destination-prefix mask next-hop-address*

The following **ip route** command is not supported when you configure static routes in an MPLS environment and enable load sharing where the next hop can be reached through two paths:

- **ip route** *destination-prefix mask next-hop-address*

The following **ip route** commands are not supported when you configure static routes in an MPLS environment and enable load sharing where the destination can be reached through two next hops:

- **ip route** *destination-prefix mask* **next-hop1**
- **ip route** *destination-prefix mask* **next-hop2**

Use the *interface* an *next-hop* arguments when specifying static routes.

Supported Static Routes in an MPLS VPN Environment

The following **ip route vrf** commands are supported when you configure static routes in an MPLS VPN environment, and the next hop and interface are in the same VRF:

- **ip route vrf** *vrf-name destination-prefix mask next-hop-address*
- **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address*
- **ip route vrf** *vrf-name destination-prefix mask* **interface1 next-hop1**
- **ip route vrf** *vrf-name destination-prefix mask* **interface2 next-hop2**

The following **ip route vrf** commands are supported when you configure static routes in an MPLS VPN environment, and the next hop is in the global table in the MPLS cloud in the global routing table. For example, these commands are supported when the next hop is pointing to the Internet gateway.

- **ip route vrf** *vrf-name destination-prefix mask next-hop-address* **global**
- **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address* (This command is supported when the next hop and interface are in the core.)

The following **ip route** commands are supported when you configure static routes in an MPLS VPN environment and enable load sharing with static nonrecursive routes and a specific outbound interface:

- **ip route** *destination-prefix mask* **interface1 next-hop1**
- **ip route** *destination-prefix mask* **interface2 next-hop2**

Unsupported Static Routes in an MPLS VPN Environment That Uses the TFIB

The following **ip route** command is not supported when you configure static routes in an MPLS VPN environment, the next hop is in the global table in the MPLS cloud within the core, and you enable load sharing where the next hop can be reached through two paths:

- **ip route vrf** *destination-prefix mask next-hop-address* **global**

The following **ip route** commands are not supported when you configure static routes in an MPLS VPN environment, the next hop is in the global table in the MPLS cloud within the core, and you enable load sharing where the destination can be reached through two next hops:

- **ip route vrf** *destination-prefix mask* **next-hop1 global**
- **ip route vrf** *destination-prefix mask* **next-hop2 global**

The following **ip route vrf** commands are not supported when you configure static routes in an MPLS VPN environment, and the next hop and interface are in the same VRF:

- **ip route vrf** *vrf-name destination-prefix mask* **next-hop1** *vrf-name destination-prefix mask* **next-hop1**
- **ip route vrf** *vrf-name destination-prefix mask* **next-hop2**

Supported Static Routes in an MPLS VPN Environment Where the Next Hop Resides in the Global Table on the CE Device

The following **ip route vrf** command is supported when you configure static routes in an MPLS VPN environment, and the next hop is in the global table on the customer edge (CE) side. For example, the following command is supported when the destination prefix is the CE device’s loopback address, as in external Border Gateway Protocol (EBGP) multihop cases.

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

The following **ip route** commands are supported when you configure static routes in an MPLS VPN environment, the next hop is in the global table on the CE side, and you enable load sharing with static nonrecursive routes and a specific outbound interface:

- **ip route** *destination-prefix mask* **interface1 nexthop1**
- **ip route** *destination-prefix mask* **interface2 nexthop2**

Information About MPLS Virtual Private Networks

MPLS Virtual Private Network Definition

Before defining a Multiprotocol Label Switching virtual private network (MPLS VPN), you must define a VPN in general. A VPN is:

- An IP-based network delivering private network services over a public infrastructure
- A set of sites that are allowed to communicate with each other privately over the Internet or other public or private networks

Conventional VPNs are created by configuring a full mesh of tunnels or permanent virtual circuits (PVCs) to all sites in a VPN. This type of VPN is not easy to maintain or expand, because adding a new site requires changing each edge device in the VPN.

MPLS-based VPNs are created in Layer 3 and are based on the peer model. The peer model enables the service provider and the customer to exchange Layer 3 routing information. The service provider relays the data between the customer sites without the customer’s involvement.

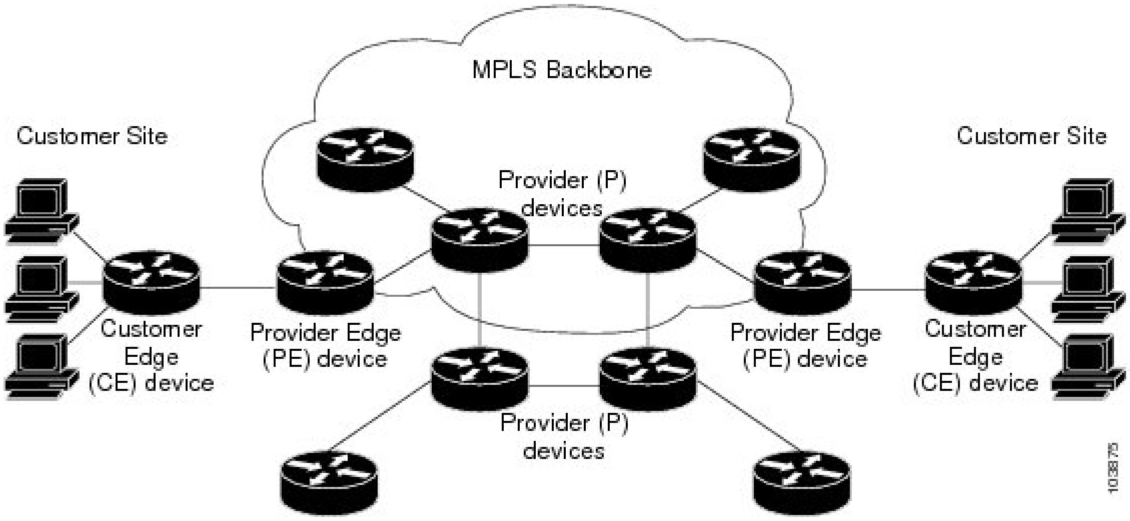
MPLS VPNs are easier to manage and expand than conventional VPNs. When a new site is added to an MPLS VPN, only the service provider’s edge device that provides services to the customer site needs to be updated.

The different parts of the MPLS VPN are described as follows:

- Provider (P) device—Device in the core of the provider network. P devices run MPLS switching, and do not attach VPN labels to routed packets. The MPLS label in each route is assigned by the provider edge (PE) device. VPN labels are used to direct data packets to the correct egress device.
- PE device—Device that attaches the VPN label to incoming packets based on the interface or subinterface on which they are received. A PE device attaches directly to a customer edge (CE) device.
- Customer (C) device—Device in the ISP or enterprise network.
- CE device—Edge device on the network of the ISP that connects to the PE device on the network. A CE device must interface with a PE device.

The figure below shows a basic MPLS VPN.

Figure 1: Basic MPLS VPN Terminology


MPLS Backbone
Customer Site
Customer Site
Provider (P) devices
Customer Edge (CE) device
Provider Edge (PE) device
Provider Edge (PE) device
Customer Edge (CE) device
Provider (P) devices
103875

How an MPLS Virtual Private Network Works

Multiprotocol Label Switching virtual private network (MPLS VPN) functionality is enabled at the edge of an MPLS network. The provider edge (PE) device performs the following:

- Exchanges routing updates with the customer edge (CE) device.
- Translates the CE routing information into VPNv4 routes.
- Exchanges VPNv4 routes with other PE devices through the Multiprotocol Border Gateway Protocol (MP-BGP).

The following sections describe how MPLS VPN works:

How Virtual Routing and Forwarding Tables Work in an MPLS Virtual Private Network

Each virtual private network (VPN) is associated with one or more virtual routing and forwarding (VRF) instances. A VRF defines the VPN membership of a customer site attached to a PE device. A VRF consists of the following components:

- An IP routing table
- A derived Cisco Express Forwarding table
- A set of interfaces that use the forwarding table
- A set of rules and routing protocol parameters that control the information that is included in the routing table

A one-to-one relationship does not necessarily exist between customer sites and VPNs. A site can be a member of multiple VPNs. However, a site can associate with only one VRF. A site’s VRF contains all the routes available to the site from the VPNs of which it is a member.

Packet forwarding information is stored in the IP routing table and the Cisco Express Forwarding table for each VRF. A separate set of routing and Cisco Express Forwarding tables is maintained for each VRF. These tables prevent information from being forwarded outside a VPN, and they also prevent packets that are outside a VPN from being forwarded to a device within the VPN.

How VPN Routing Information Is Distributed in an MPLS Virtual Private Network

The distribution of virtual private network (VPN) routing information is controlled through the use of VPN route target communities, implemented by Border Gateway Protocol (BGP) extended communities. VPN routing information is distributed as follows:

- When a VPN route that is learned from a customer edge (CE) device is injected into BGP, a list of VPN route target extended community attributes is associated with it. Typically the list of route target community extended values is set from an export list of route targets associated with the virtual routing and forwarding (VRF) instance from which the route was learned.
- An import list of route target extended communities is associated with each VRF. The import list defines route target extended community attributes that a route must have in order for the route to be imported into the VRF. For example, if the import list for a particular VRF includes route target extended communities A, B, and C, then any VPN route that carries any of those route target extended communities—A, B, *or* C—is imported into the VRF.

MPLS Forwarding

Based on routing information stored in the virtual routing and forwarding (VRF) IP routing table and VRF Cisco Express Forwarding table, packets are forwarded to their destination using Multiprotocol Label Switching (MPLS).

A provider edge (PE) device binds a label to each customer prefix learned from a customer edge (CE) device and includes the label in the network reachability information for the prefix that it advertises to other PE devices. When a PE device forwards a packet received from a CE device across the provider network, it labels the packet with the label learned from the destination PE device. When the destination PE device receives the labeled packet, it pops the label and uses it to direct the packet to the correct CE device. Label forwarding across the provider backbone is based on either dynamic label switching or traffic engineered paths. A customer data packet carries two levels of labels when traversing the backbone:

- The top label directs the packet to the correct PE device.
- The second label indicates how that PE device should forward the packet to the CE device.

Major Components of an MPLS Virtual Private Network

An Multiprotocol Label Switching (MPLS)-based virtual private network (VPN) has three major components:

- VPN route target communities—A VPN route target community is a list of all members of a VPN community. VPN route targets need to be configured for each VPN community member.
- Multiprotocol BGP (MP-BGP) peering of VPN community provider edge (PE) devices—MP-BGP propagates virtual routing and forwarding (VRF) reachability information to all members of a VPN community. MP-BGP peering must be configured on all PE devices within a VPN community.
- MPLS forwarding—MPLS transports all traffic between all VPN community members across a VPN service-provider network.

A one-to-one relationship does not necessarily exist between customer sites and VPNs. A given site can be a member of multiple VPNs. However, a site can associate with only one VRF. A customer-site VRF contains all the routes available to the site from the VPNs of which it is a member.

Benefits of an MPLS Virtual Private Network

Multiprotocol Label Switching virtual private networks (MPLS VPNs) allow service providers to deploy scalable VPNs and build the foundation to deliver value-added services, such as the following:

Connectionless Service

A significant technical advantage of MPLS VPNs is that they are connectionless. The Internet owes its success to its basic technology, TCP/IP. TCP/IP is built on a packet-based, connectionless network paradigm. This means that no prior action is necessary to establish communication between hosts, making it easy for two parties to communicate. To establish privacy in a connectionless IP environment, current VPN solutions impose a connection-oriented, point-to-point overlay on the network. Even if it runs over a connectionless network, a VPN cannot take advantage of the ease of connectivity and multiple services available in connectionless networks. When you create a connectionless VPN, you do not need tunnels and encryption for network privacy, thus eliminating significant complexity.

Centralized Service

Building VPNs in Layer 3 allows delivery of targeted services to a group of users represented by a VPN. A VPN must give service providers more than a mechanism for privately connecting users to intranet services. It must also provide a way to flexibly deliver value-added services to targeted customers. Scalability is critical, because customers want to use services privately in their intranets and extranets. Because MPLS VPNs are seen as private intranets, you may use new IP services such as:

- Multicast
- Quality of service (QoS)
- Telephony support within a VPN
- Centralized services including content and web hosting to a VPN

You can customize several combinations of specialized services for individual customers. For example, a service that combines IP multicast with a low-latency service class enables video conferencing within an intranet.

Scalability

If you create a VPN using connection-oriented, point-to-point overlays, Frame Relay, or ATM virtual connections (VCs), the VPN’s key deficiency is scalability. Specifically, connection-oriented VPNs without fully meshed connections between customer sites are not optimal. MPLS-based VPNs, instead, use the peer model and Layer 3 connectionless architecture to leverage a highly scalable VPN solution. The peer model requires a customer site to peer with only one provider edge (PE) device as opposed to all other customer edge (CE) devices that are members of the VPN. The connectionless architecture allows the creation of VPNs in Layer 3, eliminating the need for tunnels or VCs.

Other scalability issues of MPLS VPNs are due to the partitioning of VPN routes between PE devices and the further partitioning of VPN and Interior Gateway Protocol (IGP) routes between PE devices and provider (P) devices in a core network.

- PE devices must maintain VPN routes for those VPNs who are members.
- P devices do not maintain any VPN routes.

This increases the scalability of the provider's core and ensures that no one device is a scalability bottleneck.

Security

MPLS VPNs offer the same level of security as connection-oriented VPNs. Packets from one VPN do not inadvertently go to another VPN.

Security is provided in the following areas:

- At the edge of a provider network, ensuring packets received from a customer are placed on the correct VPN.
- At the backbone, VPN traffic is kept separate. Malicious spoofing (an attempt to gain access to a PE device) is nearly impossible because the packets received from customers are IP packets. These IP packets must be received on a particular interface or subinterface to be uniquely identified with a VPN label.

Ease of Creation

To take full advantage of VPNs, customers must be able to easily create new VPNs and user communities. Because MPLS VPNs are connectionless, no specific point-to-point connection maps or topologies are required. You can add sites to intranets and extranets and form closed user groups. Managing VPNs in this manner enables membership of any given site in multiple VPNs, maximizing flexibility in building intranets and extranets.

Flexible Addressing

To make a VPN service more accessible, customers of a service provider can design their own addressing plan, independent of addressing plans for other service provider customers. Many customers use private address spaces, as defined in RFC 1918, and do not want to invest the time and expense of converting to public IP addresses to enable intranet connectivity. MPLS VPNs allow customers to continue to use their present address spaces without network address translation (NAT) by providing a public and private view of the address. A NAT is required only if two VPNs with overlapping address spaces want to communicate. This enables customers to use their own unregistered private addresses, and communicate freely across a public IP network.

Integrated QoS Support

QoS is an important requirement for many IP VPN customers. It provides the ability to address two fundamental VPN requirements:

- Predictable performance and policy implementation
- Support for multiple levels of service in an MPLS VPN

Network traffic is classified and labeled at the edge of the network before traffic is aggregated according to policies defined by subscribers and implemented by the provider and transported across the provider core. Traffic at the edge and core of the network can then be differentiated into different classes by drop probability or delay.

Straightforward Migration

For service providers to quickly deploy VPN services, use a straightforward migration path. MPLS VPNs are unique because you can build them over multiple network architectures, including IP, ATM, Frame Relay, and hybrid networks.

Migration for the end customer is simplified because there is no requirement to support MPLS on the CE device and no modifications are required to a customer's intranet.

How to Configure MPLS Virtual Private Networks

Configuring the Core Network

Assessing the Needs of MPLS Virtual Private Network Customers

Before you configure a Multiprotocol Label Switching virtual private network (MPLS VPN), you need to identify the core network topology so that it can best serve MPLS VPN customers. Perform this task to identify the core network topology.

SUMMARY STEPS

1. Identify the size of the network.
2. Identify the routing protocols in the core.
3. Determine if you need MPLS VPN High Availability support.
4. Determine if you need Border Gateway Protocol (BGP) load sharing and redundant paths in the MPLS VPN core.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Identify the size of the network.	Identify the following to determine the number of devices and ports that you need: • How many customers do you need to support? • How many VPNs are needed per customer? • How many virtual routing and forwarding instances are there for each VPN?
Step 2	Identify the routing protocols in the core.	Determine which routing protocols you need in the core network.
Step 3	Determine if you need MPLS VPN High Availability support.	MPLS VPN Nonstop Forwarding and Graceful Restart are supported on select devices and Cisco software releases. Contact Cisco Support for the exact requirements and hardware support.

	Command or Action	Purpose
Step 4	Determine if you need Border Gateway Protocol (BGP) load sharing and redundant paths in the MPLS VPN core.	For configuration steps, see the “Load Sharing MPLS VPN Traffic” feature module in the *MPLS Layer 3 VPNs Inter-AS and CSC Configuration Guide*.

Configuring MPLS in the Core

To enable Multiprotocol Label Switching (MPLS) on all devices in the core, you must configure either of the following as a label distribution protocol:

- MPLS Label Distribution Protocol (LDP). For configuration information, see the “MPLS Label Distribution Protocol (LDP)” module in the *MPLS Label Distribution Protocol Configuration Guide*.
- MPLS Traffic Engineering Resource Reservation Protocol (RSVP). For configuration information, see the “MPLS Traffic Engineering and Enhancements” module in the *MPLS Traffic Engineering Path Calculation and Setup Configuration Guide*.

Connecting the MPLS Virtual Private Network Customers

Defining VRFs on the PE Devices to Enable Customer Connectivity

Use this procedure to define a virtual routing and forwarding (VRF) configuration for IPv4. To define a VRF for IPv4 and IPv6, see the “Configuring a Virtual Routing and Forwarding Instance for IPv6" section in the “IPv6 VPN over MPLS" module in the *MPLS Layer 3 VPNs Configuration Guide*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
6. **exit**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf vpn1`	Defines the virtual private network (VPN) routing instance by assigning a virtual routing and forwarding (VRF) name and enters VRF configuration mode. • The *vrf-name* argument is the name assigned to a VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:1`	Creates routing and forwarding tables. • The *route-distinguisher* argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter a route distinguisher (RD) in either of these formats: • 16-bit AS number:your 32-bit number, for example, 101:3 • 32-bit IP address:your 16-bit number, for example, 10.0.0.1:1
Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Device(config-vrf)# route-target both 100:1`	Creates a route-target extended community for a VRF. • The **import** keyword imports routing information from the target VPN extended community. • The **export** keyword exports routing information to the target VPN extended community. • The **both** keyword imports routing information from and exports routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both route-target extended communities.
Step 6	**exit** **Example:** `Device(config-vrf)# exit`	(Optional) Exits to global configuration mode.

Configuring VRF Interfaces on PE Devices for Each VPN Customer

To associate a virtual routing and forwarding (VRF) instance with an interface or subinterface on the provider edge (PE) devices, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*

4. **ip vrf forwarding** *vrf-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface`	Specifies the interface to configure and enters interface configuration mode. • The *type* argument specifies the type of interface to be configured. • The *number* argument specifies the port, connector, or interface card number.
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding vpn1`	Associates a VRF with the specified interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		(Optional) Exits to privileged EXEC mode.

Configuring Routing Protocols Between the PE and CE Devices

Configure the provider edge (PE) device with the same routing protocol that the customer edge (CE) device uses. You can configure the Border Gateway Protocol (BGP), Routing Information Protocol version 2 (RIPv2), or static routes between the PE and CE devices.

Configuring RIPv2 as the Routing Protocol Between the PE and CE Devices

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router rip**
4. **version** {**1** | **2**}

5. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
6. **network** *ip-address*
7. **redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**]
8. **exit-address-family**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router rip** **Example:** `Device(config)# router rip`	Enables the Routing Information Protocol (RIP).

Step 4	**version** {**1** | **2**}
	Example:
	`Device(config-router)# version 2`
		Specifies RIP version used globally by the device.

Step 5	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 vrf vpn1`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specifies the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**network** *ip-address* **Example:** `Device(config-router-af)# network 192.168.7.0`	Enables RIP on the PE-to-CE link.

	Command or Action	Purpose
Step 7	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**] **Example:** `Device(config-router-af)# redistribute bgp 200`	Redistributes routes from one routing domain into another routing domain. • For the RIPv2 routing protocol, use the **redistribute bgp** *as-number* command.
Step 8	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode.

Step 9	**end**
	Example:
	`Device(config-router)# end`
		(Optional) Exits to privileged EXEC mode.

Configuring Static Routes Between the PE and CE Devices

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route vrf** *vrf-name*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**]
6. **redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**]
7. **exit-address-family**
8. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ip route vrf** *vrf-name* **Example:** `Device(config)# ip route vrf vpn1`	Defines static route parameters for every provider edge-to-customer edge (PE-to-CE) session and enters router configuration mode.
Step 4	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 vrf vpn1`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**] **Example:** `Device(config-router-af)# redistribute static`	Redistributes routes from one routing domain into another routing domain. • To redistribute virtual routing and forwarding (VRF) static routes into the VRF Border Gateway Protocol (BGP) table, use the **redistribute static** command. See the command reference page for information about other arguments and keywords.
Step 6	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**] **Example:** `Device(config-router-af)# redistribute connected`	Redistributes routes from one routing domain into another routing domain. • To redistribute directly connected networks into the VRF BGP table, use the **redistribute connected** command.
Step 7	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode.

Step 8	**end**
	Example:
	`Device(config-router)# end`
		(Optional) Exits to privileged EXEC mode.

Verifying the Virtual Private Network Configuration

A route distinguisher must be configured for the virtual routing and forwarding (VRF) instance, and Multiprotocol Label Switching (MPLS) must be configured on the interfaces that carry the VRF. Use the **show ip vrf** command to verify the route distinguisher (RD) and interface that are configured for the VRF.

SUMMARY STEPS

1. **show ip vrf**

DETAILED STEPS

show ip vrf

Displays the set of defined VRF instances and associated interfaces. The output also maps the VRF instances to the configured route distinguisher.

Verifying Connectivity Between MPLS Virtual Private Network Sites

To verify that the local and remote customer edge (CE) devices can communicate across the Multiprotocol Label Switching (MPLS) core, perform the following tasks:

Verifying IP Connectivity from CE Device to CE Device Across the MPLS Core

SUMMARY STEPS

1. **enable**
2. **ping** [*protocol*] {*host-name* | *system-address*}
3. **trace** [*protocol*] [*destination*]
4. **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | *protocol* [*process-id*]] | [**list** [*access-list-name* | *access-list-number*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

Step 2 **ping** [*protocol*] {*host-name* | *system-address*}

Diagnoses basic network connectivity on AppleTalk, Connectionless-mode Network Service (CLNS), IP, Novell, Apollo, Virtual Integrated Network Service (VINES), DECnet, or Xerox Network Service (XNS) networks. Use the **ping** command to verify the connectivity from one CE device to another.

Step 3 **trace** [*protocol*] [*destination*]

Discovers the routes that packets take when traveling to their destination. The **trace** command can help isolate a trouble spot if two devices cannot communicate.

Step 4 **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | *protocol* [*process-id*]] | [**list** [*access-list-name* | *access-list-number*]

Displays the current state of the routing table. Use the *ip-address* argument to verify that CE1 has a route to CE2. Verify the routes learned by CE1. Make sure that the route for CE2 is listed.

Verifying That the Local and Remote CE Devices Are in the PE Routing Table

SUMMARY STEPS

1. **enable**
2. **show ip route vrf** *vrf-name* [*prefix*]
3. **show ip cef vrf** *vrf-name* [*ip-prefix*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

Step 2 **show ip route vrf** *vrf-name* [*prefix*]

Displays the IP routing table associated with a virtual routing and forwarding (VRF) instance. Check that the loopback addresses of the local and remote customer edge (CE) devices are in the routing table of the provider edge (PE) devices.

Step 3 **show ip cef vrf** *vrf-name* [*ip-prefix*]

Displays the Cisco Express Forwarding forwarding table associated with a VRF. Check that the prefix of the remote CE device is in the Cisco Express Forwarding table.

Configuration Examples for MPLS Virtual Private Networks

Example: Configuring an MPLS Virtual Private Network Using RIP

PE Configuration	CE Configuration
	ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.9 255.255.255.255 ! interface ip address 192.0.2.1 255.255.255.0 no cdp enable router rip version 2 timers basic 30 60 60 120 redistribute connected network 10.0.0.0 network 192.0.2.0 no auto-summary

PE Configuration	CE Configuration
ip vrf vpn1 rd 100:1 route-target export 100:1 route-target import 100:1 ! ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.1 255.255.255.255 ! interface ip vrf forwarding vpn1 ip address 192.0.2.3 255.255.255.0 no cdp enable interface ip address 192.0.2.2 255.255.255.0 mpls label protocol ldp mpls ip ! router rip version 2 timers basic 30 60 60 120 ! address-family ipv4 vrf vpn1 version 2 redistribute bgp 100 metric transparent network 192.0.2.0 distribute-list 20 in no auto-summary exit-address-family ! router bgp 100 no synchronization bgp log-neighbor changes neighbor 10.0.0.3 remote-as 100 neighbor 10.0.0.3 update-source Loopback0 no auto-summary ! address-family vpnv4 neighbor 10.0.0.3 activate neighbor 10.0.0.3 send-community extended bgp scan-time import 5 exit-address-family ! address-family ipv4 vrf vpn1 redistribute connected redistribute rip no auto-summary no synchronization exit-address-family	

Example: Configuring an MPLS Virtual Private Network Using Static Routes

PE Configuration	CE Configuration
ip vrf vpn1 rd 100:1 route-target export 100:1 route-target import 100:1 ! ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.1 255.255.255.255 ! interface ip vrf forwarding vpn1 ip address 192.0.2.3 255.255.255.0 no cdp enable ! interface ip address 192.168.0.1 255.255.0.0 mpls label protocol ldp mpls ip ! router ospf 100 network 10.0.0. 0.0.0.0 area 100 network 192.168.0.0 255.255.0.0 area 100 ! router bgp 100 no synchronization bgp log-neighbor changes neighbor 10.0.0.3 remote-as 100 neighbor 10.0.0.3 update-source Loopback0 no auto-summary ! address-family vpnv4 neighbor 10.0.0.3 activate neighbor 10.0.0.3 send-community extended bgp scan-time import 5 exit-address-family ! address-family ipv4 vrf vpn1 redistribute connected redistribute static no auto-summary no synchronization exit-address-family ! ip route vrf vpn1 10.0.0.9 255.255.255.255 192.0.2.2 ip route vrf vpn1 192.0.2.0 255.255.0.0 192.0.2.2	ip cef ! interface Loopback0 ip address 10.0.0.9 255.255.255.255 ! interface ip address 192.0.2.2 255.255.0.0 no cdp enable ! ip route 10.0.0.9 255.255.255.255 192.0.2.3 3 ip route 198.51.100.0 255.255.255.0 192.0.2.3 3

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Description of commands associated with MPLS and MPLS applications	Cisco IOS Multiprotocol Label Switching Command Reference
Configuring Cisco Express Forwarding	“Configuring Basic Cisco Express Forwarding” module in the *Cisco Express Forwarding Configuration Guide*
Border Gateway Protocol (BGP) load sharing	“Load Sharing MPLS VPN Traffic” module in the *MPLS Layer 3 VPNs Inter-AS and CSC Configuration Guide*
Configuring LDP	“MPLS Label Distribution Protocol (LDP)” module in the *MPLS Label Distribution Protocol Configuration Guide*
Configuring MPLS Traffic Engineering Resource Reservation Protocol (RSVP)	“"MPLS Traffic Engineering and Enhancements” module in the *MPLS Traffic Engineering Path Calculation and Setup Configuration Guide*
IPv6 VPN over MPLS	“IPv6 VPN over MPLS” module in the *MPLS Layer 3 VPNs Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS Virtual Private Networks

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 1: Feature Information for MPLS Virtual Private Networks

Feature Name	Releases	Feature Information
MPLS Virtual Private Networks	12.0(5)T 12.1(5)T 12.2(8)T 12.3(2)T	The MPLS Virtual Private Networks feature allows a set of sites that to be interconnected by means of a Multiprotocol Label Switching (MPLS) provider core network. At each customer site, one or more customer edge (CE) devices attach to one or more provider edge (PE) devices.

CHAPTER 2

Multiprotocol BGP MPLS VPN

A Multiprotocol Label Switching (MPLS) virtual private network (VPN) consists of a set of sites that are interconnected by means of an MPLS provider core network. At each site, there are one or more customer edge (CE) devices, which attach to one or more provider edge (PE) devices. PEs use the Multiprotocol-Border Gateway Protocol (MP-BGP) to dynamically communicate with each other.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for Multiprotocol BGP MPLS VPN

Configure MPLS virtual private networks (VPNs) in the core.

Information About Multiprotocol BGP MPLS VPN

MPLS Virtual Private Network Definition

Before defining a Multiprotocol Label Switching virtual private network (MPLS VPN), you must define a VPN in general. A VPN is:

- An IP-based network delivering private network services over a public infrastructure
- A set of sites that are allowed to communicate with each other privately over the Internet or other public or private networks

Conventional VPNs are created by configuring a full mesh of tunnels or permanent virtual circuits (PVCs) to all sites in a VPN. This type of VPN is not easy to maintain or expand, because adding a new site requires changing each edge device in the VPN.

MPLS-based VPNs are created in Layer 3 and are based on the peer model. The peer model enables the service provider and the customer to exchange Layer 3 routing information. The service provider relays the data between the customer sites without the customer’s involvement.

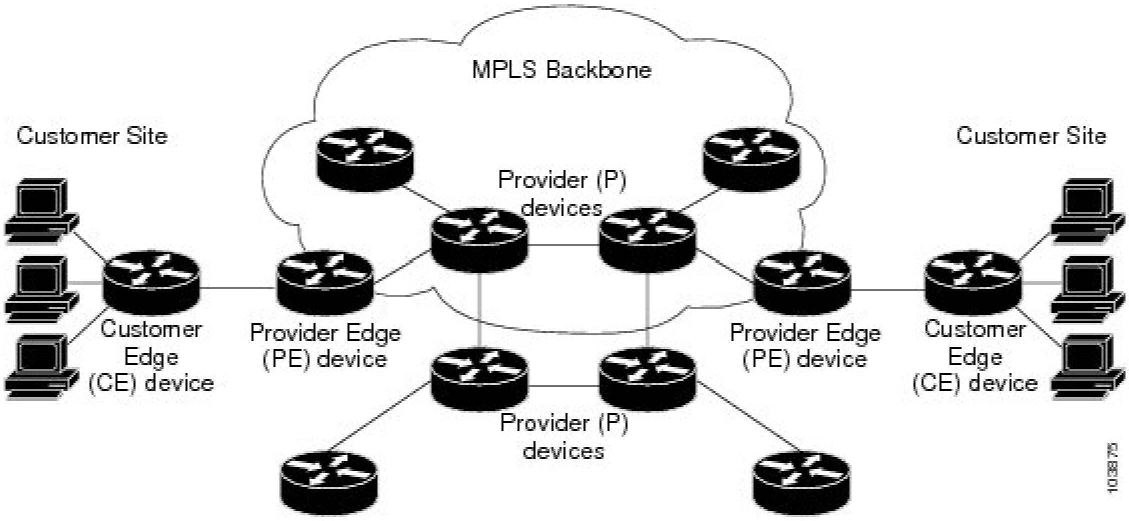
MPLS VPNs are easier to manage and expand than conventional VPNs. When a new site is added to an MPLS VPN, only the service provider’s edge device that provides services to the customer site needs to be updated.

The different parts of the MPLS VPN are described as follows:

- Provider (P) device—Device in the core of the provider network. P devices run MPLS switching, and do not attach VPN labels to routed packets. The MPLS label in each route is assigned by the provider edge (PE) device. VPN labels are used to direct data packets to the correct egress device.
- PE device—Device that attaches the VPN label to incoming packets based on the interface or subinterface on which they are received. A PE device attaches directly to a customer edge (CE) device.
- Customer (C) device—Device in the ISP or enterprise network.
- CE device—Edge device on the network of the ISP that connects to the PE device on the network. A CE device must interface with a PE device.

The figure below shows a basic MPLS VPN.

Figure 2: Basic MPLS VPN Terminology


MPLS Backbone
Customer Site
Customer Site
Provider (P) devices
Customer Edge (CE) device
Provider Edge (PE) device
Provider Edge (PE) device
Customer Edge (CE) device
Provider (P) devices
103875

How an MPLS Virtual Private Network Works

Multiprotocol Label Switching virtual private network (MPLS VPN) functionality is enabled at the edge of an MPLS network. The provider edge (PE) device performs the following:

- Exchanges routing updates with the customer edge (CE) device.
- Translates the CE routing information into VPNv4 routes.
- Exchanges VPNv4 routes with other PE devices through the Multiprotocol Border Gateway Protocol (MP-BGP).

The following sections describe how MPLS VPN works:

How Virtual Routing and Forwarding Tables Work in an MPLS Virtual Private Network

Each virtual private network (VPN) is associated with one or more virtual routing and forwarding (VRF) instances. A VRF defines the VPN membership of a customer site attached to a PE device. A VRF consists of the following components:

- An IP routing table
- A derived Cisco Express Forwarding table
- A set of interfaces that use the forwarding table
- A set of rules and routing protocol parameters that control the information that is included in the routing table

A one-to-one relationship does not necessarily exist between customer sites and VPNs. A site can be a member of multiple VPNs. However, a site can associate with only one VRF. A site’s VRF contains all the routes available to the site from the VPNs of which it is a member.

Packet forwarding information is stored in the IP routing table and the Cisco Express Forwarding table for each VRF. A separate set of routing and Cisco Express Forwarding tables is maintained for each VRF. These tables prevent information from being forwarded outside a VPN, and they also prevent packets that are outside a VPN from being forwarded to a device within the VPN.

How VPN Routing Information Is Distributed in an MPLS Virtual Private Network

The distribution of virtual private network (VPN) routing information is controlled through the use of VPN route target communities, implemented by Border Gateway Protocol (BGP) extended communities. VPN routing information is distributed as follows:

- When a VPN route that is learned from a customer edge (CE) device is injected into BGP, a list of VPN route target extended community attributes is associated with it. Typically the list of route target community extended values is set from an export list of route targets associated with the virtual routing and forwarding (VRF) instance from which the route was learned.
- An import list of route target extended communities is associated with each VRF. The import list defines route target extended community attributes that a route must have in order for the route to be imported into the VRF. For example, if the import list for a particular VRF includes route target extended communities A, B, and C, then any VPN route that carries any of those route target extended communities—A, B, *or* C—is imported into the VRF.

MPLS Forwarding

Based on routing information stored in the virtual routing and forwarding (VRF) IP routing table and VRF Cisco Express Forwarding table, packets are forwarded to their destination using Multiprotocol Label Switching (MPLS).

A provider edge (PE) device binds a label to each customer prefix learned from a customer edge (CE) device and includes the label in the network reachability information for the prefix that it advertises to other PE devices. When a PE device forwards a packet received from a CE device across the provider network, it labels the packet with the label learned from the destination PE device. When the destination PE device receives the labeled packet, it pops the label and uses it to direct the packet to the correct CE device. Label forwarding across the provider backbone is based on either dynamic label switching or traffic engineered paths. A customer data packet carries two levels of labels when traversing the backbone:

- The top label directs the packet to the correct PE device.
- The second label indicates how that PE device should forward the packet to the CE device.

BGP Distribution of VPN Routing Information

A provider edge (PE) device can learn an IP prefix from the following sources:

- A customer edge (CE) device by static configuration
- A Border Gateway Protocol (BGP) session with the CE device
- A Routing Information Protocol (RIP) exchange with the CE device

The IP prefix is a member of the IPv4 address family. After the PE device learns the IP prefix, the PE converts it into a VPN-IPv4 prefix by combining it with an 8-byte route distinguisher (RD). The generated prefix is a member of the VPN-IPv4 address family. It uniquely identifies the customer address, even if the customer site is using globally nonunique (unregistered private) IP addresses. The route distinguisher used to generate the VPN-IPv4 prefix is specified by a configuration command associated with the virtual routing and forwarding (VRF) instance on the PE device.

BGP distributes reachability information for VPN-IPv4 prefixes for each VPN. BGP communication occurs at two levels:

- Within an IP domains, known as an autonomous system (interior BGP [IBGP])
- Between autonomous systems (external BGP [EBGP])

PE-PE or PE-RR (route reflector) sessions are IBGP sessions, and PE-CE sessions are EBGP sessions. In an Enhanced Interior Gateway Routing Protocol (EIGRP) PE-CE environment, when an EIGRP internal route is redistributed into BGP by one PE, and then back into EIGRP by another PE, the originating router ID for the route is set to the router ID of the second PE, replacing the original internal router ID.

BGP propagates reachability information for VPN-IPv4 prefixes among PE devices by means of the BGP multiprotocol extensions (refer to RFC 2283, *Multiprotocol Extensions for BGP-4*), which define support for address families other than IPv4. Using the extensions ensures that the routes for a given VPN are learned only by other members of that VPN, enabling members of the VPN to communicate with each other.

Major Components of an MPLS Virtual Private Network

An Multiprotocol Label Switching (MPLS)-based virtual private network (VPN) has three major components:

- VPN route target communities—A VPN route target community is a list of all members of a VPN community. VPN route targets need to be configured for each VPN community member.
- Multiprotocol BGP (MP-BGP) peering of VPN community provider edge (PE) devices—MP-BGP propagates virtual routing and forwarding (VRF) reachability information to all members of a VPN community. MP-BGP peering must be configured on all PE devices within a VPN community.
- MPLS forwarding—MPLS transports all traffic between all VPN community members across a VPN service-provider network.

A one-to-one relationship does not necessarily exist between customer sites and VPNs. A given site can be a member of multiple VPNs. However, a site can associate with only one VRF. A customer-site VRF contains all the routes available to the site from the VPNs of which it is a member.

How to Configure Multiprotocol BGP MPLS VPN

Configuring Multiprotocol BGP Connectivity on the PE Devices and Route Reflectors

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **address-family vpnv4** [**unicast**]

8. **neighbor** {*ip-address* | *peer-group-name*} **send-community extended**
9. **neighbor** {*ip-address* | *peer-group-name*} **activate**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 100`	Configures a Border Gateway Protocol (BGP) routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP devices and tags the routing information passed along. The range is 0 to 65535. Private autonomous system numbers that can be used in internal networks are 64512 to 65535.
Step 4	**no bgp default ipv4-unicast** **Example:** `Device(config-router)# no bgp default ipv4-unicast`	(Optional) Disables the IPv4 unicast address family on all neighbors. • Use the **no bgp default ipv4-unicast** command if you are using this neighbor for Multiprotocol Label Switching (MPLS) routes only.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 10.0.0.1 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a neighboring BGP device. • The *ip-address* argument specifies the IP address of the neighbor.

	Command or Action	Purpose
		• The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**address-family vpnv4** [**unicast**] **Example:** `Device(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes. • The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **send-community extended** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a neighboring BGP device. • 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 10	**end**
	Example:
	`Device(config-router-af)# end`
		(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

You can enter a **show ip bgp neighbor** command to verify that the neighbors are up and running. If this command is not successful, enter a **debug ip bgp** *ip-address* **events** command, where *ip-address* is the IP address of the neighbor.

Configuring BGP as the Routing Protocol Between the PE and CE Devices

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*

6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **exit-address-family**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 100`	Configures a Border Gateway Protocol (BGP) routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP devices and tags the routing information passed along. The range is 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:** `Device(config-router)# address-family ipv4 vrf vpn1`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router-af)# neighbor 10.0.0.1 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.

	Command or Action	Purpose
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a neighboring BGP device. • 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	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode.

Step 8	**end**
	Example:
	`Device(config-router)# end`
		(Optional) Exits to privileged EXEC mode.

Verifying the Virtual Private Network Configuration

A route distinguisher must be configured for the virtual routing and forwarding (VRF) instance, and Multiprotocol Label Switching (MPLS) must be configured on the interfaces that carry the VRF. Use the **show ip vrf** command to verify the route distinguisher (RD) and interface that are configured for the VRF.

SUMMARY STEPS

1. **show ip vrf**

DETAILED STEPS

show ip vrf

Displays the set of defined VRF instances and associated interfaces. The output also maps the VRF instances to the configured route distinguisher.

Verifying Connectivity Between MPLS Virtual Private Network Sites

To verify that the local and remote customer edge (CE) devices can communicate across the Multiprotocol Label Switching (MPLS) core, perform the following tasks:

Verifying IP Connectivity from CE Device to CE Device Across the MPLS Core

SUMMARY STEPS

1. **enable**
2. **ping** [*protocol*] {*host-name* | *system-address*}
3. **trace** [*protocol*] [*destination*]
4. **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | *protocol* [*process-id*]] | [**list** [*access-list-name* | *access-list-number*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

Step 2 **ping** [*protocol*] {*host-name* | *system-address*}

Diagnoses basic network connectivity on AppleTalk, Connectionless-mode Network Service (CLNS), IP, Novell, Apollo, Virtual Integrated Network Service (VINES), DECnet, or Xerox Network Service (XNS) networks. Use the **ping** command to verify the connectivity from one CE device to another.

Step 3 **trace** [*protocol*] [*destination*]

Discovers the routes that packets take when traveling to their destination. The **trace** command can help isolate a trouble spot if two devices cannot communicate.

Step 4 **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | *protocol* [*process-id*]] | [**list** [*access-list-name* | *access-list-number*]

Displays the current state of the routing table. Use the *ip-address* argument to verify that CE1 has a route to CE2. Verify the routes learned by CE1. Make sure that the route for CE2 is listed.

Verifying That the Local and Remote CE Devices Are in the PE Routing Table

SUMMARY STEPS

1. **enable**
2. **show ip route vrf** *vrf-name* [*prefix*]
3. **show ip cef vrf** *vrf-name* [*ip-prefix*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

Step 2 **show ip route vrf** *vrf-name* [*prefix*]

Displays the IP routing table associated with a virtual routing and forwarding (VRF) instance. Check that the loopback addresses of the local and remote customer edge (CE) devices are in the routing table of the provider edge (PE) devices.

Step 3 **show ip cef vrf** *vrf-name* [*ip-prefix*]

Displays the Cisco Express Forwarding forwarding table associated with a VRF. Check that the prefix of the remote CE device is in the Cisco Express Forwarding table.

Configuration Examples for Multiprotocol BGP MPLS VPN

Example: Configuring an MPLS Virtual Private Network Using BGP

PE Configuration

```
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
ip cef
mpls ldp router-id Loopback0 force
mpls label protocol ldp
!
interface Loopback0
 ip address 10.0.0.1 255.255.255.255
!
interface
 ip vrf forwarding vpn1
 ip address 192.0.2.3 255.255.255.0
 no cdp enable
!
interface
ip address 192.0.2.2 255.255.255.0
mpls label protocol ldp
mpls ip
!
router ospf 100
network 10.0.0. 0.0.0.0 area 100
network 192.0.2.1 255.255.255.0 area 100
!
router bgp 100
 no synchronization
 bgp log-neighbor changes
 neighbor 10.0.0.3 remote-as 100
 neighbor 10.0.0.3 update-source Loopback0
no auto-summary
  !
address-family vpnv4
 neighbor 10.0.0.3 activate
 neighbor 10.0.0.3 send-community extended
 bgp scan-time import 5
 exit-address-family
  !
address-family ipv4 vrf vpn1
 redistribute connected
 neighbor 198.51.100.1 remote-as 200
 neighbor 198.51.100.1 activate
 neighbor 198.51.100.1 as-override
 neighbor 198.51.100.1 advertisement-interval
5
 no auto-summary
 no synchronization
 exit-address-family
```

CE Configuration

```
ip cef
mpls ldp router-id Loopback0 force
mpls label protocol ldp
!
interface Loopback0
 ip address 10.0.0.9 255.255.255.255
!
interface
 ip address 198.51.100.1 255.255.255.0
 no cdp enable
!
router bgp 200
 bgp log-neighbor-changes
 neighbor 198.51.100.2 remote-as 100
  !
address-family ipv4
 redistribute connected
 neighbor 198.51.100.2 activate
 neighbor 198.51.100.2
advertisement-interval 5
 no auto-summary
 no synchronization
 exit-address-family
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Description of commands associated with MPLS and MPLS applications	Cisco IOS Multiprotocol Label Switching Command Reference
Configuring MPLS virtual private networks	“MPLS Virtual Private Networks” module in the *MPLS Layer 3 VPNs Configuration Guide*

Standards and RFCs

RFC	Title
RFC 2283	*Multiprotocol Extensions for BGP-4*
RFC 2547	*BGP/MPLS VPNs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Multiprotocol BGP MPLS VPN

For information on compatibility of this feature with route processors (RP), see Cisco ASR 900 Series Aggregation Services Routers Feature Compatibility Matrix.

Table 2: Feature Information for Multiprotocol BGP MPLS VPN

Feature Name	Releases	Feature Information
Multiprotocol BGP MPLS VPN	12.0(11)ST 12.2(9)S 12.2(17b)SXA 12.2(27)SBB 12.3(8)T 15.2(1)S Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.5S	An MPLS VPN consists of a set of sites that are interconnected through the MPLS provider core network. At each site, there are one or more CEs, which attach to one or more PEs. The Multiprotocol BGP MPLS VPN feature allows PEs to use the MP-BGP to dynamically communicate with each other. In Cisco IOS Release 12.0(11)ST, this feature was introduced. In Cisco IOS Release 12.2(9)S, 12.2(17b)SXA, 12.2(27)SBB, 12.3(8)T, and 15.2(1)S, this feature was integrated. In Cisco IOS XE Release 2.1, this feature was implemented on Cisco ASR 1000 Series Routers. In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router. No commands were introduced or modified.

CHAPTER 3

MPLS VPN OSPF PE and CE Support

The MPLS VPN OSPF PE and CE Support feature allows service providers to configure Open Shortest Path First (OSPF) between provider edge (PE) and customer edge (CE) devices in a Multiprotocol Label Switching (MPLS) virtual private network (VPN). This feature increases flexibility when devices exchange routing information among sites because a separate router ID for each interface or subinterface is configured on a PE device attached to multiple CE devices within a VPN. An MPLS VPN consists of a set of sites that are interconnected by means of an MPLS provider core network. At each customer site, one or more CE devices attach to one or more PE devices.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for MPLS VPN OSPF PE and CE Support

- Configure MPLS Layer 3 VPNs.
- Configure the Border Gateway Protocol (BGP) in the core.

Information About MPLS VPN OSPF PE and CE Support

Overview of MPLS VPN OSPF PE and CE Support

This feature allows service providers to configure Open Shortest Path First (OSPF) between provider edge (PE) and customer edge (CE) devices in an MPLS VPN network.

This feature increases flexibility when devices exchange routing information among sites because a separate router ID for each interface or subinterface is configured on a PE device attached to multiple CE devices within a VPN.

How to Configure MPLS VPN OSPF PE and CE Support

Configuring OSPF as the Routing Protocol Between the PE and CE Devices

Perform this task to configure PE-to-CE routing sessions that use Open Shortest Path First (OSPF).

Note The Cisco implementation of OSPF in an MPLS VPN PE-CE environment is compliant with RFC 4576.

Before you begin

Configure the PE device with the same routing protocol that the CE device uses.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id* [**vrf** *vpn-name*]
4. **network** *ip-address wildcard-mask* **area** *area-id*
5. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
6. **redistribute** *protocol* | [**process-id**] | {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**]
7. **exit-address-family**
8. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* [**vrf** *vpn-name*] **Example:** `Device(config)# router ospf 1 vrf grc`	Enables OSPF routing and enters router configuration mode. • The *process-id* argument identifies the OSPF process. • The **vrf** *vpn-name* keyword and argument identify a virtual private network (VPN). Create a separate OSPF process for each virtual routing and forwarding (VRF) instance that will receive VPN routes.
Step 4	**network** *ip-address wildcard-mask* **area** *area-id* **Example:** `Device(config-router)# network 10.0.0.1 0.0.0.3 area 20`	Defines the interfaces on which OSPF runs and to defines the area ID for those interfaces. • The *ip-address* argument identifies the IP address. • The *wildcard-mask* argument identifies the IP-address-type mask that includes “don’t care” bits. • The *area-id* argument identifies the area that is to be associated with the OSPF address range. It can be specified as either a decimal value or an IP address. To associate areas with IP subnets, specify a subnet address as the value of the *area-id* argument.
Step 5	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 vrf vpn1`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**redistribute** *protocol* | [**process-id**] | {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**] **Example:** `Device(config-router-af)#  redistribute rip metric 1 subnets`	Redistributes routes from one routing domain into another routing domain. You may need to include several protocols to ensure that all interior Border Gateway Protocol (IBGP) routes are distributed into the VRF.

	Command or Action	Purpose
Step 7	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode.

Step 8	**end**
	Example:
	`Device(config-router)# end`
		(Optional) Exits to privileged EXEC mode.

Verifying Connectivity Between MPLS Virtual Private Network Sites

To verify that the local and remote customer edge (CE) devices can communicate across the Multiprotocol Label Switching (MPLS) core, perform the following tasks:

Verifying IP Connectivity from CE Device to CE Device Across the MPLS Core

SUMMARY STEPS

1. **enable**
2. **ping** [*protocol*] {*host-name* | *system-address*}
3. **trace** [*protocol*] [*destination*]
4. **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | *protocol* [*process-id*]] | [**list** [*access-list-name* | *access-list-number*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

Step 2 **ping** [*protocol*] {*host-name* | *system-address*}

Diagnoses basic network connectivity on AppleTalk, Connectionless-mode Network Service (CLNS), IP, Novell, Apollo, Virtual Integrated Network Service (VINES), DECnet, or Xerox Network Service (XNS) networks. Use the **ping** command to verify the connectivity from one CE device to another.

Step 3 **trace** [*protocol*] [*destination*]

Discovers the routes that packets take when traveling to their destination. The **trace** command can help isolate a trouble spot if two devices cannot communicate.

Step 4 **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | *protocol* [*process-id*]] | [**list** [*access-list-name* | *access-list-number*]

Displays the current state of the routing table. Use the *ip-address* argument to verify that CE1 has a route to CE2. Verify the routes learned by CE1. Make sure that the route for CE2 is listed.

Verifying That the Local and Remote CE Devices Are in the PE Routing Table

SUMMARY STEPS

1. **enable**
2. **show ip route vrf** *vrf-name* [*prefix*]
3. **show ip cef vrf** *vrf-name* [*ip-prefix*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

Step 2 **show ip route vrf** *vrf-name* [*prefix*]

Displays the IP routing table associated with a virtual routing and forwarding (VRF) instance. Check that the loopback addresses of the local and remote customer edge (CE) devices are in the routing table of the provider edge (PE) devices.

Step 3 **show ip cef vrf** *vrf-name* [*ip-prefix*]

Displays the Cisco Express Forwarding forwarding table associated with a VRF. Check that the prefix of the remote CE device is in the Cisco Express Forwarding table.

Configuration Examples for MPLS VPN OSPF PE and CE Support

Example: Configuring an MPLS VPN Using OSPF

PE Configuration	CE Configuration
ip vrf vpn1 rd 100:1 route-target export 100:1 route-target import 100:1 ! ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.1 255.255.255.255 ! interface FastEthernet0/0/0 ip vrf forwarding vpn1 ip address 34.0.0.2 255.0.0.0 no cdp enable ! router ospf 1000 vrf vpn1 log-adjacency-changes redistribute bgp 100 metric-type 1 subnets network 10.0.0.13 0.0.0.0 area 10000 network 34.0.0.0 0.255.255.255 area 10000 ! router bgp 100 no synchronization bgp log-neighbor changes neighbor 10.0.0.3 remote-as 100 neighbor 10.0.0.3 update-source Loopback0 no auto-summary ! address-family vpnv4 neighbor 10.0.0.3 activate neighbor 10.0.0.3 send-community extended bgp scan-time import 5 exit-address-family ! address-family ipv4 vrf vpn1 redistribute connected redistribute ospf 1000 match internal external 1 external 2 no auto-summary no synchronization exit-address-family	ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.9 255.255.255.255 ! interface FastEthernet0/0/0 ip address 34.0.0.1 255.0.0.0 no cdp enable ! router ospf 1000 log-adjacency-changes auto-cost reference-bandwidth 1000 redistribute connected subnets network 34.0.0.0 0.255.255.255 area 1000 network 10.0.0.0 0.0.0.0 area 1000

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco Master Command List, All Releases
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 4576	*Using a Link State Advertisement (LSA) Options Bit to Prevent Looping in BGP/MPLS IP Virtual Private Networks (VPNs)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS VPN OSPF PE and CE Support

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

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

Table 3: Feature Information for MPLS VPN OSPF PE and CE Support

Feature Name	Releases	Feature Information
MPLS VPN OSPF PE and CE Support	12.0(5)T 12.0(11)ST 12.0(21)ST 12.2(17b)SXA 12.2(28)SB Cisco IOS XE Release 2.1	The MPLS VPN OSPF PE and CE Support feature allows service providers to configure Open Shortest Path First (OSPF) between provider edge (PE) and customer edge (CE) devices in a Multiprotocol Label Switching (MPLS) virtual private network (VPN). In Cisco IOS Release 12.0(5)T, this feature was introduced. In Cisco IOS Release 12.0(11)ST, 12.0(21)ST, 12.2(17b)SXA, and 12.2(28)SB, this feature was integrated. In Cisco IOS XE Release 2.1, this feature was implemented on Cisco ASR 1000 Series Routers. No commands were introduced or modified.

CHAPTER 4

MPLS VPN Support for EIGRP Between PE and CE

The MPLS VPN Support for EIGRP Between PE and CE feature allows service providers to configure the Enhanced Interior Gateway Routing Protocol (EIGRP) between provider edge (PE) and customer edge (CE) devices in a Multiprotocol Label Switching (MPLS) virtual private network (VPN) and offer MPLS VPN services to those customers that require native support for EIGRP. An MPLS VPN consists of a set of sites that are interconnected by an MPLS provider core network. At each customer site, one or more CE devices attach to one or more PE devices.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for MPLS VPN Support for EIGRP Between PE and CE

- Configure MPLS Layer 3 VPNs.
- Configure the Border Gateway Protocol (BGP) in the network core.

Information About MPLS VPN Support for EIGRP Between PE and CE

Overview of MPLS VPN Support for EIGRP Between PE and CE

Using the Enhanced Interior Gateway Routing Protocol (EIGRP) between the provider edge (PE) and customer edge (CE) devices allows you to transparently connect EIGRP customer networks through an MPLS-enabled Border Gateway Protocol (BGP) core network so that EIGRP routes are redistributed through the VPN across the BGP network as internal BGP (iBGP) routes.

How to Configure MPLS VPN Support for EIGRP Between PE and CE

Configuring EIGRP as the Routing Protocol Between the PE and CE Devices

To configure PE-to-CE routing sessions that use EIGRP, perform this task.

Before you begin

Configure the PE device with the same routing protocol that the CE device uses.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no synchronization**
5. **neighbor** *ip-address* **remote-as** *as-number*
6. **neighbor** *ip-address* **update-source loopback** *interface-number*
7. **address-family vpnv4**
8. **neighbor** *ip-address* **activate**
9. **neighbor** *ip-address* **send-community extended**
10. **exit-address-family**
11. **address-family ipv4 vrf** *vrf-name*
12. **redistribute eigrp** *as-number* [**metric** *metric-value*] [**route-map** *map-name*]
13. **no synchronization**
14. **exit-address-family**
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 10`	Enters router configuration mode, and creates a BGP routing process.
Step 4	**no synchronization** **Example:** `Device(config-router)# no synchronization`	Configures BGP to send advertisements without waiting to synchronize with the IGP.
Step 5	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 10.0.0.1 remote-as 10`	Establishes peering with the specified neighbor or peer group. - In this step, you are establishing an iBGP session with the PE device that is connected to the CE device at the other CE site.
Step 6	**neighbor** *ip-address* **update-source loopback** *interface-number* **Example:** `Device(config-router)# neighbor 10.0.0.1 update-source loopback 0`	Configures BGP to use any operational interface for TCP connections. - This configuration step is not required. However, the BGP routing process will be less susceptible to the effects of interface or link flapping.
Step 7	**address-family vpnv4** **Example:** `Device(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions that use standard IPv4 address prefixes, such as BGP, RIP, and static routing sessions.
Step 8	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 activate`	Establishes peering with the specified neighbor or peer group. - In this step, you are activating the exchange of VPNv4 routing information between the PE devices.

	Command or Action	Purpose
Step 9	**neighbor** *ip-address* **send-community extended** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 send-community extended`	Configures the local device to send extended community attribute information to the specified neighbor. • This step is required for the exchange of EIGRP extended community attributes.
Step 10	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and enters router configuration mode.
Step 11	**address-family ipv4 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf RED`	Configures an IPv4 address family for the EIGRP VRF and enters address family configuration mode. • An address-family VRF needs to be configured for each EIGRP VRF that runs between the PE and CE devices.
Step 12	**redistribute eigrp** *as-number* [**metric** *metric-value*] [**route-map** *map-name*] **Example:** `Device(config-router-af)# redistribute eigrp 101`	Redistributes the EIGRP VRF into BGP. • The autonomous system number from the CE network is configured in this step.
Step 13	**no synchronization** **Example:** `Device(config-router-af)# no synchronization`	Configures BGP to send advertisements without waiting to synchronize with the IGP.
Step 14	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and enters router configuration mode.

Step 15	**end**
	Example:
	`Device(config-router)# end`
		Exits router configuration mode and enters privileged EXEC mode.

Configuring EIGRP Redistribution in the MPLS VPN

Perform this task on every PE device that provides VPN services to enable EIGRP redistribution in the MPLS VPN.

Before you begin

The metric must be configured for routes from external EIGRP autonomous systems and non-EIGRP networks before these routes can be redistributed into an EIGRP CE device. The metric can be configured in the redistribute statement using the **redistribute** (IP) command or can be configured with the **default-metric** (EIGRP) command. If an external route is received from another EIGRP autonomous system or a non-EIGRP network without a configured metric, the route will not be advertised to the CE device.

Note Redistribution between native EIGRP VRFs is not supported. This is designed behavior.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **network** *ip-address wildcard-mask*
6. **redistribute bgp** {*as-number*} [**metric** *bandwidth delay reliability load mtu*] [**route-map** *map-name*]
7. **autonomous-system** *as-number*
8. **exit-address-family**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *as-number* **Example:** `Device(config)# router eigrp 1`	Enters router configuration mode and creates an EIGRP routing process. • The EIGRP routing process for the PE device is created in this step.
Step 4	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 vrf RED`	Enters address-family configuration mode and creates a VRF. • The VRF name must match the VRF name that was created in the previous section.
Step 5	**network** *ip-address wildcard-mask*	Specifies the network for the VRF.

	Command or Action	Purpose
	Example: `Device(config-router-af)# network 172.16.0.0 0.0.255.255`	• The network statement is used to identify which interfaces to include in EIGRP. The VRF must be configured with addresses that fall within the wildcard-mask range of the network statement.
Step 6	**redistribute bgp** {*as-number*} [**metric** *bandwidth delay reliability load mtu*] [**route-map** *map-name*] **Example:** `Device(config-router-af)# redistribute bgp 10 metric 10000 100 255 1 1500`	Redistributes BGP into the EIGRP. • The autonomous system number and metric of the BGP network are configured in this step. BGP must be redistributed into EIGRP for the CE site to accept the BGP routes that carry the EIGRP information. A metric must also be specified for the BGP network and is configured in this step.
Step 7	**autonomous-system** *as-number* **Example:** `Device(config-router-af)# autonomous-system 101`	Specifies the autonomous system number of the EIGRP network for the customer site.
Step 8	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and enters router configuration mode.

Step 9	**end**
	Example:
	`Device(config-router)# end`
		Exits router configuration mode and enters privileged EXEC mode.

Verifying Connectivity Between MPLS Virtual Private Network Sites

To verify that the local and remote customer edge (CE) devices can communicate across the Multiprotocol Label Switching (MPLS) core, perform the following tasks:

Verifying IP Connectivity from CE Device to CE Device Across the MPLS Core

SUMMARY STEPS

1. **enable**
2. **ping** [*protocol*] {*host-name* | *system-address*}
3. **trace** [*protocol*] [*destination*]
4. **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | *protocol* [*process-id*]] | [**list** [*access-list-name* | *access-list-number*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

Step 2 **ping** [*protocol*] {*host-name* | *system-address*}

Diagnoses basic network connectivity on AppleTalk, Connectionless-mode Network Service (CLNS), IP, Novell, Apollo, Virtual Integrated Network Service (VINES), DECnet, or Xerox Network Service (XNS) networks. Use the **ping** command to verify the connectivity from one CE device to another.

Step 3 **trace** [*protocol*] [*destination*]

Discovers the routes that packets take when traveling to their destination. The **trace** command can help isolate a trouble spot if two devices cannot communicate.

Step 4 **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | *protocol* [*process-id*]] | [**list** [*access-list-name* | *access-list-number*]

Displays the current state of the routing table. Use the *ip-address* argument to verify that CE1 has a route to CE2. Verify the routes learned by CE1. Make sure that the route for CE2 is listed.

Verifying That the Local and Remote CE Devices Are in the PE Routing Table

SUMMARY STEPS

1. **enable**
2. **show ip route vrf** *vrf-name* [*prefix*]
3. **show ip cef vrf** *vrf-name* [*ip-prefix*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

Step 2 **show ip route vrf** *vrf-name* [*prefix*]

Displays the IP routing table associated with a virtual routing and forwarding (VRF) instance. Check that the loopback addresses of the local and remote customer edge (CE) devices are in the routing table of the provider edge (PE) devices.

Step 3 **show ip cef vrf** *vrf-name* [*ip-prefix*]

Displays the Cisco Express Forwarding forwarding table associated with a VRF. Check that the prefix of the remote CE device is in the Cisco Express Forwarding table.

Configuration Examples for MPLS VPN Support for EIGRP Between PE and CE

Example: Configuring an MPLS VPN Using EIGRP

PE Configuration	CE Configuration
	ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.9 255.255.255.255 ! interface FastEthernet0/0/0 ip address 34.0.0.1 255.0.0.0 no cdp enable ! router eigrp 1000 network 34.0.0.0 auto-summary

PE Configuration	CE Configuration
ip vrf vpn1 rd 100:1 route-target export 100:1 route-target import 100:1 ! ip cef mpls ldp router-id Loopback0 force mpls label protocol ldp ! interface Loopback0 ip address 10.0.0.1 255.255.255.255 interface FastEthernet0/0/0 ip vrf forwarding vpn1 ip address 34.0.0.2 255.0.0.0 no cdp enable interface FastEthernet1/1/0 ip address 30.0.0.1 255.0.0.0 mpls label protocol ldp mpls ip router eigrp 1000 auto-summary ! address-family ipv4 vrf vpn1 redistribute bgp 100 metric 10000 100 255 1 1500 network 34.0.0.0 distribute-list 20 in no auto-summary autonomous-system 1000 exit-address-family ! router bgp 100 no synchronization bgp log-neighbor changes neighbor 10.0.0.3 remote-as 100 neighbor 10.0.0.3 update-source Loopback0 no auto-summary ! address-family vpnv4 neighbor 10.0.0.3 activate neighbor 10.0.0.3 send-community extended bgp scan-time import 5 exit-address-family ! address-family ipv4 vrf vpn1 redistribute connected redistribute eigrp no auto-summary no synchronization exit-address-family	

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco Master Command List, All Releases
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 4576	*Using a Link State Advertisement (LSA) Options Bit to Prevent Looping in BGP/MPLS IP Virtual Private Networks (VPNs)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS VPN Support for EIGRP Between PE and CE

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 4: Feature Information for MPLS VPN Support for EIGRP Between PE and CE

Feature Name	Releases	Feature Information
MPLS VPN Support for EIGRP Between PE and CE	12.0(22)S 12.2(15)T 12.2(18)S 12.2(18)SXD 12.2(27)SBB 12.3(2)T Cisco IOS XE Release 2.1	The MPLS VPN Support for EIGRP Between PE and CE feature allows service providers to configure the Enhanced Interior Gateway Routing Protocol (EIGRP) between provider edge (PE) and customer edge (CE) devices in a Multiprotocol Label Switching (MPLS) virtual private network (VPN) and offer MPLS VPN services to those customers that require native support for EIGRP. In Cisco IOS Release 12.0(22)S, this feature was introduced. In Cisco IOS Release 12.2(15)T, 12.2(18)S, 12.2(18)SXD, 12.2(27)SBB, and 12.3(2)T, this feature was integrated. In Cisco IOS XE Release 2.1, this feature was implemented on Cisco ASR 1000 Series Routers. No commands were introduced or modified.

CHAPTER 5

IPv6 VPN over MPLS

The Border Gateway Protocol over Multiprotocol Label Switching VPN feature is an implementation of the provider edge (PE)-based Virtual Private Network (VPN) model. In principle, there is no difference between IPv4 and IPv6 VPNs. In both IPv4 and IPv6, multiprotocol Border Gateway Protocol (BGP) is the center of the Multiprotocol Label Switching (MPLS) VPN for IPv6 (VPNv6) architecture. It is used to distribute IPv6 routes over the service provider backbone, using the same procedures to work with overlapping addresses, redistribution policies, and scalability issues.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for IPv6 VPN over MPLS

Your network must be running the following services before you configure IPv6 VPN operation:

- Multiprotocol Label Switching (MPLS) in provider backbone devices
- MPLS with Virtual Private Network (VPN) code in provider devices with VPN provider edge (PE) devices
- Border Gateway Protocol (BGP) in all devices providing a VPN service

- Cisco Express Forwarding switching in every MPLS-enabled device
- Class of Service (CoS) feature

Restrictions for IPv6 VPN over MPLS

IPv6 VPN over MPLS (6VPE) supports a Multiprotocol Label Switching (MPLS) IPv4-signaled core. An MPLS IPv6-signaled core is not supported.

Information About IPv6 VPN over MPLS

IPv6 VPN over MPLS Overview

Multiprotocol Border Gateway Protocol (BGP) is the center of the Multiprotocol Label Switching (MPLS) IPv6 Virtual Private Network (VPN) architecture in both IPv4 and IPv6. It is used to distribute IPv6 routes over the service provider backbone, using the same procedures to work with overlapping addresses, redistribution policies, and scalability issues.

Although IPv6 should not have overlapping address space, IPv6 addresses are prepended with a route distinguisher (RD). A network layer reachability information (NLRI) 3-tuple format (which contains length, IPv6 prefix, and label) is defined to distribute these routes using multiprotocol BGP. The extended community attribute (for example, the route target) is used to control redistribution of routing information by tagging exported routes and filtering imported ones.

For scalability, route reflectors can be used to concentrate routing paths and avoid a full provider edge (PE) mesh. BGP features in IPv6, such as route refresh, automatic route filtering, and outbound route filtering, help reduce the number of routes held in each PE. This document focuses on the following differences between IPv6 and IPv4:

- Creation of a new multiprotocol BGP IPv6 VPN address family and specification of a IPv6 VPN address format
- Specification of a new IPv6 VPN NLRI
- Specification of BGP next-hop encoding when the device has an IPv4-based MPLS core

Some IPv6 VPN features, such as interprovider and Carrier Supporting Carrier (CSC) topologies, are specific to BGP-MPLS IPv6 VPN. Others, such as the link between Autonomous System Boundary Routers (ASBRs), might support IPv4 only, IPv6 only, or both, regardless of the address family being transported.

Addressing Considerations for IPv6 VPN over MPLS

Regardless of the Virtual Private Network (VPN) model deployed, an addressing plan must be defined for the VPN that allows hosts to communicate with other sites using one site within one VPN, as well as with public resources.

VPN IPv4 sites often use private addressing for their addressing plan. These addresses do not need to be registered, and they are not routable on the public network. Whenever a host within a private site needs to access a public domain, it goes through a device that finds a public address on its behalf. With IPv4, this can be a network address translator or an application proxy.

Given the larger address space available with IPv6, the easiest approach to IPv6 addressing is to use IPv6 global addresses for the private addressing plan. Another approach is to use unique local addresses (ULAs). ULAs are easy to filter at site boundaries based on their scope. ULAs are also Internet service provider (ISP)-independent and can be used for communications inside a site without any permanent or intermittent Internet connectivity.

In IPv6 VPN over MPLS (6VPE), ULAs are treated as regular global addresses. The device configuration filters ULA prefixes to prevent them from appearing in the public domain. Link-local addresses on the peer will not be announced by Border Gateway Protocol (BGP) (IPv6 or IPv6 VPN) speakers.

A host within a private site that needs to access a public domain can do so through an IPv6 application proxy (such as a web proxy for accessing web pages), which accesses the public resource on the host's behalf with a global routable address, or the host can use a public address of its own. In the latter case, if ULAs have been deployed, the IPv6 host also is configured with a routable global address. A source address selection algorithm is used to select one or the other, based on the destination address.

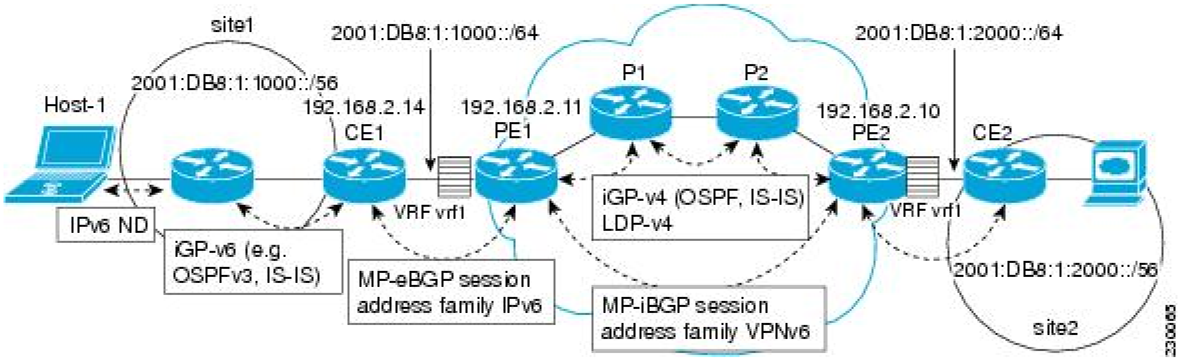
Basic IPv6 VPN over MPLS Functionality

IPv6 VPN over MPLS (6VPE) takes advantage of the coexistence between IPv6 and IPv4 by leveraging an existent Multiprotocol Label Switching (MPLS) IPv4 core network:

IPv6 VPN Architecture Overview

The figure below illustrates the important aspects of the IPv6 Virtual Private Network (VPN) architecture.

Figure 3: Simple IPv6 VPN Architecture



The customer edge (CE) devices are connected to the provider's backbone using provider edge (PE) devices. The PE devices are connected using provider (P1 and P2 in the figure above) devices. The provider (P) devices are unaware of VPN routes, and, in the case of IPv6 over MPLS (6VPE), might support only IPv4. Only PE devices perform VPN-specific tasks. For 6VPE, the PE devices are dual-stack (IPv4 and IPv6) devices.

The routing component of the VPN operation is divided into core routing and edge routing. Core routing, which involves PE devices and P devices, typically is performed by an IPv4 Interior Gateway Protocol (IGP) such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS). In the figure above, the IGP distributes only routes internal to the provider's autonomous system. The core routing enables connectivity among P and PE devices.

Edge routing takes place in two directions: routing between PE pairs and routing between a PE and a CE. Routing between PE pairs is achieved using multiprotocol internal BGP (iBGP) using the IPv6 VPN address family. This method distributes routes learned from CEs through PE-CE routing, using appropriate route export policies at the ingress PE device and appropriate route import policies at the egress PE device.

Routing between the CE and its PE is achieved using a routing protocol that is VPN routing and forwarding (VRF) aware. Static routes, external BGP (eBGP), and Enhanced Interior Gateway Routing Protocol (EIGRP) are VRF-instance aware. In the figure above, eBGP is used between the CE (CE1) and the PE (PE1). At the same time, the CE runs an IPv6 IGP within the VPN site (site1 in the figure above). The CE redistributes IGP routes into multiprotocol-eBGP address family IPv6. At the PE, these routes are installed in the VRF named vrf1, and forwarded to the remote PEs (PE2 in the figure above), according to export policies defined for this VRF.

IPv6 VPN Next Hop

When the device announces a prefix using the MP_REACH_NLRI attribute, the Multiprotocol Border Gateway Protocol (MP-BGP) running on one provider edge (PE) inserts a BGP next hop in the update message sent to a remote PE. This next hop is either propagated from the received update (for instance, if the PE is a route reflector), or it is the address of the PE sending the update message (the egress PE).

For the IPv6 Virtual Private Network (VPN) address family, the next hop must be an IPv6 VPN address, regardless of the nature of the network between the PE speakers. Because the route distinguisher (RD) has no significance (the address is not part of any VPN), it is set to 0. If the provider network is a native IPv6 network, the remaining part of the next hop is the IPv6 address of the egress PE. Otherwise, it is an IPv4 address used as an IPv6-mapped address (for example, ::FFFF:IPv4-address).

MPLS Forwarding

When it receives IPv6 traffic from one customer site, the ingress provider edge (PE) device uses Multiprotocol Label Switching (MPLS) to tunnel IPv6 Virtual Private Network (VPN) packets over the backbone toward the egress PE device identified as the Border Gateway Protocol (BGP) next hop. The ingress PE device prepends the IPv6 packets with the outer and inner labels before putting the packet on the egress interface.

Under normal operation, a provider (P) device along the forwarding path does not look inside the frame beyond the first label. The provider (P) device either swaps the incoming label with an outgoing one or removes the incoming label if the next device is a PE device. Removing the incoming label is called penultimate hop popping. The remaining label (BGP label) is used to identify the egress PE interface toward the customer site. The label also hides the protocol version (IPv6) from the last P device, which it would otherwise need to forward an IPv6 packet.

A P device is ignorant of the IPv6 VPN routes. The IPv6 header remains hidden under one or more MPLS labels. When the P device receives an MPLS-encapsulated IPv6 packet that cannot be delivered, it has two options. If the P device is IPv6 aware, it exposes the IPv6 header, builds an Internet Control Message Protocol (ICMP) for IPv6 message, and sends the message, which is MPLS encapsulated, to the source of the original packet. If the P device is not IPv6 aware, it drops the packet.

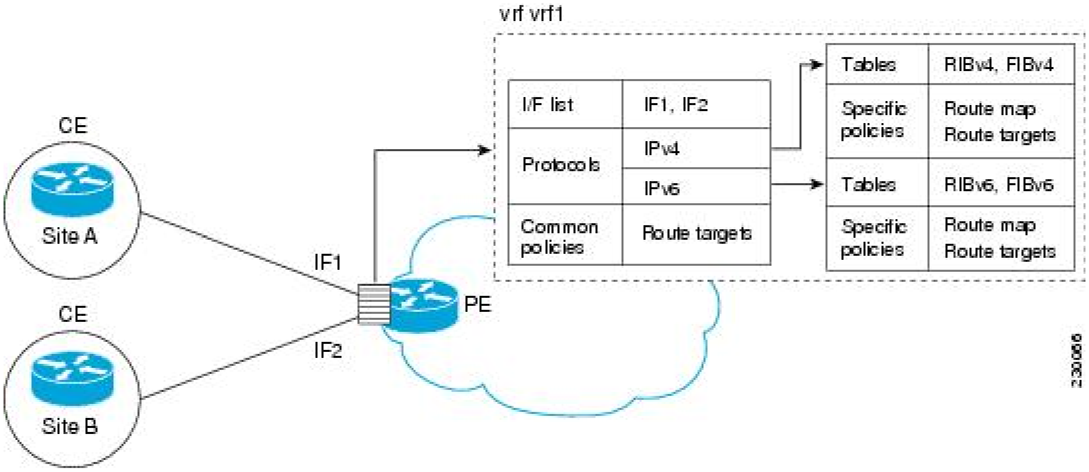
VRF Concepts

A virtual routing and forwarding (VRF) entity works with a private customer-specific Routing Information Base (RIB) and Forwarding Information Base (FIB). Although IPv4 and IPv6 routing tables are distinct, it is convenient for the two protocols to share the same VRF for a specific customer.

IPv6 VPN customers are likely to be existing VPNv4 customers that are either deploying dual-stack hosts and devices or shadowing some of their IPv4 infrastructure with IPv6 nodes. Several deployment models are possible. Some customers use separate logical interfaces for IPv4 and IPv6 and define separate VRFs on each. Although this approach provides flexibility to configure separate policies for IPv4 and IPv6, it prevents sharing the same policy. Another approach, the multiprotocol VRF, keeps a single VRF on the provider edge-customer edge (PE-CE) interface, and enables it for IPv4, IPv6, or both. It is then possible to define common or separate policies for each IP version. With this approach, a VRF is better defined as the set of tables, interfaces, and policies found at the PE, and is used by sites of a particular VPN connected to this PE.

The figure below illustrates the multiprotocol VRF, in which the VRF named vrf1is enabled for both IPv4 and IPv6 and is associated with two interfaces (IF1, IF2), two sets of tables (IPv4 RIB and FIB and IPv6 RIB and FIB), and a set of common or distinct policies.

Figure 4: Multiprotocol VRF



IPv6 VPN Scalability

Provider edge (PE)-based Virtual Private Networks (VPNs) such as Border Gateway Protocol-Multiprotocol Label Switching (BGP-MPLS) IPv6 VPN scale better than customer edge (CE)-based VPNs. A network designer must consider scaling when designing the network. The following points need to be considered:

- Routing table size, which includes the size of virtual routing and forwarding (VRF) tables and BGP tables
- Number of BGP sessions, which grows as a square number of PEs

Routing table size concerns occur with PEs that handle many customer sites. Not only do these PEs have one Routing Information Base (RIB) and Forwarding Information Base (FIB) per connected customer, but also the PEs' BGP tables, which total all entries from individual VRFs, grow accordingly. Another scalability problem occurs when the number of PEs in the provider network grows beyond a certain level. Assuming that a significant number of sites belonging to the same VPN are spread over many PEs, the number of multiprotocol BGP sessions may rapidly become prohibitive: (n -1) x n /2, where n is the number of PEs.

The following features are included in IPv6 VPN over MPLS:

- Route refresh and automatic route filtering—Limits the size of routing tables, because only routes imported into a VRF are kept locally. When the import policy changes, a route refresh can be sent to query a retransmission of routing updates.
- Outbound route filtering (ORF)—Allows the ingress PE to advertise filters to the egress PE so that updates are not sent unnecessarily over the network.

- Route reflectors—Route reflectors (RRs) are internal BGP (iBGP) peers that propagate iBGP routes learned from other iBGP peers. RRs are used to concentrate iBGP sessions.

Advanced IPv6 MPLS VPN Functionality

Advanced Multiprotocol Label Switching (MPLS) features such as accessing the Internet from a Virtual Private Network (VPN) for IPv4, multiautonomous-system backbones, and Carrier Supporting Carriers (CSCs) are generally the same for IPv6 as for IPv4. However, there are differences in addressing and in the way IPv6 over MPLS (6VPE) operates over an IPv4 backbone.

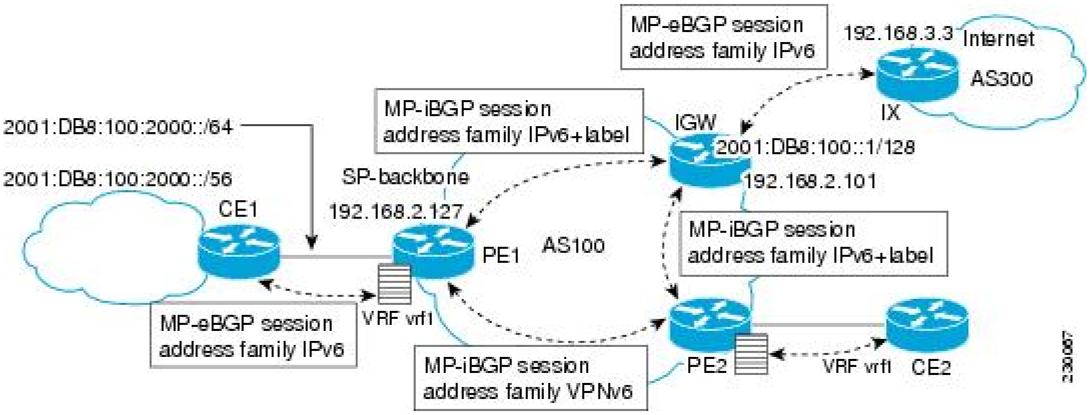
The following sections describe concepts for advanced IPv6 MPLS VPN functionality:

Internet Access

Most Virtual Private Network (VPN) sites require access to the Internet. RFC 4364 describes a set of models for enabling IPv4 and IPv6 VPN access to the Internet. In one model, one interface is used by the customer edge (CE) to connect to the Internet and a different one to connect to the virtual routing and forwarding (VRF) instance. Another model is in which all Internet routes are redistributed into the VRF; however, this approach has the disadvantage of requiring the Internet routes be replicated in each VRF.

In one scenario, a static route is inserted into the VRF table, with a next hop that points to the Internet gateway found in the IPv6 default table. The figure below illustrates this scenario, in which Internet access is provided to the customer in the VRF named vrf1.

Figure 5: Internet Access Topology



A customer site that has access public resources over the Internet must be known by a public prefix. Unlike IPv4, IPv6 does not offer a Network Address Translation (NAT) mechanism that translates private addresses into public addresses when leaving the site boundaries. This implies that hosts within the site speak with public addresses and appear in the public domain.

For outbound traffic, the default route configured in the VRF table at ingress provider edge (PE1) directs traffic for destinations outside the VPN to the Internet gateway.

For inbound traffic, a route must exist at the Internet gateway to direct the traffic for a customer site via its PE of attachment (PE1 in the figure above). This route can be distributed by the ingress PE (PE1) using multiprotocol internal Border Gateway Protocol (iBGP) (with the IPv6 address family configuration), so no specific configuration is needed on a per-VPN PE basis at the Internet gateway. Nevertheless, for inbound traffic at PE1, a route must exist in the default table for the customer site global prefix pointing to the VRF of the site.

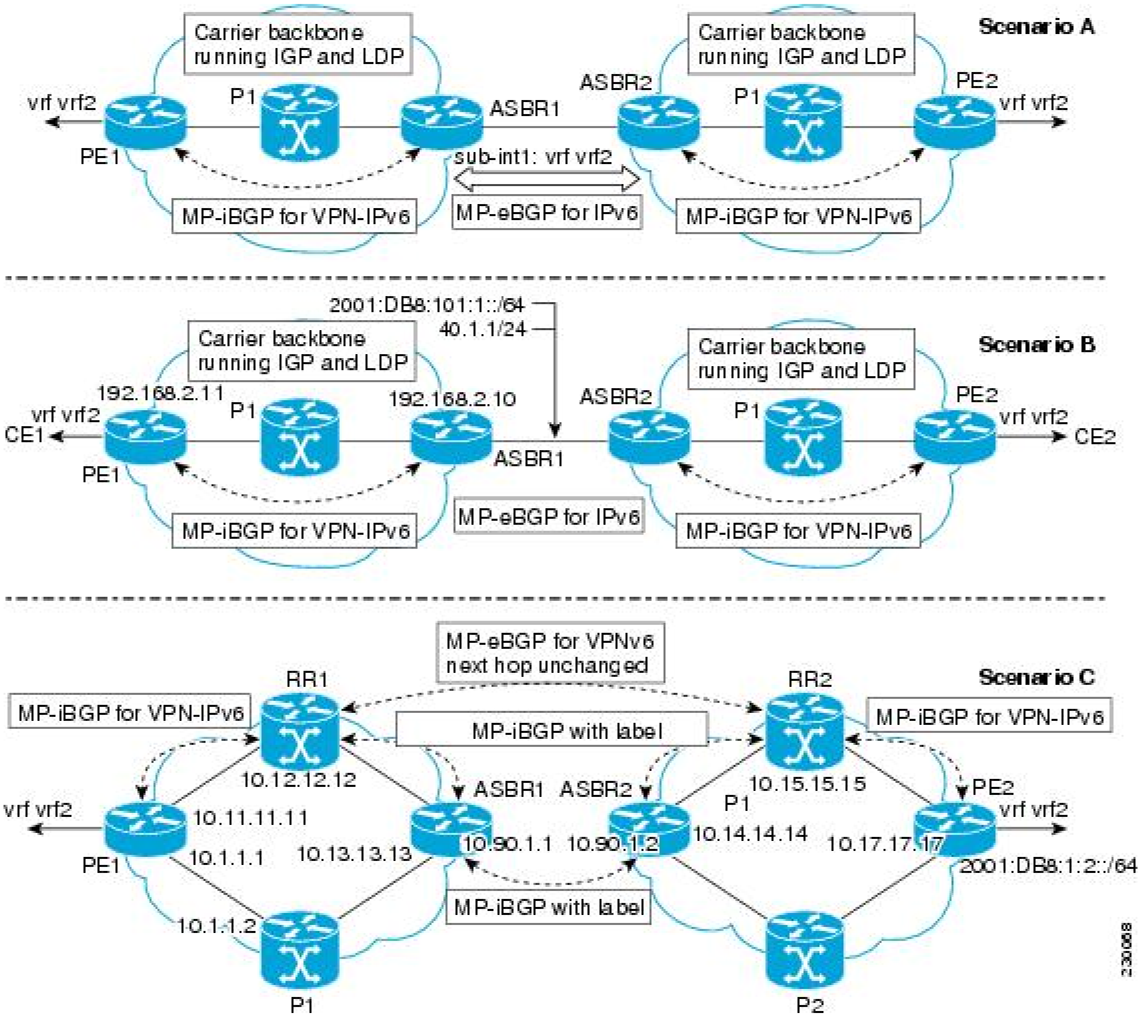
Multiautonomous-System Backbones

The problem of interprovider Virtual Private Networks (VPNs) is similar for IPv6 and IPv4, assuming that IPv6 was deployed everywhere IPv4 was deployed.

In IPv6 deployments that cross autonomous system boundaries, providers may have to obtain a peering model, or work with the peering model put in place for VPNv4.

The figure below illustrates interprovider scenarios in IPv6 VPN.

Figure 6: Interprovider Scenarios



Depending on the network protocol used between Autonomous System Boundary Routers (ASBRs), the three scenarios shown in the figure above can have several implementation options. For instance, scenario B, which suggests a multiprotocol external Border Gateway Protocol (eBGP) IPv6 VPN peering between ASBRs, could use either an IPv6 or an IPv4 link.

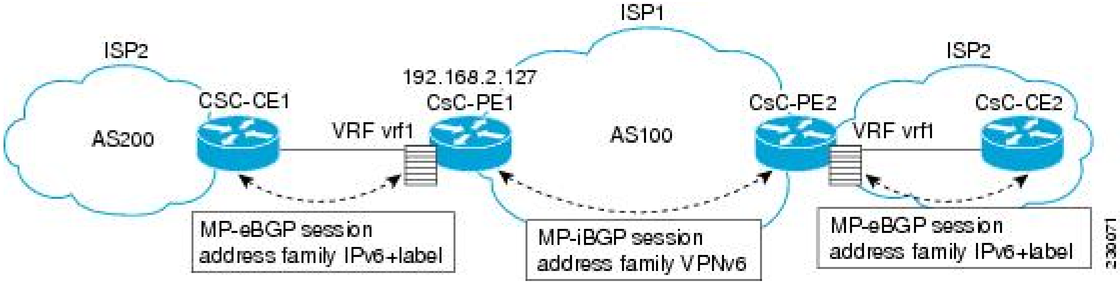
In scenario C, multihop multiprotocol eBGP redistributes IPv6 VPN routes across route reflectors in different autonomous systems. Labeled IPv4 routes to the provider edge (PE) devices (in the IPv6 over MPLS case) need to be advertised across ASBRs so that a complete labeled switch path is set up end to end.

Carrier Supporting Carriers

The Carrier Supporting Carrier (CSC) feature provides Virtual Private Network (VPN) access to a customer service provider, so this service needs to exchange routes and send traffic over the Internet service provider (ISP) Multiprotocol Label Switching (MPLS) backbone. The only difference from a regular provider edge (PE) is that it provides MPLS-to-MPLS forwarding on the CSC-customer edge (CE) to CSC-PE interface, rather than IP-to-MPLS forwarding.

The figure below highlights the two ISPs' interface.

Figure 7: CSC IPv6 over MPLS Configuration Example



How to Configure IPv6 VPN over MPLS

Configuring a Virtual Routing and Forwarding Instance for IPv6

A virtual routing and forwarding (VRF) instance is an address family-independent object that can be enabled and configured for each of the supported address families. Configuring a VRF consists of the following three steps:

- Configuring the address-family-independent part of the VRF
- Enabling and configuring IPv4 for the VRF
- Enabling and configuring IPv6 for the VRF

A VRF is given a name and a route distinguisher (RD). The RD is configured outside the context of the address family, although the RD is used to distinguish overlapping addresses within the context of a particular Border Gateway Protocol (BGP) address family. Having separate RDs for IPv4 VPN addresses and IPv6 VPN addresses does not matter. On Cisco devices, the RDs are the same in order to simplify configuration and VPN management.

Users can configure policies in common between IPv4 and IPv6 when not using an address family context. This feature is shared route targets (import and export), and it is useful in a migration scenario, where IPv4 policies already are configured and IPv6 policies should be the same as the IPv4 policies.

The IPv4 and IPv6 address family can each be enabled and configured separately. Note that the route-target policies entered at this level override global policies that may have been specified during address family-independent configuration.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
6. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
7. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
8. **exit**
9. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**]
10. **route-target** {**import** | **export** | **both**} *route-target-ext-community*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vrf1`	Configures a VPN VRF routing table and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:1`	Specifies the RD for a VRF.
Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Device(config-vrf)# route target import 100:10`	Specifies the route target VPN extended communities for both IPv4 and IPv6.
Step 6	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **Example:** `Device(config)# address-family ipv4`	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 7	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:**	Specifies the route target VPN extended communities specific to IPv4.

	Command or Action	Purpose
	`Device(config-vrf-af)# route target import 100:11`	
Step 8	**exit** **Example:** `Device(config-vrf-af)# exit`	Exits address family configuration mode on this VRF.
Step 9	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**] **Example:** `Device(config-vrf)# address-family ipv6`	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv6 address prefixes.
Step 10	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Device(config-vrf-af)# route target import 100:12`	Specifies the route target VPN extended communities specific to IPv6.

Binding a VRF to an Interface

In order to specify which interface belongs to which virtual routing and forwarding (VRF) instance, use the **vrf forwarding** command for both IPv4 and IPv6. An interface cannot belong to more than one VRF. When the interface is bound to a VRF, previously configured addresses (IPv4 and IPv6) are removed, and they must be reconfigured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask* [**secondary**]
6. **ipv6 address** {*ipv6-address* / *prefix-length* | *prefix-name sub-bits*/*prefix-length*

DETAILED STEPS

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

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding vrf1`	Associates a VPN VRF with an interface or subinterface. • Note that any address, IPv4 or IPv6, that was configured prior to entering this command will be removed.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.10.10.1 255.255.255.0`	Configures an IPv4 address on the interface.
Step 6	**ipv6 address** {*ipv6-address* / *prefix-length* | *prefix-name sub-bits*/*prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:100:1::1/64`	Configures an IPv6 address on the interface.

Configuring a Static Route for PE-to-CE Routing

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix* / *prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*]

DETAILED STEPS

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

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix* / *prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*] **Example:** `Device(config)# ipv6 route vrf vrf1 ::/0 2001:DB8:200::1 nexthop-vrf default`	Installs the specified IPv6 static route using the specified next hop.

Configuring eBGP PE-to-CE Routing Sessions

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**]
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**] **Example:** `Device(config-router)# address-family ipv6 vrf vrf1`	Enters address family configuration mode.

	Command or Action	Purpose
Step 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router-af)# neighbor 2001:DB8:100:1::2 remote-as 200`	Adds an entry to the multiprotocol BGP neighbor table.
Step 6	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 2001:DB8:100:1::2 activate`	Enables the exchange of information for this address family with the specified BGP neighbor.

Configuring the IPv6 VPN Address Family for iBGP

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family vpnv6** [**unicast**]
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
9. **extended**] **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the BGP routing process.

	Command or Action	Purpose
Step 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.2.11 remote-as 100`	Adds an entry to the multiprotocol Border Gateway Protocol (BGP) neighbor table. • In IPv6 VPN, the peer address typically is an IPv4 address, in order to enable the BGP session to be transported over the IPv4-based core network.
Step 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 192.168.2.11 update-source Loopback 0`	Enables the BGP session to use a source address on the specified interface.
Step 6	**address-family vpnv6** [**unicast**] **Example:** `Device(config-router)# address-family vpnv6`	Places the device in address family configuration mode for configuring routing sessions.
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.2.11 activate`	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **Example:** `Device(config-router-af)# neighbor 192.168.2.11 send-community extended`	Specifies that a communities attribute should be sent to the BGP neighbor.
Step 9	**extended**] **exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode.

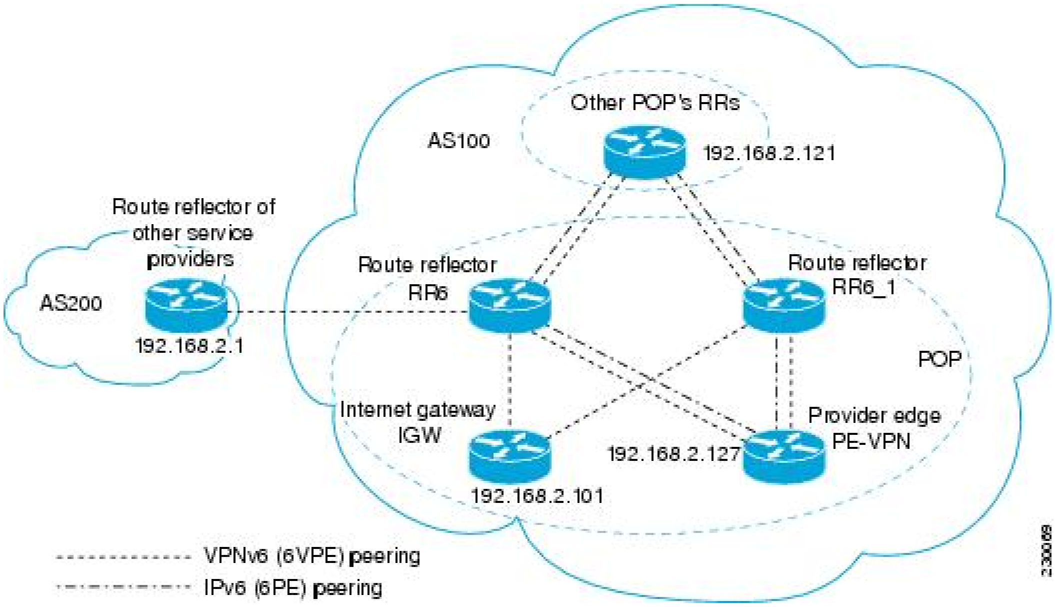
Configuring Route Reflectors for Improved Scalability

In this task, two route reflectors (RRs) are configured for redundancy reasons. Deploying RRs improves scalability by drastically reducing the number of Border Gateway Protocol (BGP) sessions. One RR usually peers with many internal Border Gateway Protocol (iBGP) speakers, preventing a full mesh of BGP sessions.

In a Multiprotocol Label Switching (MPLS)-based core, RRs are not part of the label switch paths and can be located anywhere in the network. For example, in a flat RR design, RRs can be deployed at Level 1 points of presence (POPs) and peer together in a full-mesh topology. In a hierarchical RR design, RRs could be deployed at Level 1 and Level 2 POPs, with Level 1 POPs peering together and with Level 2 RRs.

In a typical case where IPv6 over MPLS (6VPE) is deployed in a preexisting MPLS network (for example, providing VPNv4 services), it is likely that some RR design is already in place, and a similar RR infrastructure for IPv6 Virtual Private Network (VPN) services can be deployed. The figure below illustrates the main peering points between the RR in the ISP POP and the set of its RR clients.

Figure 8: Route Reflector Peering Design



The following list of BGP RR clients must be configured at each IPv6 RR (RR6 and RR6_1 in the figure above) device, at each POP:

- Provider edge (PE) devices (PE-VPN) of the POP providing IPv6 VPN access to the ISP customers. This includes both IPv6 VPN (6VPE) peering for interconnecting customer sites and IPv6 peering (6PE) for providing Internet access to VPN customers (see the “Configuring Internet Access” section).
- Internet gateway (IGW) located in the POP in order to provide PE customers with access to the IPv6 Internet (see the see the “Configuring Internet Access” section).
- RRs from other service providers. This feature is used to provide interautonomous-system connectivity, and it includes both IPv6 and IPv6 VPN peering. This service is described in the “Configuring a Multiautonomous-System Backbone for IPv6 VPN” section.
- RRs in other POPs. All RRs peer together, with both IPv6 and IPv6 VPN address families enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*

9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
10. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
11. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
12. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]
13. **address-family ipv6**
14. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
15. **neighbor** { *ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
16. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
17. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
18. **neighbor** { *ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
19. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
20. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
21. **neighbor** { *ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
22. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
23. **exit**
24. **address-family vpnv6** [**unicast**
25. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
26. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
27. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
28. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
29. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
30. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
31. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
32. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
33. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
34. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop-unchanged** [**allpaths**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the BGP routing process.

	Command or Action	Purpose
Step 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.2.101 remote-as 100`	Adds an entry to the multiprotocol BGP neighbor table, and provides peering with the Internet gateway in order to provide Internet access.
Step 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 192.168.2.101 update-source Loopback 0`	Enables the BGP session to use a source address on the specified interface.
Step 6	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.2.121 remote-as 100`	Adds an entry to the multiprotocol BGP neighbor table, and provides peering with the other POP's RR.
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 192.168.2.121 update-source Loopback 0`	Enables the BGP session to use a source address on the specified interface.
Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.2.127 remote-as 100`	Adds an entry to the multiprotocol BGP neighbor table.
Step 9	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 192.168.2.127 update-source Loopback 0`	Enables the BGP session to use a source address on the specified interface.
Step 10	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.2.1 remote-as 200`	(Optional) Adds an entry to the multiprotocol BGP neighbor table, and provides peering with the RR of the peer ISP in order to provide inter-VPN service.

	Command or Action	Purpose
Step 11	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 192.168.2.1 update-source Loopback 0`	(Optional) Enables the BGP session to use a source address on the specified interface.
Step 12	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*] **Example:** `Device(config-router)# neighbor 192.168.2.1 ebgp-multihop`	(Optional) Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.
Step 13	**address-family ipv6** **Example:** `Device(config-router)# address-family ipv6`	(Optional) Enters address family configuration mode in order to provide Internet access service.
Step 14	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.2.101 activate`	(Optional) Enables the exchange of information for this address family with the specified neighbor.
Step 15	**neighbor** { *ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **Example:** `Device(config-router-af)# neighbor 192.168.2.101 send-label`	(Optional) Enables a BGP device to send MPLS labels with BGP routes to a neighboring BGP device.
Step 16	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client** **Example:** `Device(config-router-af)# neighbor 192.168.2.101 route-reflector-client`	(Optional) Configures the device as a BGP route reflector and configures the specified neighbor as its client.
Step 17	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.2.121 activate`	(Optional) Enables the exchange of information for this address family with the specified BGP neighbor.

	Command or Action	Purpose
Step 18	**neighbor** { *ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **Example:** `Device(config-router-af)# neighbor 192.168.2.121 send-label`	(Optional) Enables a BGP device to send MPLS labels with BGP routes to a neighboring BGP device.
Step 19	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client** **Example:** `Device(config-router-af)# neighbor 192.168.2.121 route-reflector-client`	(Optional) Configures the specified neighbor as a route reflector client.
Step 20	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.2.127 activate`	(Optional) Enables the exchange of information for this address family with the specified BGP neighbor.
Step 21	**neighbor** { *ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **Example:** `Device(config-router-af)# neighbor 192.168.2.127 send-label`	(Optional) Enables a BGP device to send MPLS labels with BGP routes to a neighboring BGP device.
Step 22	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client** **Example:** `Device(config-router-af)# neighbor 192.168.2.127 route-reflector-client`	(Optional) Configures the specified neighbor as a route reflector client.
Step 23	**exit** **Example:** `Device(config-router-af)# exit`	(Optional) Exits address family configuration mode.
Step 24	**address-family vpnv6** [**unicast** **Example:** `Device(config-router)# address-family vpnv6`	Places the device in address family configuration mode for configuring routing sessions.
Step 25	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:**	Enables the exchange of information for this address family with the specified BGP neighbor.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor 192.168.2.121 activate`	
Step 26	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **Example:** `Device(config-router-af)# neighbor 192.168.2.21 send-community extended`	Specifies that a communities attribute should be sent to the BGP neighbor.
Step 27	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client** **Example:** `Device(config-router-af)# neighbor 192.168.2.121 route-reflector-client`	Configures the specified neighbor as a route reflector client.
Step 28	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.2.127 activate`	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 29	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **Example:** `Device(config-router-af)# neighbor 192.168.2.127 send-community extended`	Specifies that a communities attribute should be sent to the BGP neighbor.
Step 30	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client** **Example:** `Device(config-router-af)# neighbor 192.168.2.127 route-reflector-client`	Configures the specified neighbor as a route reflector client.
Step 31	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.2.1 activate`	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 32	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **Example:**	Specifies that a communities attribute should be sent to the BGP neighbor.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor 192.168.2.1 send-community extended`	
Step 33	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client** **Example:** `Device(config-router-af)# neighbor 192.168.2.1 route-reflector-client`	Configures the specified neighbor as a route reflector client.
Step 34	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop-unchanged** [**allpaths** **Example:** `Device(config-router-af)# neighbor 192.168.2.1 next-hop-unchanged allpaths`	Enables an EBGP multihop peer to propagate to the next hop unchanged for paths.

Configuring Internet Access

Customers with IPv6 Virtual Private Network (VPN) access need to have access to the Internet through IPv6. The design of this service is similar to a global Internet access service. IPv6 VPN over MPLS (6VPE) devices located in a Level 1 point of presence (POP) (colocated with an IGW device) can access the Internet gateway (IGW) natively, whereas 6VPE devices located in Level 2 and Level 3 POPs with no direct access to the IGW can access the IGW in their closest Level 1 POP over 6PE.

Configuring VPN Internet access in such a 6VPE device involves configuring Border Gateway Protocol (BGP) peering with the IGW (in most cases through the IPv6 RR, as described in the "Configuring Route Reflectors for Improved Scalability" section). Then the user must configure cross-table routing to enable communication between the private domain (the VRF) and the public domain (the Internet).

The figure above illustrates the following configuration tasks:

Configuring the Internet Gateway

Configuring iBGP 6PE Peering to the VPN PE

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family ipv6**
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.2.127 remote-as 100`	Adds an entry to the multiprotocol BGP neighbor table to provide peering with the Virtual Private Network (VPN) provider edge (PE).
Step 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 192.168.2.127 update-source Loopback 0`	Enables the BGP session to use a source address on the specified interface.
Step 6	**address-family ipv6** **Example:** `Device(config-router)# address-family ipv6`	Enters address family configuration mode in order to exchange global table reachability.
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.2.127 activate`	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **Example:** `Device(config-router-af)# neighbor 192.168.2.127 send-label`	Enables a BGP device to send Multiprotocol Label Switching (MPLS) labels with BGP routes to a neighboring BGP device, and allows the PE VPN to reach the Internet gateway over MPLS.

Configuring the Internet Gateway as the Gateway to the Public Domain

Use the 6PE peering configuration established in the "Configuring iBGP 6PE Peering to the VPN PE" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv6**
5. **network** *ipv6-address/prefix-length*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	**address-family ipv6** **Example:** `Device(config-router)# address-family ipv6`	Enters address family configuration mode in order to exchange global table reachability.
Step 5	**network** *ipv6-address/prefix-length* **Example:** `Device(config-router-af)# network 2001:DB8:100::1/128`	Configures the network source of the next hop to be used by the provider edge (PE) Virtual Private Network (VPN).
Step 6	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode.

Configuring eBGP Peering to the Internet

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **address-family ipv6**
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
7. **aggregate-address** *address mask* [**as-set**] [**summary-only**] [**suppress-map** *map-name*] [**advertise-map** *map-name*] [**attribute-map** *map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor FE80::300::1 GigabitEthernet0/0/0 remote-as 300`	Adds an entry to the multiprotocol BGP neighbor table, and provides peering with PE (PE-VPN). • Note that the peering is done over link-local addresses.
Step 5	**address-family ipv6** **Example:** `Device(config-router)# address-family ipv6`	Enters address family configuration mode in order to exchange global table reachability.
Step 6	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor FE80::300::1 GigabitEthernet0/0/0 activate`	Enables the exchange of information for this address family with the specified BGP neighbor.

	Command or Action	Purpose
Step 7	**aggregate-address** *address mask* [**as-set**] [**summary-only**] [**suppress-map** *map-name*] [**advertise-map** *map-name*] [**attribute-map** *map-name*] **Example:** `Device(config-router-af)# aggregate-address 2001:DB8::/32 summary-only`	Creates an aggregate prefix before advertising it to the Internet.

Configuring the IPv6 VPN PE

Configuring a Default Static Route from the VRF to the Internet Gateway

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix/prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix/prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*] **Example:** `Device(config)# ipv6 route vrf vrf1 ::/0 2001:DB8:100::1 nexthop-vrf default`	Configures a default static route from the VRF to the Internet gateway to allow outbound traffic to leave the VRF.

Configuring a Static Route from the Default Table to the VRF

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix/prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix/prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*] **Example:** `Device(config)# ipv6 route 2001:DB8:100:2000::/64 nexthop-vrf vrf1`	Configures a static route from the default table to the VRF to allow inbound traffic to reach the VRF.

Configuring iBGP 6PE Peering to the Internet Gateway

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**]
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
9. **network** *ipv6-address/prefix-length*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.2.101 remote-as 100`	Adds an entry to the multiprotocol BGP neighbor table for peering with the Internet gateway.
Step 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 192.168.2.101 update-source Loopback 0`	Enables the BGP session to use a source address on the specified interface.
Step 6	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**] **Example:** `Device(config-router)# address-family ipv6`	Enters address family configuration mode to exchange global table reachability.
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.2.101 activate`	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **Example:** `Device(config-router-af)# neighbor 192.168.2.101 send-label`	Enables label exchange for this address family to this neighbor to enable the Virtual Private Network (VPN) provider edge (PE) to reach the Internet gateway over Multiprotocol Label Switching (MPLS).

	Command or Action	Purpose
Step 9	**network** *ipv6-address/prefix-length* **Example:** `Device(config-router-af)# network 2001:DB8:100:2000::/64`	Provides the virtual routing and forwarding (VRF) prefix to the Internet gateway.

Configuring a Multiautonomous-System Backbone for IPv6 VPN

Two Virtual Private Network (VPN) sites may be connected to different autonomous systems because the sites are connected to different service providers. The provider edge (PE) devices attached to that VPN is then unable to maintain the internal Border Gateway Protocol (iBGP) connections with each other or with a common route reflector. In this situation, there must be some way to use external BGP (eBGP) to distribute VPN-IPv6 addresses.

The following configuration example illustrates two scenarios, one in which a multiprotocol eBGP-IPv6 VPN peering between autonomous system boundary routers (ASBRs) uses an IPv4 link, and the same scenario using an IPv6 link. If the peering between ASBRs is performed over an IPv4 link, the BGP configuration on ASBR1 is as follows:

```
router bgp 1001
no bgp default ipv4-unicast
no bgp default route-target filter
neighbor 192.1.1.1 remote-as 1002
neighbor 192.168.2.11 remote-as 1001
neighbor 192.168.2.11 update-source Loopback1
!
 address-family vpnv6
!Peering to ASBR2 over an IPv4 link
 neighbor 192.1.1.1 activate
 neighbor 192.1.1.1 send-community extended
!Peering to PE1 over an IPv4 link
 neighbor 192.168.2.11 activate
 neighbor 192.168.2.11 next-hop-self
 neighbor 192.168.2.11 send-community extended
```

If the peering between ASBRs is performed over an IPv6 link, the BGP configuration on ASBR1 is as follows:

```
router bgp 1001
neighbor 2001:DB8:101::72d remote-as 1002
!
 address-family vpnv6
!Peering to ASBR2 over an IPv6 link
 neighbor 2001:DB8:101::72d activate
 neighbor 2001:DB8:101::72d send-community extended
```

The next several tasks describe how to configure the PE VPN for a multiautonomous-system backbone using multihop multiprotocol eBGP to redistribute VPN routes across route reflectors (RRs) in different autonomous systems. Labeled IPv4 routes to the PEs are advertised across ASBRs so that a complete label switch path (LSP) is set up end to end.

In this scenario, the ASBRs are not VPN aware; only the RRs are VPN aware. The following configuration should be available and understood:

- The ASBRs are providing the PEs' loopback addresses to service providers they peer with. That includes:

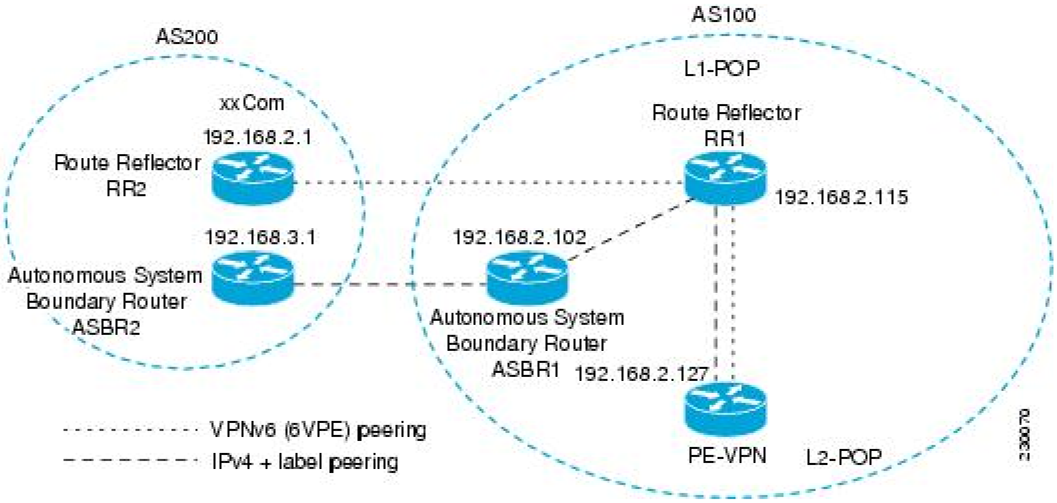
  - The VPN PE's IPv4 loopback address (/32) for enabling next-hop resolution at the remote service provider location.
  - The VPN RR's IPv4 loopback address (/32) for enabling interprovider (inter-RR) eBGP peering.

- For the VPN PE's IPv4 loopback address, the address providing is performed over multiprotocol BGP, with the label, up to the remote PEs, so that the label establishes an end-to-end LSP. Therefore, the following MP-BGP peering was set up for VPNv4:
  - VPN PEs are iBGP peering with VPN RRs.
  - ASBRs are iBGP peering with VPN RRs.
  - ASBRs are eBGP peering with the remote service provider ASBR.
- The VPN RRs of each service provider are peering together over eBGP and exchanging VPN routes. The next hop is forwarded unchanged, so that the end-to-end LSP is not via RRs.

To enable IPv6 VPN interautonomous-system access in this scenario, the ISP needs to modify the configurations at the PE VPN and at the RR. The same RRs are set up to provide a similar service for VPNv4. In that context, because the peering between the RR and the ASBR and between ASBRs is solely to exchange labels for IPv4 next hops used by both IPv4 VPN and IPv6 VPN, the ASBRs remain completely IPv6 unaware, and no configuration change is required there.

The figure below shows the BGP peering points required to enable IPv6 interprovider connectivity from the PE-VPN device (providing IPv6 VPN access) to the xxCom network.

Figure 9: BGP Peering Points for Enabling Interautonomous System Scenario C



The following additional BGP peerings are necessary to enable interautonomous-system communication from the IPv6 VPN PE located in the Level 2 point of presence (POP):

- IPv4 with label peering from the PE VPN to the route reflector named RR1 (which is already configured if VPNv4 interautonomous system is deployed on the same nodes, using the same LSP).
- IPv4 with label peering from RR1 to ASBR1.
- IPv4 with label peering between ASBR1 and ASBR2.
- IPv6 VPN peering between RR1 and RR2 (which is the route reflector in the other autonomous systems) to exchange IPv6 VPN routes.

- IPv6 VPN peering with RR1. If the same route reflectors used to scale the IPv6 VPN service are used for interautonomous-system capability, then this function might also be already configured (see the “Configuring Route Reflectors for Improved Scalability” section).

Configuring the multiautonomous-system backbone for IPv6 VPN consists of the following tasks:

Configuring the PE VPN for a Multiautonomous-System Backbone

Configuring iBGP IPv6 VPN Peering to a Route Reflector

Perform this task to configure internal Border Gateway Protocol (iBGP) IPv6 Virtual Private Network (VPN) peering to a route reflector named RR1.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family vpnv6** [**unicast**]
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the BGP routing process.
Step 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.2.115 remote-as 100`	Adds an entry to the multiprotocol BGP neighbor table for peering with the route reflector with interautonomous-system functionality.

	Command or Action	Purpose

Step 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
	Example:
	`Device(config-router)# neighbor 192.168.2.115`
	`update-source Loopback 0`
		Enables the BGP session to use a source address on the specified interface.

Step 6	**address-family vpnv6** [**unicast**] **Example:** `Device(config-router)# address-family vpnv6`	(Optional) Places the device in address family configuration mode for configuring routing sessions.

Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
	Example:
	`Device(config-router-af)# neighbor 192.168.2.115`
	`activate`
		Enables the exchange of information for this address family with the specified BGP neighbor.

Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
	Example:
	`Device(config-router-af)# neighbor 192.168.2.115`
	`send-community extended`
		Specifies that a communities attribute should be sent to the BGP neighbor.

Step 9	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode.

Configuring IPv4 and Label iBGP Peering to a Route Reflector

Perform this task to configure IPv4 and label internal Border Gateway Protocol (iBGP) peering to a route reflector named RR1.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the BGP routing process.
Step 4	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.2.115 activate`	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 6	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **Example:** `Device(config-router-af)# neighbor 192.168.2.115 send-label`	Enables label exchange for this address family to this neighbor in order to receive remote provider edge (PE) peer IPv4 loopback with label via RR1 in order to set up an end-to-end label switch path (LSP).

Configuring the Route Reflector for a Multiautonomous-System Backbone

Configuring Peering to the PE VPN

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*

6. **address-family vpnv6** [**unicast**]
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
9. **exit**
10. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
11. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
12. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
13. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.2.115 remote-as 100`	Adds an entry to the multiprotocol BGP neighbor table for peering with the route reflector for InterAS.
Step 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 192.168.2.115 update-source Loopback 0`	Enables the BGP session to use a source address on the specified interface.
Step 6	**address-family vpnv6** [**unicast**] **Example:** `Device(config-router)# address-family vpnv6`	(Optional) Places the device in address family configuration mode.

	Command or Action	Purpose
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.2.115 activate`	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **Example:** `Device(config-router-af)# neighbor 192.168.2.115 send-community extended`	Specifies that a community attribute should be sent to the BGP neighbor.
Step 9	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode.
Step 10	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 11	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.2.115 activate`	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 12	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **Example:** `Device(config-router-af)# neighbor 192.168.2.115 send-label`	Enables label exchange for this address family to this neighbor in order to send to the local provider edge (PE) the remote PE IPv4 loopback with a label in order to set up an end-to-end label switch path (LSP).
Step 13	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode.

Configuring the Route Reflector

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family vpnv6** [**unicast**]
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
10. **exit**
11. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
12. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
13. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
14. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.2.127 remote-as 100`	Adds an entry to the multiprotocol BGP neighbor table for peering with the Virtual Private Network (VPN) provider edge (PE) for Interns.
Step 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:**	Enables the BGP session to use a source address on the specified interface.

	Command or Action	Purpose

	`Device(config-router)# neighbor 192.168.2.127`
	`update-source Loopback 0`	

Step 6	**address-family vpnv6** [**unicast**] **Example:** `Device(config-router)# address-family vpnv6`	(Optional) Places the device in address family configuration mode.

Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
	Example:
	`Device(config-router-af)# neighbor 192.168.2.127`
	` activate`
		Enables the exchange of information for this address family with the specified neighbor.

Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
	Example:
	`Device(config-router-af)# neighbor 192.168.2.127`
	` send-community extended`
		Specifies that a community attribute should be sent to the BGP neighbor.

Step 9	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
	Example:
	`Device(config-router-af)# neighbor 192.168.2.127`
	` route-reflector-client`
		Configures the specified neighbor as a route reflector client.

Step 10	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode.
Step 11	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.

Step 12	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
	Example:
	`Device(config-router-af)# neighbor 192.168.2.127`
	` activate`
		Enables the exchange of information for this address family with the specified neighbor.

Step 13	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**	Enables label exchange for this address family to this neighbor in order to send to the local PE the remote PE

	Command or Action	Purpose
	Example: `Device(config-router-af)# neighbor 192.168.2.127 send-label`	IPv4 loopback with a label in order to set up an end-to-end LSP.
Step 14	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode.

Configuring Peering to the Autonomous System Boundary Router

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.2.102 remote-as 100`	Adds an entry to the multiprotocol BGP neighbor table for peering with the ASBR1.

	Command or Action	Purpose
Step 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 192.168.2.102 update-source Loopback 0`	Enables the BGP session to use a source address on the specified interface.
Step 6	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.2.102 activate`	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **Example:** `Device(config-router-af)# neighbor 192.168.2.102 send-label`	Enables label exchange for this address family to this neighbor in order to receive the remote PE IPv4 loopback with the label set to an end-to-end label switch path (LSP).

Configuring Peering to Another ISP Route Reflector

Perform this task to configure peering to an Internet service provider (ISP) route reflector named RR2.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]
7. **address-family vpnv6** [**unicast**]
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
10. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop-unchanged** [**allpaths**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.2.1 remote-as 100`	Adds an entry to the multiprotocol BGP neighbor table for external BGP (eBGP) peering with RR2.
Step 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 192.168.2.1 update-source Loopback 0`	Enables the BGP session to use a source address on the specified interface.
Step 6	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*] **Example:** `Device(config-router)# neighbor 192.168.2.1 ebgp-multihop`	(Optional) Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.
Step 7	**address-family vpnv6** [**unicast**] **Example:** `Device(config-router)# address-family vpnv6`	(Optional) Places the device in address family configuration mode for configuring routing sessions.
Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.2.1 activate`	Enables the exchange of information for this address family with the specified BGP neighbor.

	Command or Action	Purpose
Step 9	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**] **Example:** `Device(config-router-af)# neighbor 192.168.2.1 send-community extended`	Specifies that a communities attribute should be sent to the BGP neighbor.
Step 10	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop-unchanged** [**allpaths**] **Example:** `Device(config-router-af)# neighbor 192.168.2.1 next-hop-unchanged allpaths`	Enables an eBGP multihop peer to propagate to the next hop unchanged for paths.

Configuring the ASBR

Configuring Peering with Router Reflector RR1

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
9. exit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:**	Configures the Border Gateway Protocol (BGP) routing process.

	Command or Action	Purpose
	`Device(config)# router bgp 100`	
Step 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.2.115 remote-as 100`	Adds an entry to the multiprotocol BGP neighbor table for peering with RR1.
Step 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 192.168.2.115 update-source Loopback 0`	Enables the BGP session to use a source address on the specified interface.
Step 6	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.2.115 activate`	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **Example:** `Device(config-router-af)# neighbor 192.168.2.115 send-label`	Enables label exchange for this address family to this neighbor in order to send to the local PE the remote PE IPv4 loopback with a label in order to set up an end-to-end label switch path (LSP).
Step 9	exit **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode.

Configuring Peering with the Other ISP ASBR2

SUMMARY STEPS

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

4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]
7. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
10. **network** {*network-number* [**mask** *network-mask*] | *nsap-prefix*} [**route-map** *map-tag*]
11. **network** {*network-number* [**mask** *network-mask*] | *nsap-prefix*} [**route-map** *map-tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the Border Gateway Protocol (BGP) routing process.
Step 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.3.1 remote-as 100`	Adds an entry to the multiprotocol BGP neighbor table for peering with the ASBR2.
Step 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Device(config-router)# neighbor 192.168.3.1 update-source Loopback 0`	Enables the BGP session to use a source address on the specified interface.
Step 6	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*] **Example:** `Device(config-router)# neighbor 192.168.3.1 ebgp-multihop`	Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.

	Command or Action	Purpose
Step 7	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.3.1 activate`	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 9	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **Example:** `Device(config-router-af)# neighbor 192.168.3.1 send-label`	Enables label exchange for this address family to this neighbor in order to receive the remote provider edge (PE) IPv4 loopback with a label in order to set up an end-to-end label switch path (LSP).
Step 10	**network** {*network-number* [**mask** *network-mask*] | *nsap-prefix*} [**route-map** *map-tag*] **Example:** `Device(config-router-af)# network 192.168.2.27 mask 255.255.255.255`	Flags a network as local to this autonomous system and enters the network to the BGP table. This configuration is for the PE VPN loopback.
Step 11	**network** {*network-number* [**mask** *network-mask*] | *nsap-prefix*} [**route-map** *map-tag*] **Example:** `Device(config-router-af)# network 192.168.2.15 mask 255.255.255.255`	Flags a network as local to this autonomous system and enters the network to the BGP table. This configuration is for the RR1 loopback.

Configuring CSC for IPv6 VPN

Perform this task to configure CsC-PE1 peering configuration with CsC-CE1.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *name*
4. **router bgp** *autonomous-system-number*
5. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**]
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**

8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**hostname** *name* **Example:** `Device(config)# hostname CSC-PE1`	Specifies or modifies the host name for the network server.
Step 4	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the Border Gateway Protocol (BGP) routing process.
Step 5	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**] **Example:** `Device(config-router)# address-family ipv6 vrf ISP2`	Enters address family configuration mode.

Step 6	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
	Example:
	`Device(config-router-af)# neighbor FE80::866C:99`
	`GigabitEthernet0/0/0 remote-as 200`
		Adds an entry to the multiprotocol BGP neighbor table.

Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **activate**
	Example:
	`Device(config-router-af)# neighbor FE80::866C:99`
	`GigabitEthernet0/0/0 activate`
		Enables the exchange of information for this address family with the specified BGP neighbor.

Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label** **Example:**	Enables label exchange for this address family to this neighbor.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor FE80::866C:99 GigabitEthernet0/0/0 send-label`	

Configuration Examples for IPv6 VPN over MPLS

Examples: IPv6 VPN over MPLS Routing

Example: BGP IPv6 Activity Summary

```
Device# show bgp ipv6 unicast summary

For address family: IPv6 Unicast
BGP router identifier 192.168.2.126, local AS number 33751
BGP table version is 15, main routing table version 15
12 network entries using 1692 bytes of memory
22 path entries using 1672 bytes of memory
5/4 BGP path/bestpath attribute entries using 580 bytes of memory
14 BGP rrinfo entries using 336 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 4328 total bytes of memory
Dampening enabled. 0 history paths, 0 dampened paths
BGP activity 13/1 prefixes, 23/1 paths, scan interval 60 secs
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.2.146     4 33751     991     983       15    0    0 16:26:21       10
192.168.2.147     4 33751     991     983       15    0    0 16:26:22       10
FE80::4F6B:44 GigabitEthernet1/0/0
                4 20331     982     987       15    0    0 14:55:52        1
```

Example: Dumping the BGP IPv6 Tables

Each table (for example, BGP IPv6, BGP IPv6 VPN) can be reviewed individually, as shown in the following example:

```
Device# show bgp ipv6 unicast
BGP table version is 15, local router ID is 192.168.2.126
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric    LocPrf Weight Path
* i2001:DB8:100::/48 ::FFFF:192.168.2.101     0    100      0 10000 ?
*>i                  ::FFFF:192.168.2.101     0    100      0 10000 ?
* i2001:DB8::1/128   ::FFFF:192.168.2.101     0    100      0  i
*>i                  ::FFFF:192.168.2.101     0    100      0  i
```

Example: Dumping the IPv6 Routing Tables

IPv6 routing tables identify each routing protocol contributor to routable entries, as shown in the following example:

```
Device# show ipv6 route
IPv6 Routing Table - default - 13 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
       IA - ISIS interarea, IS - ISIS summary
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
B   2001:DB8:100::/48 [200/0]
     via 192.168.2.101 Default-IP-Routing-Table, indirectly connected
B   2001:DB8::1/128 [200/0]
     via 192.168.2.101 Default-IP-Routing-Table, c
LC  2001:DB8::26/128 [0/0]
     via Loopback0, receive
```

From an IPv6 routing perspective, entries reachable over the MPLS backbone are listed as being indirectly connected, because MPLS is providing a Layer 2 tunnel mechanism.

Examples: IPv6 VPN over MPLS Forwarding

Example: PE-CE Connectivity

The **ipv6 ping** and **traceroute** commands are useful to check connectivity from a provider edge (PE) to a customer edge (CE), whether locally attached or remote over the Multiprotocol Label Switching (MPLS) backbone.

When a device is locally attached, one can use the **ipv6 ping** command with the CE link-local address (used for external BGP peering), as shown in the following example:

```
Device# ping FE80::4F6B:44%
Loopback0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::4F6B:44, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/33/48 ms
```

The **ipv6 ping** command also can be used to test remote PE or CE reachability, but only IPv6 global addresses can be used (link-local addresses are not advertised beyond the link):

```
Device# ping 2001:DB8:1120:1::44
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:1120:1:44::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/33/48 ms
```

Note that the **ping ipv6** and **traceroute** command functions over MPLS require PEs and CEs to announce one IPv6 global prefix. Each 6PE device announces 2001:DB8::PE#/128, filtered at the autonomous system edge. Each IPv6 CE configures 2001:DB8:*prefix*:CE#/128 and announces it as part as its less-specific prefix (2001:DB8:*prefix*::/n).

Reachability of remote PEs and CEs can be tested by using the **traceroute** command. If you have configured all PEs with the **no mpls ip propagate-ttl forwarded** command, when the **traceroute** command is executed from a CE, its output will show only the IPv6 nodes:

```
Device# traceroute 2001:DB8::1
Type escape sequence to abort.
Tracing the route to 2001:DB8::1
  1 2001:DB8::26 [AS 33751] 32 msec 32 msec 20 msec
  2 2001:DB8::1 [AS 33751] [MPLS: Label 73 Exp 0] 20 msec 20 msec 20 msec
  3 2001:DB8::1 [AS 33751] 28 msec 20 msec 20 msec
```

After the P devices have been upgraded with images that support ICMPv6, the **traceroute** command executed on the PE device (Time to Live [TTL] is then propagated) will also show P devices' responses, as shown in the following example:

```
Device# traceroute 2001:DB8::1

Type escape sequence to abort.
Tracing the route to 2001:DB8::1
  1 ::FFFF:172.20.25.1 [MPLS: Labels 38/73 Exp 0] 40 msec 32 msec 32 msec
  2 ::FFFF:172.20.10.1 [MPLS: Labels 30/73 Exp 0] 60 msec 32 msec 32 msec
  3 2001:DB8::1 [MPLS: Label 73 Exp 0] 32 msec 32 msec 16 msec
```

When run from a 6VPE device, both the **ping ipv6** and **traceroute** commands accept a *vrf* argument, exactly as in the case of VPNv4.

Note that the **traceroute** command is useful for evaluating the path across the MPLS backbone, but not for troubleshooting data-plane failures. The P devices are IPv6 unaware (and are also VPNv4 unaware), so the ICMPv6 messages that they generate in response to the **traceroute** command are forwarded to the egress PE using the received label stack. The egress PE can route the ICMPv6 message to the source of the traceroute. When the MPLS path is broken, it is also broken from the ICMP message, which cannot reach the egress PE.

Examples: PE Imposition Path

On Cisco devices, the most useful tool for troubleshooting the imposition path for IPv6 is the **show ipv6 cef** command.

You can use the **show ipv6 cef** command to display the IPv6 forwarding table with label stacks used for each destination prefix, as shown in the following example:

```
Device# show ipv6 cef

2001:DB8:100::/48
  nexthop 172.20.25.1 GigabitEtherent0/0/0 label 38 72
2001:DB8::1/128
  nexthop 172.20.25.1 GigabitEtherent0/0/0 label 38 73
2001:DB8::26/128
  attached to Loopback0, receive
```

You can use the **show ipv6 cef** command to display details for a specific IPv6 entry in the forwarding table and to analyze how the destination was resolved and the label stack computed, as shown in the following example:

```
Device# show ipv6 cef 2001:DB8:100::/48 internal

2001:DB8:100::/48, epoch 0, RIB[B], refcount 4
  sources: RIB
..
  recursive via 192.168.2.101[IPv4:Default] label 72, fib 0252B1F8, 1 terminal fib
    path 024F56A8, path list 024F0BA8, share 0/1, type attached nexthop
    ifnums: (none)
     path_list contains at least one resolved destination(s). HW IPv4 notified.
    nexthop 172.20.25.1 GigabitEtherent0/0/0 label 38, adjacency IP adj out of
GigabitEtherent0/0/0 0289BEF0
  output chain: label 72 label 38 TAG adj out of GigabitEtherent0/0/0 0289BD80
```

The detailed output in the previous example shows that each label composing the label stack has a different origin that can be tracked down individually. The Border Gateway Protocol (BGP) table has the bottom label, as shown in the following example:

```
Device# show bgp ipv6 unicast 2001:DB8:100::/48

BGP routing table entry for 2001:DB8:100::/48, version 2
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
     1
  10000
    ::FFFF:192.168.2.101 (metric 30) from 192.168.2.147 (192.168.2.147)
      Origin incomplete, metric 0, localpref 100, valid, internal
      Originator: 192.168.2.101, Cluster list: 192.168.2.147,
      mpls labels in/out nolabel/72
  10000
    ::FFFF:192.168.2.101 (metric 30) from 192.168.2.146 (192.168.2.146)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Originator: 192.168.2.101, Cluster list: 192.168.2.146,
      mpls labels in/out nolabel/72
```

Label Distribution Protocol (LDP), as shown in this example, displays the other labels:

```
Device# show mpls ldp bindings 192.168.2.101 32

  lib entry: 192.168.2.101/32, rev 56
        local binding:  label: 40
        remote binding: lsr: 192.168.2.119:0, label: 38
Device# show mpls ldp bindings 172.20.25.0 24
  lib entry: 172.20.25.0/24, rev 2
        local binding: label: imp-null
        remote binding: lsr: 192.168.2.119:0, label: imp-null
```

Examples: PE Disposition Path

Use the following examples to troubleshoot the disposition path.

The following example shows the Multiprotocol Label Switching (MPLS) forwarding table information for troubleshooting the disposition path.

```
Device# show mpls forwarding-table

Local  Outgoing      Prefix            Bytes Label   Outgoing   Next Hop
Label  Label or VC   or Tunnel Id      Switched      interface
16     Pop Label     192.168.2.114/32    0             GE0/0/0   point2point
17     26            192.168.2.146/32    0             GE0/0/0   point2point
..
72     No Label      2001:DB8:100::/48   63121         GE1/0/0   point2point
73     Aggregate     2001:DB8::1/128   24123
```

The following example shows the label used for switching, which has been announced by iBGP (6PE in this example) and can be checked:

```
Device# show bgp ipv6 2001:DB8:100::/48

BGP routing table entry for 2001:DB8:100::/48, version 2
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     2
  10000
```

```
    FE80::2710:2 (FE80::2710:2) from FE80::2710:2 GigabitEthernet1/0/0 (192.168.2.103)
      Origin incomplete, metric 0, localpref 100, valid, external, best,
```

Examples: Label Switch Path

Because the 6PE and 6VPE label switch path (LSP) endpoints are IPv4 addresses, the IPv4 tools for troubleshooting LSPs are useful for detecting data-plane failures that would lead to IPv6 traffic black-holing.

The following example displays the LSP IPv4 end to analyze the LSP:

```
Device# show ipv6 route 2001:DB8::1/128

Routing entry for 2001:DB8::1/128
  Known via "bgp 33751", distance 200, metric 0, type internal
  Route count is 1/1, share count 0
  Routing paths:
    192.168.2.101%Default-IP-Routing-Table indirectly connected
      MPLS Required
      Last updated 02:42:12 ago
```

The following example shows the traceroute LSP:

```
Device# traceroute mpls ipv4 192.168.2.101/32 verbose

Tracing MPLS Label Switched Path to 192.168.2.101/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target,
       'M' - malformed request
Type escape sequence to abort.
  0 172.20.25.2 0.0.0.0 MRU 1500 [Labels: 38 Exp: 0]
R 1 172.20.25.1 0.0.0.0 MRU 1500 [Labels: 30 Exp: 0] 40 ms, ret code 6
R 2 172.20.10.1 0.0.0.0 MRU 1504 [Labels: implicit-null Exp: 0] 60 ms, ret code 6
! 3 172.20.40.1 48 ms
```

Examples: IPv6 VPN over MPLS VRF

Examples: VRF Information

The following entries show VRF information for 6VPE.

The following is sample output from a Cisco Express Forwarding FIB associated with a virtual routing and forwarding (VRF) instance named cisco1:

```
Device# show ipv6 cef vrf cisco1

 2001:8::/64
  attached to GigabitEthernet0/0/1
 2001:8::3/128
  receive
 2002:8::/64
  nexthop 10.1.1.2 GigabitEthernet0/1/0 label 22 19
 2010::/64
  nexthop 2001:8::1 GigabitEthernet0/0/1
 2012::/64
  attached to Loopback1
 2012::1/128
  receive
```

The following is sample output regarding an IPv6 routing table associated with a VRF named cisco1:

```
Device# show ipv6 route vrf cisco1

IPv6 Routing Table cisco1 - 6 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
C   2001:8::/64 [0/0]
     via ::, GigabitEthernet0/0/1
L   2001:8::3/128 [0/0]
     via ::, GigabitEthernet0/0/1
B   2002:8::/64 [200/0]
     via ::FFFF:192.168.1.4,
B   2010::/64 [20/1]
     via 2001:8::1,
C   2012::/64 [0/0]
     via ::, Loopback1
L   2012::1/128 [0/0]
     via ::, Loopback1
```

Example: IPv6 VPN Configuration Using IPv4 Next Hop

The following example illustrates a 6VPE next hop:

```
interface Loopback0
 ip address 192.168.2.11 255.255.255.255
!
router bgp 100
 neighbor 192.168.2.10 remote-as 100
 neighbor 192.168.2.10 update-source Loopback0
!
 address-family vpnv6
 neighbor 192.168.2.10 activate
 neighbor 192.168.2.10 send-community extended
 exit-address-family
```

By default, the next hop advertised will be the IPv6 Virtual Private Network (VPN) address:

```
[0:0]::FFFF:192.168.2.10
```

Note that it is a 192-bit address in the format of [RD]::FFFF:IPv4-address.

When the Border Gateway Protocol (BGP) IPv6 VPN peers share a common subnet, the MP_REACH_NLRI attribute contains a link-local address next hop in addition to the global address next hop. This situation typically occurs in an interautonomous-system topology when autonomous system boundary routers (ASBRs) are facing each other. In that case, the link-local next hop is used locally, and the global next hop is readvertised by BGP.

The BGP next hop is the keystone for building the label stack. The inner label is obtained from the BGP network layer reachability information (NLRI), and the outer label is the Label Distribution Protocol (LDP) label to reach the IPv4 address embedded into the BGP next hop.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco Master Command List, All Releases
IPv6 addressing and connectivity	*IPv6 Configuration Guide Library*
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	IPv6 Feature Mapping
Configuring MPLS Layer 3 VPNs	"MPLS Virtual Private Networks" module in the *MPLS Layer 3 VPNs Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	IPv6 RFCs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Glossary

- **6VPE device**—Provider edge device providing BGP-MPLS IPv6 VPN service over an IPv4-based MPLS core. It is a IPv6 VPN PE, dual-stack device that implements 6PE concepts on the core-facing interfaces.
- **customer edge (CE) device**—A service provider device that connects to VPN customer sites.

- **Forwarding Information Base (FIB)**—Table containing the information necessary to forward IP datagrams. At a minimum, the FIB contains the interface identifier and next-hop information for each reachable destination network prefix.
- **inbound route filtering (IRF)**—A BGP capability used for filtering incoming BGP updates that are not to be imported by the receiving PE device.
- **IPv6 provider edge device (6PE device)**—Device running a BGP-based mechanism to interconnect IPv6 islands over an MPLS-enabled IPv4 cloud.
- **IPv6 VPN address**—A IPv6 VPN address is a 24-byte identifier, beginning with an 8-byte route distinguisher (RD) and ending with a 16-byte IPv6 address. Sometimes it is called an IPv6 VPN address.
- **IPv6 VPN address family**—The address-family identifier (AFI) identifies a particular network-layer protocol and the subsequent AFI (SAFI) provides additional information. The AFI IPv6 SAFI VPN (AFI=2, SAFI=128) is called the IPv6 VPN address family. Sometimes it is called the IPv6 VPN address family. Similarly AFI IPv4 SAFI VPN is the VPNv4 address family.
- **network layer reachability information (NLRI)**—BGP sends routing update messages containing NLRI to describe a route and how to get there. In this context, an NLRI is a prefix. A BGP update message carries one or more NLRI prefixes and the attributes of a route for the NLRI prefixes; the route attributes include a BGP next hop gateway address and community values.
- **outbound route filtering (ORF)**—A BGP capability used to filtering outgoing BGP routing updates.
- **point of presence (POP)**—Physical location where an interexchange carrier installed equipment to interconnect with a local exchange carrier.
- **provider edge (PE) device**—A service provider device connected to VPN customer sites.
- **route distinguisher (RD)**—A 64-bit value prepended to an IPv6 prefix to create a globally unique IPv6 VPN address.
- **Routing Information Base (RIB)**—Also called the routing table.
- **Virtual routing and forwarding (VRF)**—A VPN routing and forwarding instance in a PE.
- **VRF table**—A routing and a forwarding table associated to a VRF. This is a customer-specific table that enables the PE device to maintain independent routing states for each customer.

CHAPTER 6

Configuring Route Maps to Control the Distribution of MPLS Labels Between Routers in an MPLS VPN

Route maps enable you to specify which routes are distributed with Multiprotocol Label Switching (MPLS) labels. Route maps also enable you to specify which routes with MPLS labels a router receives and adds to its Border Gateway Protocol (BGP) table.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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.

Restrictions for Using Route Maps with MPLS VPNs

You can use route maps with MPLS VPN Inter-AS with Autonomous System Boundary Routers (ASBRs) exchanging IPv4 routes with MPLS labels. You cannot use route maps with MPLS VPN Inter-AS with ASBRs exchanging VPN-IPv4 addresses.

Prerequisites for Using Route Maps with MPLS VPNs

Before you configure and apply route maps, you need to create an access control list (ACL) and specify the routes that the router should distribute with MPLS labels.

Information About Route Maps in MPLS VPNs

When routers are configured to distribute routes with MPLS labels, all the routes are encoded with the multiprotocol extensions and contain MPLS labels. You can use a route map to control the distribution of MPLS labels between routers.

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 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.

Route maps work with ACLs. You enter the routes into an ACL and then specify the ACL when you configure the route map. You can configure a router to accept only routes that are specified in the route map. The router checks the routes listed in the BGP update message against the list of routes in the specified ACL. If a route in the BGP update message matches a route in the ACL, the route is accepted and added to the BGP table.

How to Configure Route Maps in an MPLS VPN

Perform the following tasks to enable routers to send MPLS labels with the routes specified in the route maps:

Configuring a Route Map for Incoming Routes

Perform this task to create a route map to filter arriving routes. You create an ACL 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** *map-name* [**permit** | **deny**] *sequence-number*
5. **match ip address** {*access-list-number* [*access-list-number*...| *access-list-name*...] *access-list-name* [*access-list-number*...| *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name*....]]
6. **match mpls-label**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**route-map** *map-name* [**permit** | **deny**] *sequence-number* **Example:** `Router(config-router)# route-map csc-mpls-routes-in permit`	Enters route map configuration mode and creates a route map with the name you specify. • The *map-name* argument identifies the name of the route map. • The **permit** keyword allows the actions to happen if all conditions are met. • A **deny** keyword prevents any actions from happening if all conditions are met. • 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-number...*| *access-list-name...*] *access-list-name* [*access-list-number...*| *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name....*]] **Example:** `Router(config-route-map)# match ip address acl-in`	Distributes any routes that have a destination network number address that is permitted by a standard access list, an extended access list, or a prefix list, or performs policy routing on packets. • The *access-list-number...* argument is a number of a standard or extended access list. It can be an integer from 1 to 199. The ellipsis indicates that multiple values can be entered.

	Command or Action	Purpose
		• The *access-list-name*... argument is a name of a standard or extended access list. It can be an integer from 1 to 199. The ellipsis indicates that multiple values can be entered. • The **prefix-list** keyword distributes routes based on a prefix list. • The *prefix-list-name*... argument is a name of a specific prefix list. The ellipsis indicates that multiple values can be entered.
Step 6	**match mpls-label** **Example:** `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	**exit** **Example:** `Router(config-route-map)# exit`	Exits route map configuration mode and returns to global configuration mode.

Configuring a Route Map for Outgoing 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** *map-name* [**permit** | **deny**] *sequence-number*
5. **match ip address** {*access-list-number* [*access-list-number*...| *access-list-name*...}] | *access-list-name* [*access-list-number*...| *access-list-name* | **prefix-list** *prefix-list-name* [*prefix-list-name*....]]
6. **set mpls-label**
7. **exit**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**route-map** *map-name* [**permit** | **deny**] *sequence-number* **Example:** `Router(config-router)# route-map csc-mpls-routes-out permit`	Enters route map configuration mode and creates a route map with the name you specify. • The *map-name* argument identifies the name of the route map. • The **permit** keyword allows the actions to happen if all conditions are met. • A **deny** keyword prevents any actions from happening if all conditions are met. • 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-number...*| *access-list-name...*}] | *access-list-name* [*access-list-number...*| *access-list-name* | **prefix-list** *prefix-list-name* [*prefix-list-name....*]] **Example:** `Router(config-route-map)# match ip address acl-out`	Distributes any routes that have a destination network number address that is permitted by a standard access list, an extended access list, or a prefix list, or performs policy routing on packets. • The *access-list-number...* argument is a number of a standard or extended access list. It can be an integer from 1 to 199. The ellipsis indicates that multiple values can be entered. • The *access-list-name...* argument is a name of a standard or extended access list. It can be an integer from 1 to 199. The ellipsis indicates that multiple values can be entered.

	Command or Action	Purpose
		• The **prefix-list** keyword distributes routes based on a prefix list. • The *prefix-list-name*... argument is a name of a specific prefix list. The ellipsis indicates that multiple values can be entered.
Step 6	**set mpls-label** **Example:** `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	**exit** **Example:** `Router(config-route-map)# exit`	Exits route map configuration mode and returns to global configuration mode.

Applying the Route Maps to the MPLS VPN Edge Routers

This configuration is optional.

Perform this task to enable the edge routers 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** *map-name* **in**
6. **neighbor** *ip-address* **route-map** *map-name* **out**
7. **neighbor** *ip-address* **send-label**
8. **exit-address-family**
9. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 vrf vpn1`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**neighbor** *ip-address* **route-map** *map-name* **in** **Example:** `Router(config-router-af)# neighbor pp.0.0.1 route-map csc-mpls-routes-in in`	Applies a route map to incoming routes. • The *ip-address* argument specifies the router to which the route map is to be applied. • The *map-name* argument specifies the name of the route map. • The **in** keyword applies the route map to incoming routes.
Step 6	**neighbor** *ip-address* **route-map** *map-name* **out** **Example:** `Router(config-router-af)# neighbor pp.0.0.1 route-map csc-mpls-route-out out`	Applies a route map to outgoing routes. • The *ip-address* argument specifies the router to which the route map is to be applied. • The *map-name* argument specifies the name of the route map. • The **out** keyword applies the route map to outgoing routes.
Step 7	**neighbor** *ip-address* **send-label** **Example:** `Router(config-router-af)# neighbor pp.0.0.1 send-label`	Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighboring router.

	Command or Action	Purpose
Step 8	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits from address family configuration mode.

Step 9	**end**
	Example:
	`Router(config-router)# end`
		(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

You can enter a **show route-map** *map-name* command to verify that the route map is applied to the PE routers.

Note After you make any changes to a route map, you need to reset the BGP connection for the changes to take effect.

Configuration Examples for Route Maps in MPLS VPNs

Using a Route Map in an MPLS VPN Inter-AS Network Example

In this example, a route map is applied to an autonomous system border router (ASBR) that exchanges IPv4 routes and MPLS labels with another ASBR.

- A route map called OUT specifies that the ASBR should distribute the PE1 route (ee.ee) with labels and the RR1 route (aa.aa) without labels.
- A route map called IN specifies that the ASBR 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
 tag-switching 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
```

Using a Route Map in an MPLS VPN CSC Network Example

The following example creates two route maps, which are named:

- IN for incoming routes
- OUT for outgoing routes

The route maps specify the following:

- If an IP address in an incoming BGP update message matches an IP address in access list 99, the route is added to the BGP table.
- If an IP address in an outbound BGP update message matches an IP address in access list 88, the router distributes that route.

The route maps are applied to the CSC-PE router with the address qq.0.0.1.

```
address-family ipv4 vrf vpn2
 neighbor qq.0.0.1 remote-as 200
 neighbor qq.0.0.1 activate
 neighbor qq.0.0.1 as-override
 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
!
access-list 88 permit rr.rr.rr.rr
access-list 88 permit ss.ss.ss.ss
access-list 88 permit tt.tt.tt.tt
access-list 99 permit uu.uu.uu.uu
access-list 99 permit vv.vv.vv.vv
access-list 99 permit ww.ww.ww.ww
!
route-map IN permit 1
 match ip address 99
!
route-map OUT permit 1
 match ip address 88
 set mpls-label
!
```

Additional References

Related Documents

Related Topic	Document Title
Basic MPLS VPNs	Configuring MPLS Layer 3 VPNs
MPLS VPN Carrier Supporting Carrier	• MPLS VPN Carrier Supporting Carrier Using LDP and an IGP • MPLS VPN Carrier Supporting Carrier with BGP

Related Topic	Document Title
MPLS VPN InterAutonomous Systems	• MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels • MPLS VPN Inter-AS with ASBRs Exchanging VPN-IPv4 Addresses

Standards

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

MIBs

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

RFCs

RFC	Title
RFC 2547	BGP/MPLS VPNs

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of 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. The Cisco Technical Support website contains thousands of 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. 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 Route Maps in MPLS VPNs

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

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

Table 5: Feature Information for Route Maps in MPLS VPNs

Feature Name	Releases	Feature Configuration Information
This feature was included as part of the following features: • MPLS VPN Inter-Autonomous Systems - IPv4 BGP Label Distribution • MPLS VPN Carrier Supporting Carrier with IPv4 BGP Label Distribution	12.0(21)ST 12.0(22)S 12.0(23)S 12.2(13)T 12.0(24)S 12.2(14)S 12.0(27)S 12.0(29)S	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.

CHAPTER 7

Assigning an ID Number to an MPLS VPN

You can identify Virtual Private Networks (VPNs) by a VPN identification number, as described in RFC 2685. This implementation of the MPLS VPN ID feature is used for identifying a VPN.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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.

Restrictions for MPLS VPN ID

The MPLS VPN ID feature is not used to control the distribution of routing information or to associate IP addresses with MPLS VPN ID numbers in the Multiprotocol Border Gateway Protocol (MP-BGP) VPNv4 routing updates.

Information About MPLS VPN ID

Introduction to MPLS VPN ID

You can identify Virtual Private Networks (VPNs) by a VPN identification number, as described in RFC 2685. This implementation of the VPN ID feature is used for identifying a VPN. The VPN ID feature is not used to control the distribution of routing information or to associate IP addresses with VPN ID numbers in the Multiprotocol Border Gateway Protocol (MP-BGP) VPNv4 routing updates.

Multiple VPNs can be configured in a device. A VPN is private and uses a private address space that might also be used by another VPN or by the Internet. The IP address used in a VPN is only significant to the VPN in which it exists. You can use a VPN name (a unique ASCII string) to reference a specific VPN configured in the device. Alternately, you can use a VPN ID to identify a particular VPN in the device. The VPN ID follows a standard specification (RFC 2685). To ensure that the VPN has a consistent VPN ID, assign the same VPN ID to all the devices in the service provider network that services that VPN.

Note Configuration of a VPN ID for a VPN is optional. You can still use a VPN name to identify configured VPNs in the device. The VPN name is not affected by the VPN ID configuration. These are two independent mechanisms to identify VPNs.

Components of the MPLS VPN ID

Each MPLS VPN ID defined by RFC 2685 consists of the following elements:

- An Organizational Unique Identifier (OUI), a three-octet hex number: The IEEE Registration Authority assigns OUIs to any company that manufactures components under the ISO/IEC 8802 standard. The OUI is used to generate universal LAN MAC addresses and protocol identifiers for use in local and metropolitan area network applications. For example, an OUI for Cisco Systems is 00-03-6B (hex).
- A Virtual Private Network (VPN) index: a four-octet hex number, which identifies the VPN within the company.

Use the following **vpn id** command and specify the VPN ID:

vpn id *oui*:*vpn-index*

A colon separates the OUI from the VPN index.

Management Applications That Use MPLS VPN IDs

You can use several applications to manage Virtual Private Networks (VPNs) by MPLS VPN ID. Remote access applications, such as the Remote Authentication Dial-In User Service (RADIUS) and Dynamic Host Configuration Protocol (DHCP), can use the MPLS VPN ID feature to identify a VPN. RADIUS can use the MPLS VPN ID to assign dial-in users to the proper VPN, based on each user’s authentication information.

Dynamic Host Configuration Protocol

Using Dynamic Host Configuration Protocol (DHCP) network administrators can centrally manage and automate the assignment of IP addresses in an organization's network. The DHCP application uses the MPLS VPN ID as follows:

1. A Virtual Private Network (VPN) DHCP client requests a connection to a provider edge (PE) device from a virtual routing and forwarding (VRF) interface.
2. The PE device determines the VPN ID associated with that interface.
3. The PE device sends a request with the VPN ID and other information for assigning an IP address to the DHCP server.
4. The DHCP server uses the VPN ID and IP address information to process the request.
5. The DHCP server sends a response back to the PE device, allowing the VPN DHCP client access to the VPN.

Remote Authentication Dial-In User Service

A RADIUS server (or daemon) provides authentication and accounting services to one or more client network access servers (NASs). RADIUS servers authenticate users and return all configuration information necessary for the client to deliver service to the users.

Typically, a user login consists of a query (Access-Request) from the NAS to the RADIUS server and a corresponding response (Access-Accept or Access-Reject) from the server.

- The Access-Request packet contains the username, encrypted password, NAS IP address, MPLS VPN ID, and port. The format of the request also provides information on the type of session that the user wants to initiate.
- The RADIUS server returns an Access-Accept response if it finds the username and verifies the password. The response includes a list of attribute-value pairs that describe the parameters to be used for this session. If the user is not authenticated, an Access-Reject is sent by the RADIUS server and access is denied.

How to Configure an MPLS VPN ID

Specifying an MPLS VPN ID

Before you begin

Each virtual routing and forwarding (VRF) instance configured on a provider edge (PE) device can have an MPLS VPN ID configured. Configure all the PE devices that belong to the same Virtual Private Network (VPN) with the same VPN ID. Make sure the VPN ID is unique to the service provider network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*

4. **vpn id** *oui*:*vpn-index* :

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf vrf1`	Creates a VRF routing table and a Cisco Express Forwarding forwarding table and enters VRF configuration mode. • *vrf-name*—Name assigned to a VRF.
Step 4	**vpn id** *oui*:*vpn-index* : **Example:** `Device(config-vrf)# vpn id a1:3f6c`	Assigns the VPN ID to the VRF. • *oui* :—An organizationally unique identifier. The IEEE organization assigns this identifier to companies. The OUI is restricted to three octets. • *vpn-index*—This value identifies the VPN within the company. This VPN index is restricted to four octets.

Verifying the MPLS VPN ID Configuration

SUMMARY STEPS

1. **enable**
2. **show ip vrf**
3. **show ip vrf id**
4. **show ip vrf detail**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

Example:

```
Device> enable
Device#
```

Step 2 **show ip vrf**

Displays information about the virtual routing and forwarding (VRF) tables on the provider edge (PE) device. This example displays three VRF tables called vpn1, vpn2, and vpn5.

Example:

```
Device# show ip vrf

  Name                              Default RD          Interfaces
  vpn1                              100:1               FastEthernet1/1/1
                                                        FastEthernet1/0/0
  vpn2                              <not set>
  vpn5                              500:1               Loopback2
```

Step 3 **show ip vrf id**

Ensures that the PE device contains the MPLS VPN ID you specified. The following example shows that only VRF tables vpn1 and vpn2 have VPN IDs assigned. The VRF table called vpn5 is not displayed, because it does not have a VPN ID.

Example:

```
Device# show ip vrf id

VPN Id          Name                              RD
2:3             vpn2                              <not set>
A1:3F6C         vpn1                              100:1
```

Step 4 **show ip vrf detail**

Displays all the VRFs on a PE device. This command displays all the MPLS VPN IDs that are configured on the device, their associated VRF names, and VRF route distinguishers (RDs). If a VRF table in the PE device has not been assigned an MPLS VPN ID, that VRF entry is not included in the output.

Example:

```
Device# show ip vrf detail

VRF vpn1; default RD 100:1; default VPNID A1:3F6C
  Interfaces:
    FastEthernet1/1/1       FastEthernet1/0/1
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:100:1
  Import VPN route-target communities
    RT:100:1                 RT:500:1
  No import route-map
  No export route-map
VRF vpn2; default RD <not set>; default VPNID 2:3
  No interfaces
  Connected addresses are not in global routing table
  No Export VPN route-target communities
  No Import VPN route-target communities
  No import route-map
  No export route-map
VRF vpn5; default RD 500:1; default VPNID <not set>
  Interfaces:
```

Configuration Examples for Assigning an ID Number to an MPLS VPN

Example: Specifying an MPLS VPN ID

The following example specifies the MPLS VPN ID assigned to the virtual routing and forwarding (VRF) table called vpn1:

```
Device# configure terminal
Device(config)# ip vrf vpn1
Device(config-vrf)# vpn id a1:3f6c
```

Example: Verifying the MPLS VPN ID Configuration

The following is sample output of the **show ip vrf detail** command, one of the commands that can be used to verify the MPLS VPN ID configuration. Use this command to see all the virtual routing and forwarding (VRF) instances on a provider edge (PE) device. This command displays all the MPLS VPN IDs that are configured on the device, their associated VRF names, and VRF route distinguishers (RDs). If a VRF table in the PE device has not been assigned a VPN ID, that VRF entry is not included in the output.

```
Device# show ip vrf detail

VRF vpn1; default RD 100:1; default VPNID A1:3F6C
  Interfaces:
    FastEthernet1/1/1       FastEthernet1/0/1
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:100:1
  Import VPN route-target communities
    RT:100:1                 RT:500:1
  No import route-map
  No export route-map
VRF vpn2; default RD <not set>; default VPNID 2:3
  No interfaces
  Connected addresses are not in global routing table
  No Export VPN route-target communities
  No Import VPN route-target communities
  No import route-map
  No export route-map
VRF vpn5; default RD 500:1; default VPNID <not set>
  Interfaces:
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco Master Command List, All Releases

Related Topic	Document Title
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference

Standards and RFCs

Standard/RFC	Title
IEEE Std 802-1990	*IEEE Local and Metropolitan Area Networks: Overview and Architecture*
RFC 2685	*Virtual Private Networks Identifier*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS VPN ID

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 6: Feature Information for MPLS VPN ID

Feature Name	Releases	Feature Configuration Information
MPLS VPN ID	12.0(17)ST 12.2(8)T 12.2(11)S 12.2(17b)SXA 12.2(27)SBB Cisco IOS XE Release 2.1	You can identify VPNs by a VPN identification number, as described in RFC 2685. This implementation of the VPN ID feature is used for identifying a VPN. In Cisco IOS Release 12.0(17)ST, this feature was introduced. In Cisco IOS Releases 12.2(8)T, 12.2(11)S, 12.2(17b)SXA, and 12.2(27)SBB, this feature was integrated. In Cisco IOS XE Release 2.1, this feature was implemented on Cisco ASR 1000 Series Aggregation Services Routers. No commands were introduced or modified.

CHAPTER 8

MPLS VPN Show Running VRF

The MPLS VPN Show Running VRF feature provides a Cisco IOS CLI option to display a subset of the running configuration on a device that is linked to a Virtual Private Network (VPN) routing and forwarding (VRF) instance. You can display the configuration of a specific VRF or of all VRFs configured on a device.

On heavily loaded devices, the display of the configuration file might require several pages or screens. As the configuration increases in size and complexity, the possibility of misconfiguration also increases. You might find it difficult to trace a problem on a device where you have several VRFs configured. A command that displays all the elements of the configuration linked to a VRF allows for easier troubleshooting on a per-VRF basis and facilitates comparisons among configurations of different VRFs on the same device.

There are no configuration tasks for the MPLS VPN Show Running VRF feature.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for MPLS VPN Show Running VRF

- A Cisco software image that supports virtual routing and forwarding (VRF) instances installed on the device
- At least one VRF configured on the device

- Cisco Express Forwarding for Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) routing and forwarding

Restrictions for MPLS VPN Show Running VRF

Any element of the running configuration of the device that is not linked directly to a virtual routing and forwarding (VRF) instance is not displayed. For example, a route map associated with a Border Gateway Protocol (BGP) neighbor in a VRF address-family configuration is not displayed. The VRF address-family configuration under BGP is displayed, but the route-map configuration is not. An exception to this general rule is the display of a controller configuration.

Information About MPLS VPN Show Running VRF

Configuration Elements Displayed for MPLS VPN Show Running VRF

You can display the running configuration associated with a specific virtual routing and forwarding (VRF) instance or all VRFs on the device by entering the **show running-config vrf** command. To display the running configuration of a specific VRF, enter the name of the VRF as an argument to the **show running-config vrf** command. For example, for a VRF named vpn3, you enter:

```
Device# show running-config vrf vpn3
```

The **show running-config vrf** command displays the following elements of the running configuration on a device:

- The VRF configuration (This includes any configuration that is applied in the VRF submode.)
- The configuration of each interface in the VRF

Entering a **show run vrf** *vpn-name* command is the same as executing a **show running-config interface** *type number* for each interface that you display by use of the **show ip vrf** *vpn-name* command. The interfaces display in the same sorted order that you would expect from the **show ip interface** command.

For a channelized interface, the configuration of the controller is displayed (as shown by the **show run controller** *controller-name* command).

For a subinterface, the configuration of the main interface is displayed.

Display of VRF Routing Protocol Configuration

Open Shortest Path First (OSPF), Routing Information Protocol (RIP), Border Gateway Protocol (BGP), Enhanced Interior Gateway Routing Protocol (EIGRP), and static routing are routing protocols that support the virtual routing and forwarding (VRF) configuration.

OSPF has one process per VRF. The **show running-config vrf** command display includes the complete configuration of any OSPF process associated with the VRF. For example, the following shows the sample display for OSPF process 101, which is associated with the VRF named vpn3:

```
router ospf 101 vrf vpn3
```

```
 log-adjacency-changes
 area 1 sham-link 10.43.43.43 10.23.23.23 cost 10
 network 172.17.0.0 0.255.255.255 area 1
```

RIP, BGP, and EIGRP support VRF address-family configuration. If a VRF address family for the VRF exists for any of these routing protocols, a configuration in the following format is displayed:

```
router
protocol
 {AS
 | PID
}
!
address-family ipv4 vrf
 vrf-name
.
.
.
```

Where the *protocol* argument is one of the following: **rip**, **bgp** or **eigrp**; the *AS* argument is an autonomous system number; the *PID* argument is a process identifier; and the *vrf-name* argument is the name of the associated VRF.

The following shows a sample display for a BGP with autonomous system number 100 associated with a VRF named vpn3:

```
!
router bgp 100
!
address-family ipv4 vrf vpn3
 redistribute connected
 redistribute ospf 101 match external 1 external 2
 no auto-summary
 no synchronization
 exit-address-family
 !
```

The **show running-config vrf** command also includes the configuration of any static routes configured in the VRF. For example:

```
ip route vrf vpn1 10.1.1.0 255.255.255.0 10.30.1.1 global
ip route vrf vpn1 10.1.2.0 255.255.255.0 10.125.1.2
```

Display of Configuration Not Directly Linked to a VRF

Any element of a configuration that is not linked directly to a virtual routing and forwarding (VRF) instance is not displayed. In some instances, the display of the configuration of an element that is not directly linked to a VRF is required.

For example, the **show running-config vrf** command displays the configuration of an E1 controller whose serial subinterfaces are in a VRF. The command displays the controller configuration and the subinterface configuration.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco Master Command List, All Releases
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS VPN Show Running VRF

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 7: Feature Information for MPLS VPN Show Running VRF

Feature Name	Releases	Feature Information
MPLS VPN Show Running VRF	12.2(28)SB 12.0(32)SY 12.2(33)SRB 12.2(33)SXH 12.4(20)T Cisco IOS XE Release 2.5	The MPLS VPN Show Running VRF feature provides a CLI option to display a subset of the running configuration on a device that is linked to a VRF. You can display the configuration of a specific VRF or of all VRFs configured on a device. A command that displays all the elements of the configuration linked to a VRF allows for easier troubleshooting on a per-VRF basis and facilitates comparisons among configurations of different VRFs on the same device. In 12.2(28)SB, this feature was introduced. In 12.0(32)SY, support was added for a Cisco IOS 12.0SY release. In 12.2(33)SRB, support was added for a Cisco IOS 12.2SR release. In 122(33)SXH, support was added for a Cisco IOS 12.2SX release. In 12.4(20)T, support was added for a Cisco IOS 12.4T release. In Cisco IOS XE Release 2.5S, support was added for the Cisco ASR 1000 Series Routers.
		The following commands were introduced or modified: **show policy-map interface brief**, **show running-config vrf**.

Glossary

BGP—Border Gateway Protocol. An interdomain routing protocol that replaces External Gateway Protocol (EGP). BGP systems exchange reachability information with other BGP systems. BGP is defined by RFC 1163.

EGP—External Gateway Protocol. An internet protocol for exchanging routing information between autonomous systems. EGP is documented in RFC 904. Not to be confused with the general term exterior gateway protocol. EGP is an obsolete protocol that was replaced by Border Gateway Protocol (BGP).

EIGRP—Enhanced Interior Gateway Routing Protocol. Advanced version of Interior Gateway Routing Protocol (IGRP) developed by Cisco. Provides superior convergence properties and operating efficiency, and combines the advantages of link state protocols with those of distance vector protocols.

IGP—Interior Gateway Protocol. An 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).

IGRP—Interior Gateway Routing Protocol. An Interior Gateway Protocol (IGP) developed by Cisco to address the issues associated with routing in large, heterogeneous networks.

MPLS—Multiprotocol Label Switching. A switching method that forwards IP traffic through the use of a label. This label instructs the devices and the switches in the network where to forward each packet based on preestablished IP routing information.

OSPF—Open Shortest Path First. A link-state, hierarchical, Interior Gateway Protocol (IGP) routing algorithm and routing protocol proposed as a successor to Routing Information Protocol (RIP) in the Internet community.

OSPF features include least-cost routing, multipath routing, and load balancing. OSPF was derived from an early version of the Intermediate System-to-Intermediate System (IS-IS) protocol.

RIP—Routing Information Protocol. Internal Gateway Protocol (IGP) supplied with UNIX Berkeley Software Distribution (BSD) systems. RIP is the most common IGP in the Internet. It uses hop count as a routing metric.

VPN—Virtual Private Network. The result of a device configuration that enables IP traffic to use tunneling to travel securely over a public TCP/IP network.

CHAPTER 9

MPLS VPN Half-Duplex VRF

The MPLS VPN Half-Duplex VRF feature provides scalable hub-and-spoke connectivity for subscribers of an Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) service. This feature addresses the limitations of hub-and-spoke topologies by removing the requirement of one virtual routing and forwarding (VRF) instance per spoke. This feature also ensures that subscriber traffic always traverses the central link between the wholesale service provider and the Internet service provider (ISP), whether the subscriber traffic is being routed to a remote network by way of the upstream ISP or to another locally or remotely connected subscriber.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for MPLS VPN Half-Duplex VRF

Half-Duplex VRF is supported with either an MPLS core network or an IP core (VRF lite) network.

Restrictions for MPLS VPN Half-Duplex VRF

The following features are not supported on interfaces configured with the MPLS VPN Half-Duplex VRF feature:

- Multicast
- MPLS VPN Carrier Supporting Carrier
- MPLS VPN Interautonomous Systems

Information About MPLS VPN Half-Duplex VRF

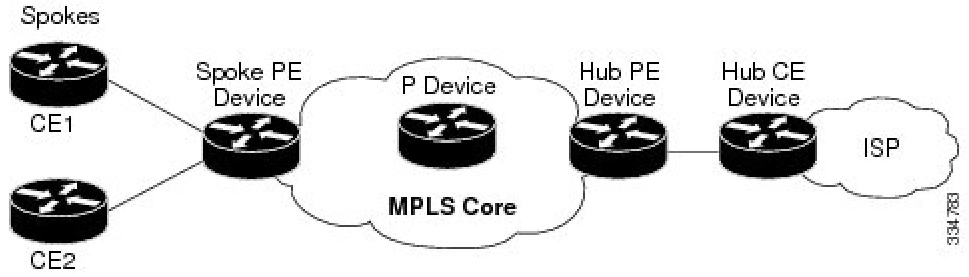
MPLS VPN Half-Duplex VRF Overview

The MPLS VPN Half-Duplex VRF feature provides:

- The MPLS VPN Half-Duplex VRF feature prevents local connectivity between subscribers at the spoke provider edge (PE) device and ensures that a hub site provides subscriber connectivity. Any sites that connect to the same PE device must forward intersite traffic using the hub site. This ensures that the routing done at the spoke site moves from the access-side interface to the network-side interface or from the network-side interface to the access-side interface, but never from the access-side interface to the access-side interface.
- The MPLS VPN Half-Duplex VRF feature prevents situations where the PE device locally switches the spokes without passing the traffic through the upstream Internet service provider (ISP). This prevents subscribers from directly connecting to each other, which causes the wholesale service provider to lose revenue.
- The MPLS VPN Half-Duplex VRF feature improves scalability by removing the requirement of one virtual routing and forwarding (VRF) instance per spoke. If the feature is not configured, when spokes are connected to the same PE device each spoke is configured in a separate VRF to ensure that the traffic between the spokes traverses the central link between the wholesale service provider and the ISP. However, this configuration is not scalable. When many spokes are connected to the same PE device, configuration of VRFs for each spoke becomes quite complex and greatly increases memory usage. This is especially true in large-scale wholesale service provider environments that support high-density remote access to Layer 3 Virtual Private Networks (VPNs).

The figure below shows a sample hub-and-spoke topology.

Figure 10: Hub-and-Spoke Topology



Upstream and Downstream VRFs

The MPLS VPN Half-Duplex VRF feature uses two unidirectional virtual routing and forwarding (VRF) instances to forward IP traffic between the spokes and the hub PE device:

- The upstream VRF forwards IP traffic from the spokes toward the hub provider edge (PE) device. This VRF typically contains only a default route but might also contain summary routes and several default routes. The default route points to the interface on the hub PE device that connects to the upstream Internet service provider (ISP). The device dynamically learns about the default route from the routing updates that the hub PE device or home gateway sends.

Note Although the upstream VRF is typically populated from the hub, it is possible also to have a separate local upstream interface on the spoke PE for a different local service that would not be required to go through the hub: for example, a local Domain Name System (DNS) or game server service.

- The downstream VRF forwards traffic from the hub PE device back to the spokes. This VRF can contain:
  - PPP peer routes for the spokes and per-user static routes received from the authentication, authorization, and accounting (AAA) server or from the Dynamic Host Control Protocol (DHCP) server
  - Routes imported from the hub PE device
  - Border Gateway Protocol (BGP), Open Shortest Path First (OSPF), Routing Information Protocol (RIP), or Enhanced Interior Gateway Routing Protocol (EIGRP) dynamic routes for the spokes

The spoke PE device redistributes routes from the downstream VRF into Multiprotocol Border Gateway Protocol (MP-BGP). That device typically advertises a summary route across the Multiprotocol Label Switching (MPLS) core for the connected spokes. The VRF configured on the hub PE device imports the advertised summary route.

A routing loop occurs when a per prefix label allocation mode is used, thereby not forwarding packets in downstream VRF. This can be prevented by using per VRF label allocation.

Reverse Path Forwarding Check

The Reverse Path Forwarding (RPF) check ensures that an IP packet that enters a device uses the correct inbound interface. The MPLS VPN Half-Duplex VRF feature supports unicast RPF check on the spoke-side interfaces. Because different virtual routing and forwarding (VRF) instances are used for downstream and upstream forwarding, the RPF mechanism ensures that source address checks occur in the downstream VRF.

Unicast RPF is disabled by default. .

How to Configure MPLS VPN Half-Duplex VRF

Configuring the Upstream and Downstream VRFs on the Spoke PE Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **address-family** {**ipv4** | **ipv6**}
6. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
7. **exit-address-family**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vrf1`	Configures a virtual routing and forwarding (VRF) table and enters VRF configuration mode. • The *vrf-name* argument is the name of the VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:1`	Creates routing and forwarding tables for a VRF. • The *route-distinguisher* argument specifies to add an 8-byte value to an IPv4 prefix to create a Virtual Private Network (VPN) IPv4 prefix. You can enter a route distinguisher in either of these formats: • 16-bit autonomous system number (ASN): your 32-bit number For example, 101:3.

	Command or Action	Purpose
		• 32-bit IP address: your 16-bit number For example, 192.168.122.15:1.

Step 5	**address-family** {**ipv4** | **ipv6**}
	Example:
	`Device(config-vrf) address-family ipv4`
		Enters VRF address family configuration mode to specify an address family for a VRF.
		• The **ipv4** keyword specifies an IPv4 address family for a VRF.
		• The **ipv6** keyword specifies an IPv6 address family for a VRF.
		Note The MPLS VPN Half Duplex VRF feature supports only the IPv4 address family.

Step 6	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target both 100:2`	Creates a route-target extended community for a VRF. • The **import** keyword specifies to import routing information from the target VPN extended community. • The **export** keyword specifies to export routing information to the target VPN extended community. • The **both** keyword specifies to import both import and export routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities.
Step 7	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode.

Step 8	**end**
	Example:
	`Device(config-vrf)# end`
		Returns to privileged EXEC mode.

Associating a VRF with an Interface

Perform the following task to associate a virtual routing and forwarding (VRF) instance with an interface, which activates the VRF.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name* [**downstream** *vrf-name2*
5. **ip address** *ip-address mask* [**secondary**]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet 0/1`	Configures an interface type and enters interface configuration mode. • The *type* argument identifies the type of interface to be configured. • The *number* argument identifies the port, connector, or interface card number.
Step 4	**vrf forwarding** *vrf-name* [**downstream** *vrf-name2* **Example:** `Device(config-if)# vrf forwarding vrf1`	Associates a VRF with an interface or subinterface. • The *vrf-name* argument is the name of the VRF. • The **downstream** *vrf-name2* keyword and argument combination is the name of the downstream VRF into which peer and per-user routes are installed.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.24.24.24 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask of the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 6	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-if) end`	

Configuring the Downstream VRF for an AAA Server

To configure the downstream VRF for an AAA (RADIUS) server in broadband or remote access situations, enter the following Cisco attribute value:

lcp:interface-config=ip vrf forwarding U downstream D

In standard VPN situations, enter instead the following Cisco attribute value:

ip:vrf-id=U downstream D

Verifying the MPLS VPN Half-Duplex VRF Configuration

SUMMARY STEPS

1. **show vrf** [**ipv4** | **ipv6**] [**brief** | **detail** | **id** | **interfaces** | **lock** | **select**] [*vrf-name*]
2. **show ip route vrf** *vrf-name*
3. **show running-config** [**interface** *type number*]

DETAILED STEPS

Step 1 **show vrf** [**ipv4** | **ipv6**] [**brief** | **detail** | **id** | **interfaces** | **lock** | **select**] [*vrf-name*]

Displays information about all of the virtual routing and forwarding (VRF) instances configured on the device, including the downstream VRF for each associated interface or virtual access interface (VAI):

Example:

```
Device# show vrf
Name     Default RD     Interfaces
Down     100:1          POS3/0/3 [D]
                        POS3/0/1 [D]
         100:3          Loopback2
                        Virtual-Access3 [D]
                        Virtual-Access4 [D]
Up       100:2          POS3/0/3
                        POS3/0/1
         100:4          Virtual-Access3
```

Use the **show vrf detail** *vrf-name* command to display detailed information about the VRF you specify, including all interfaces, subinterfaces, and VAIs associated with the VRF.

If you do not specify a value for the *vrf-name* argument, detailed information about all of the VRFs configured on the device appears.

The following example shows how to display detailed information for the VRF called vrf1, in a broadband or remote access case:

Example:

```
Device# show vrf detail vrf1
VRF D; default RD 2:0; default VPNID <not set>
  Interfaces:
         Loopback2           Virtual-Access3 [D]  Virtual-Access4 [D]
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:2:0
  Import VPN route-target communities
    RT:2:1
  No import route-map
  No export route-map
VRF U; default RD 2:1; default VPNID <not set>
  Interfaces:
    Virtual-Access3          Virtual-Access4
  Connected addresses are not in global routing table
  No Export VPN route-target communities
  Import VPN route-target communities
    RT:2:1
  No import route-map
  No export route-map
```

The following example shows the VRF detail in a standard Virtual Private Network (VPN) situation:

Example:

```
Device# show vrf detail
VRF Down; default RD 100:1; default VPNID <not set> VRF Table ID = 1
  Description: import only from hub-pe
  Interfaces:
    Pos3/0/3 [D]        Pos3/0/1:0.1 [D]
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:100:0
  Import VPN route-target communities
    RT:100:1
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
 VRF Up; default RD 100:2; default VPNID <not set> VRF Table ID = 2
  Interfaces:
    Pos3/0/1            Pos3/0/3
  Connected addresses are not in global routing table
  No Export VPN route-target communities
  Import VPN route-target communities
    RT:100:1
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
```

Step 2 **show ip route vrf** *vrf-name*

Displays the IP routing table for the VRF you specify, and information about the per-user routes installed in the downstream VRF.

The following example shows how to display the routing table for the downstream VRF named D, in a broadband or remote access situation:

Example:

```
Device# show ip route vrf D

Routing Table: D
```

```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       I - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS interarea
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is not set
 10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
U   10.0.0.2/32 [1/0] via 10.0.0.1
S   10.0.0.0/8 is directly connected, Null0
U   10.0.0.5/32 [1/0] via 10.0.0.2
C   10.8.1.2/32 is directly connected, Virtual-Access4
C   10.8.1.1/32 is directly connected, Virtual-Access3
```

The following example shows how to display the routing table for the downstream VRF named Down, in a standard VPN situation:

Example:

```
Device# show ip route vrf Down

Routing Table: Down
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       I - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is 10.13.13.13 to network 0.0.0.0
C  10.2.0.0/8 is directly connected, Pos3/0/3
     10.3.0.0/32 is subnetted, 1 subnets
B       10.4.16.16 [200/0] via 10.13.13.13, 1w3d
B  10.6.0.0/8 [200/0] via 10.13.13.13, 1w3d
C  10.0.0.0/8 is directly connected, Pos3/0/1
 10.7.0.0/16 is subnetted, 1 subnets
B   10.7.0.0 [20/0] via 10.0.0.2, 1w3d
     10.0.6.0/32 is subnetted, 1 subnets
B       10.0.6.14 [20/0] via 10.0.0.2, 1w3d
     10.8.0.0/32 is subnetted, 1 subnets
B       10.8.15.15 [20/0] via 10.0.0.2, 1w3d
B*   0.0.0.0/0 [200/0] via 10.0.0.13, 1w3d
```

The following example shows how to display the routing table for the upstream VRF named U in a broadband or remote access situation:

Example:

```
Device# show ip route vrf U
Routing Table: U
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       I - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS interarea
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is 192.168.0.20 to network 0.0.0.0
 10.0.0.0/32 is subnetted, 1 subnets
C   10.0.0.8 is directly connected, Loopback2
B*   0.0.0.0/0 [200/0] via 192.168.0.20, 1w5d
```

The following example shows how to display the routing table for the upstream VRF named Up in a standard VPN situation:

Example:

```
Device# show ip route vrf Up
Routing Table: Up
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       I - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is 10.13.13.13 to network 0.0.0.0
 10.2.0.0/32 is subnetted, 1 subnets
C   10.2.0.1 is directly connected, Pos3/0/3
     10.3.0.0/32 is subnetted, 1 subnets
B       10.3.16.16 [200/0] via 10.13.13.13, 1w3d
B  10.6.0.0/8 [200/0] via 10.13.13.13, 1w3d
 10.0.0.0/32 is subnetted, 1 subnets
C   10.0.0.1 is directly connected, Pos3/0/1
B*   0.0.0.0/0 [200/0] via 10.13.13.13, 1w3d
```

Step 3 **show running-config** [**interface** *type number*]

Displays information about the interface you specify, including information about the associated upstream and downstream VRFs.

The following example shows how to display information about subinterface POS 3/0/1:

Example:

```
Device# show running-config interface POS 3/0/1
Building configuration...
Current configuration : 4261 bytes
!
interface POS3/0/1
ip vrf forwarding Up downstream Down
ip address 10.0.0.1 255.0.0.0
end
```

Configuration Examples for MPLS VPN Half-Duplex VRF

Examples: Configuring the Upstream and Downstream VRFs on the Spoke PE Device

The following example configures an upstream virtual routing and forwarding (VRF) instance named Up:

```
Device> enable
Device# configure terminal
Device(config)# vrf definition Up
Device(config-vrf)# rd 1:0
Device(config-vrf)# address-family ipv4
Device(config-vrf-af)# route-target import 1:0
Device(config-vrf-af)# exit-address-family
```

The following example configures a downstream VRF named Down:

```
Device> enable
Device# configure terminal
Device(config)# vrf definition Down
Device(config-vrf)# rd 1:8
Device(config-vrf)# address-family ipv4
Device(config-vrf-af)# route-target import 1:8
Device(config-vrf-af)# exit-address-family
```

Example: Associating a VRF with an Interface

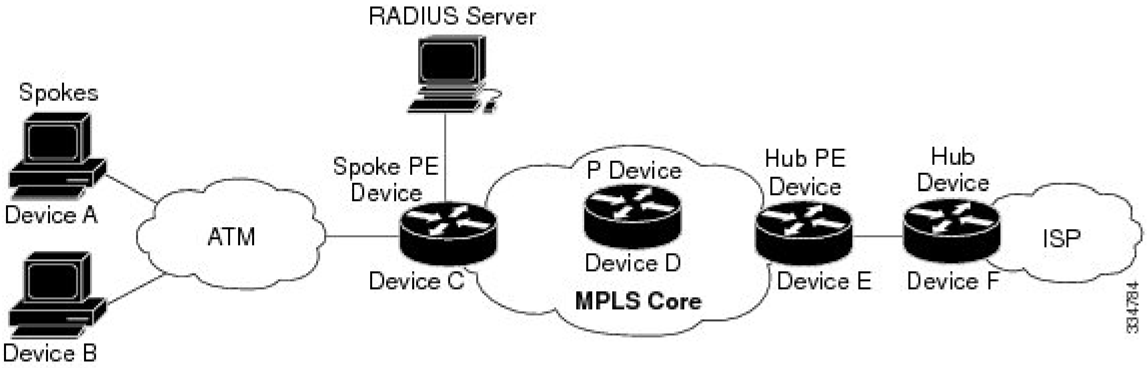
The following example associates the virtual routing and forwarding (VRF) instance named Up with POS 3/0/1 subinterface and specifies the downstream VRF named Down:

```
Device> enable
Device# configure terminal
Device(config)# interface POS 3/0/1
Device(config-if)# vrf forwarding Up downstream Down
Device(config-if)# ip address 10.0.0.1 255.0.0.0
```

Example Configuring MPLS VPN Half-Duplex VRF Using Static CE-PE Routing

This example uses the hub-and-spoke topology shown in the figure below with local authentication (that is, the RADIUS server is not used):

Figure 11: Sample Topology



```
vrf definition D
 rd 1:8
 address-family ipv4
 route-target export 1:100
 exit-address-family
!
vrf definition U
 rd 1:0
 address-family ipv4
 route-target import 1:0
 exit-address-family
```

```
!
ip cef
vpdn enable
!
vpdn-group U
 accept-dialin
  protocol pppoe
  virtual-template 1
!
interface Loopback 2
 vrf forwarding U
 ip address 10.0.0.8 255.255.255.255
!
interface ATM 2/0
 description Mze ATM3/1/2
 no ip address
 no atm ilmi-keepalive
 pvc 0/16 ilmi
!
 pvc 3/100
  protocol pppoe
!
pvc 3/101
  protocol pppoe
!
```

Example: Configuring MPLS VPN Half-Duplex VRF Using RADIUS Server and Static CE-PE Routing

The following example shows how to connect two Point-to-Point Protocol over Ethernet (PPPoE) clients to a single virtual routing and forwarding (VRF) pair on the spoke provider edge (PE) device named Device C. Although both PPPoE clients are configured in the same VRF, all communication occurs using the hub PE device. Half-duplex VRFs are configured on the spoke PE. The client configuration is downloaded to the spoke PE from the RADIUS server.

This example uses the hub-and-spoke topology shown in the figure above.

Note The wholesale provider can forward the user authentication request to the corresponding ISP. If the ISP authenticates the user, the wholesale provider appends the VRF information to the request that goes back to the PE device.

```
aaa new-model
!
aaa group server radius R
 server 10.0.20.26 auth-port 1812 acct-port 1813
!
aaa authentication ppp default group radius
aaa authorization network default group radius
!
vrf defintion D
 description Downstream VRF - to spokes
 rd 1:8
 address-family ipv4
 route-target export 1:100
 exit-address-family
!
```

```
vrf definition U
 description Upstream VRF - to hub
 rd 1:0
 address-family ipv4
 route-target import 1:0
 exit-address-family
!
ip cef
vpdn enable
!
vpdn-group U
 accept-dialin
  protocol pppoe
  virtual-template 1
!
interface Loopback2
 vrf forwarding U
 ip address 10.0.0.8 255.255.255.255
!
interface ATM2/0
  pvc 3/100
  protocol pppoe
 !
pvc 3/101
  protocol pppoe
 !
interface virtual-template 1
 no ip address
 ppp authentication chap
!
router bgp 1
 no synchronization
 neighbor 172.16.0.34 remote-as 1
 neighbor 172.16.0.34 update-source Loopback0
 no auto-summary
 !
address-family vpnv4
  neighbor 172.16.0.34 activate
  neighbor 172.16.0.34 send-community extended
  auto-summary
  exit-address-family
 !
address-family ipv4 vrf U
  no auto-summary
  no synchronization
  exit-address-family
!
address-family ipv4 vrf D
  redistribute static
  no auto-summary
  no synchronization
  exit-address-family
!
ip local pool U-pool 10.8.1.1 2.8.1.100
ip route vrf D 10.0.0.0 255.0.0.0 Null0
!
radius-server host 10.0.20.26 auth-port 1812 acct-port 1813
radius-server key cisco
```

Example: Configuring MPLS VPN Half-Duplex VRF Using Dynamic CE-PE Routing

The following example shows how to use Open Shortest Path First (OSPF) to dynamically advertise the routes on the spoke sites.

This example uses the hub-and-spoke topology shown in the figure above.

Creating the VRFs

```
vrf definition Down
rd 100:1
address-family ipv4
route-target export 100:0
exit-address-family
!
vrf definition Up
rd 100:2
address-family ipv4
route-target import 100:1
exit-address-family
```

Enabling MPLS

```
mpls ldp graceful-restart
mpls ldp router-id Loopback0 force
mpls label protocol ldp
```

Configuring BGP Toward Core

```
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 10.13.13.13 remote-as 100
 neighbor 10.13.13.13 update-source Loopback0
 !
 address-family vpnv4
 neighbor 10.13.13.13 activate
 neighbor 10.13.13.13 send-community extended
 bgp scan-time import 5
 exit-address-family
```

Configuring BGP Toward Edge

```
address-family ipv4 vrf Up
no auto-summary
no synchronization
exit-address-family
!
address-family ipv4 vrf Down
redistribute ospf 1000 vrf Down
no auto-summary
no synchronization
exit-address-family
```

Spoke PE's Core-Facing Interfaces and Processes

```
interface Loopback 0
 ip address 10.11.11.11 255.255.255.255
!
interface POS 3/0/2
 ip address 10.0.1.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
!
router ospf 100
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 network 10.11.11.11 0.0.0.0 area 100
 network 10.0.1.0 0.255.255.255 area 100
```

Spoke PE's Edge-Facing Interfaces and Processes

```
interface Loopback 100
vrf forwarding Down
 ip address 10.22.22.22 255.255.255.255
!
interface POS 3/0/1
vrf forwarding Up downstream Down
 ip address 10.0.0.1 255.0.0.0
!
interface POS 3/0/3
vrf forwarding Up downstream Down
 ip address 10.2.0.1 255.0.0.0
!
router ospf 1000 vrf Down
 router-id 10.22.22.22
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
 redistribute bgp 100 metric-type 1 subnets
 network 10.22.22.22 0.0.0.0 area 300
 network 10.0.0.0 0.255.255.255 area 300
 network 10.2.0.0 0.255.255.255 area 300
 default-information originate
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco Master Command List, All Releases
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference

Related Topic	Document Title
MPLS VPNs	“MPLS Virtual Private Networks” module
Configuring IPv4 and IPv6 VRFs	“MPLS VPN VRF CLI for IPv4 and IPv6 VPNs” module

Standards and RFCs

Standard/RFC	Title
RFC 2547	BGP/MPLS VPNs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS VPN Half-Duplex VRF

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 8: Feature Information for MPLS VPN Half-Duplex VRF

Feature Name	Releases	Feature Information
MPLS VPN - Half Duplex VRF (HDVRF) Support with Static Routing	12.3(6) 12.3(11)T 12.2(28)SB Cisco IOS XE Release 2.5	This feature ensures that VPN clients that connect to the same PE device at the edge of the MPLS VPN use the hub site to communicate. In Cisco IOS Release 12.3(6), this feature was introduced. In Cisco IOS Release 12.4(20)T, this feature was integrated. In Cisco IOS Release 12.2(28)SB, this feature was integrated In Cisco IOS XE Release 2.5, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers.

Feature Name	Releases	Feature Information
MPLS VPN Half-Duplex VRF	12.2(28)SB2 12.4(20)T 12.2(33)SRC Cisco IOS XE Release 2.5	In Cisco IOS Release 12.2(28)SB2, support for dynamic routing protocols was added. In Cisco IOS Release 12.4(20)T, this feature was integrated. In Cisco IOS Release 12.2(33)SRC, this feature was integrated. In Cisco IOS XE Release 2.5, this feature was integrated. The following commands were introduced or modified: **ip vrf forwarding** (interface configuration), **show ip interface**, **show vrf**.

CHAPTER 10

MPLS VPN VRF CLI for IPv4 and IPv6 VPNs

The MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature introduces Cisco commands that allow you to enable an IPv4 and IPv6 VPN in the same VRF instance and to simplify the migration from a single-protocol VRF configuration to a multiprotocol VRF configuration. A multiprotocol VRF allows you to share route targets policies (import and export) between IPv4 and IPv6 or to configure separate route-target policies for IPv4 and IPv6 VPNs.

This document describes how to configure a Virtual Private Network (VPN) routing and forwarding (VRF) instance for IPv4 and IPv6 VPNs and describes how to upgrade your existing single-protocol IPv4-only VRF to a multiprotocol VRF configuration.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for MPLS VPN VRF CLI for IPv4 and IPv6 VPNs

- For migration—An IPv4 Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) virtual routing and forwarding (VRF) instance must exist.

- For a new VRF configuration—Cisco Express Forwarding and an MPLS label distribution method, either Label Distribution Protocol (LDP) or MPLS traffic engineering (TE), must be enabled on all devices in the core, including the provider edge (PE) devices.

Restrictions for MPLS VPN VRF CLI for IPv4 and IPv6 VPNs

- Once you have converted to a multiprotocol virtual routing and forwarding (VRF) instance, you cannot convert the VRF back to an IPv4-only single-protocol VRF.
- You can associate an interface with only one VRF. You cannot configure a VRF for IPv4 and a different VRF for IPv6 on the same interface.
- You can configure only IPv4 and IPv6 address families in a multiprotocol VRF. Other protocols (IPX, AppleTalk, and the like) are not supported.

Information About MPLS VPN VRF CLI for IPv4 and IPv6 VPNs

VRF Concepts Similar for IPv4 and IPv6 MPLS VPNs

Virtual Private Networks (VPNs) for IPv6 use the same virtual routing and forwarding (VRF) concepts that IPv4 Multiprotocol Label Switching (MPLS) VPNs use, such as address families, route distinguishers, route targets, and VRF identifiers. Customers that use both IPv4 and IPv6 VPNs might want to share VRF policies between address families. They might want a way to define applicable VRF policies for all address families, instead of defining VRF policies for an address family individually as they do for or a single-protocol IPv4-only VRF.

Prior to the introduction of the MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature, a VRF applied only to an IPv4 address family. A one-to-one relationship existed between the VRF name and a routing and forwarding table identifier, between a VRF name and a route distinguisher (RD), and between a VRF name and a VPN ID. This configuration is called a single-protocol VRF.

The MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature introduces support for a multiple address-family (multi-AF) VRF structure. The multi-AF VRF allows you to define multiple address families under the same VRF. A given VRF, identified by its name and a set of policies, can apply to both an IPv4 VPN and an IPv6 VPN at the same time. This VRF can be activated on a given interface, even though the routing and forwarding tables are different for the IPv4 and IPv6 protocols. This configuration is called a multiprotocol VRF.

Single-Protocol VRF to Multiprotocol VRF Migration

Prior to the introduction of the MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature, you could create a single-protocol IPv4-only virtual routing and forwarding (VRF) instance. You created a single-protocol VRF by entering the **ip vrf** command. To activate the single-protocol VRF on an interface, you entered the **ip vrf forwarding** (interface configuration) command.

After the introduction of the MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature, you create a multiprotocol VRF by entering the **vrf definition** command. To activate the multiprotocol VRF on an interface, you enter the **vrf forwarding** command.

The MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature introduces the **vrf upgrade-cli multi-af-mode** {**common-policies** | **non-common-policies**} [**vrf** *vrf-name*] command that forces VRF configuration migration from a single-protocol VRF model to a multiprotocol VRF model:

- If the route-target policies apply to all address families configured in the multi-AF VRF, use the **common-policies** keyword.
- If the route-target policies apply only to the IPv4 address family that you are migrating, use the **non-common-policies** keyword.

After you enter the **vrf upgrade-cli** command and save the configuration to NVRAM, the single-protocol VRF configuration is saved as a multiprotocol VRF configuration. In the upgrade process, the**ip vrf** command is converted to the **vrf definition** command (global configuration commands) and the **ip vrf forwarding** command is converted to the **vrf forwarding** command (interface configuration command). The **vrf upgrade-cli** command has a one-time immediate effect.

You might have both IPv4-only VRFs and multiprotocol VRFs on your device. Once you create a VRF, you can edit it using only the commands in the mode in which it was created. For example, you created a VRF named vrf2 with the following multiprotocol VRF commands:

```
Device# configure terminal
Enter configuration command, one per line. End with CNTL/Z
Device(config)# vrf definition vrf2
Device(config-vrf)# rd 2:2
Device(config-vrf)# route-target import 2:2
Device(config-vrf)# route-target export 2:2
Device(config-vrf)# end
```

If you try to edit VRF vrf2 with IPv4-only VRF commands, you receive the following message:

```
Device# configure terminal
Enter configuration command, one per line. End with CNTL/Z
Device(config)# ip vrf vrf2
% Use 'vrf definition vrf2' command
```

If you try to edit an IPv4-only VRF with the multiprotocol VRF commands, you receive this message, where <vrf-name> is the name of the IPv4-only VRF:

```
% Use 'ip vrf <vrf-name>' command
```

The **ip vrf** *name* and **ip vrf forwarding** (interface configuration) commands will be available for a period of time before they are removed. Use the **vrf upgrade-cli** command to migrate your older IPv4-only VRFs to the new multiprotocol VRF configuration. When you need to create a new VRF—whether the VRF is for an IPv4 VPN, or IPv6 VPN, or both—use the multiprotocol VRF **vrf definition** and **vrf forwarding** commands that support a multi-AF configuration.

Multiprotocol VRF Configuration Characteristics

In a multiprotocol virtual routing and forwarding (VRF) configuration, you can configure both IPv4 VRFs and IPv6 VRFs under the same address family or configure separate VRFs for each IPv4 or IPv6 address family. The multiprotocol VRF configuration has the following characteristics:

- The VRF name identifies a VRF, which might have both IPv4 and IPv6 address families. On the same interface, you cannot have IPv4 and IPv6 address families using different VRF names.

- The route distinguisher (RD), VPN ID, and Simple Network Management Protocol (SNMP) context are shared by both IPv4 and IPv6 address families for a given VRF.
- The policies (route target, for example) specified in multi-AF VRF mode, outside the address-family configuration, are defaults to be applied to each address family. Route targets are the only VRF characteristics that can be defined inside and outside an address family.

The following is also true when you associate a multiprotocol VRF with an interface:

- Binding an interface to a VRF (**vrf forwarding** *vrf-name* command) removes all IPv4 and IPv6 addresses configured on that interface.
- Once you associate a VRF with a given interface, all active address families belong to that VRF. The exception is when no address of the address-family type is configured, in which case the protocol is disabled.
- Configuring an address on an interface that is bound to a VRF requires that the address family corresponding to the address type is active for that VRF. Otherwise, an error message is issued stating that the address family must be activated first in the VRF.

Backward compatibility with the single-protocol VRF CLI is supported in with the introduction of the MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature. This means that you might have single-protocol and multiprotocol CLI on the same device, but not in the same VRF configuration.

The single-protocol CLI continues to allow you to define an IPv4 address within a VRF and an IPv6 address in the global routing table on the same interface.

How to Configure MPLS VPN VRF CLI for IPv4 and IPv6 VPNs

Configuring a VRF for IPv4 and IPv6 MPLS VPNs

Perform the following task to configure a virtual routing and forwarding (VRF) instance for IPv4 and IPv6 Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). When you configure a VRF for both IPv4 and IPv6 VPNs (a multiprotocol VRF), you can choose to configure route-target policies that apply to all address families in the VRF, or you can configure route-target policies that apply to individual address families in the VRF.

The following task shows how to configure a VRF that has that has route-target policies defined for IPv4 and IPv6 VPNs in separate VRF address families.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **address-family** {**ipv4** | **ipv6**}
6. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
7. **exit-address-family**
8. **address-family** {**ipv4** | **ipv6**}

9. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vrf1`	Configures a VRF routing table and enters VRF configuration mode. • The *vrf-name* argument is the name of the VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:1`	Creates routing and forwarding tables for a VRF. • The *route-distinguisher* argument specifies to add an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter a route distinguisher in either of these formats: • 16-bit autonomous system number (ASN): your 32-bit number For example, 101:3. • 32-bit IP address: your 16-bit number For example, 192.168.122.15:1.

Step 5	**address-family** {**ipv4** | **ipv6**}
	Example:
	`Device(config-vrf) address-family ipv4`
		Enters VRF address family configuration mode to specify an address family for a VRF.
		• The **ipv4** keyword specifies an IPv4 address family for a VRF.
		• The **ipv6** keyword specifies an IPv6 address family for a VRF.

Step 6	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target both 100:2`	Creates a route-target extended community for a VRF. • The **import** keyword specifies to import routing information from the target VPN extended community. • The **export** keyword specifies to export routing information to the target VPN extended community.

	Command or Action	Purpose
		• The **both** keyword specifies to import both import and export routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities.
Step 7	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits from VRF address family configuration mode.

Step 8	**address-family** {**ipv4** | **ipv6**}
	Example:
	`Device(config-vrf) address-family ipv6`
		Enters VRF address family configuration mode to specify an address family for a VRF.
		• The **ipv4** keyword specifies an IPv4 address family for a VRF.
		• The **ipv6** keyword specifies an IPv6 address family for a VRF.

Step 9	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target both 100:3`	Creates a route-target extended community for a VRF. • The **import** keyword specifies to import routing information from the target VPN extended community. • The **export** keyword specifies to export routing information to the target VPN extended community. • The **both** keyword specifies to import both import and export routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities. Enter the **route-target** command one time for each target community.

Step 10	**end**
	Example:
	`Device(config-vrf-af)# end`
		Returns to privileged EXEC mode.

Associating a Multiprotocol VRF with an Interface

Perform the following task to associate a multiprotocol virtual routing and forwarding (VRF) instance with an interface. Associating the VRF with an interface activates the VRF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask* [**secondary**]
6. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*}
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet 0/1`	Configures an interface type and enters interface configuration mode. • The *type* argument identifies the type of interface to be configured. • The *number* argument identifies the port, connector, or interface card number.
Step 4	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding vrf1`	Associates a VRF with an interface or subinterface. • The *vrf-name* argument is the name of the VRF.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.24.24.24 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask of the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is

	Command or Action	Purpose
		omitted, the configured address is the primary IP address.

Step 6	**ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*}
	Example:
	`Device(config-if)# ipv6 address 2001:0DB8:0300:0201::/64`
		Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
		• The *ipv6-address* argument is the IPv6 address to be used.
		• The *prefix-length* argument is the length of the IPv6 prefix, which is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.
		• The *prefix-name* argument is a general prefix that specifies the leading bits of the network to be configured on the interface.
		• The *sub-bits* argument is the subprefix bits and host bits of the address to be concatenated with the prefixes provided by the general prefix specified with the *prefix-name* argument.
		The *sub-bits* argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.

Step 7	**end**
	Example:
	`Device(config-if) end`
		Returns to privileged EXEC mode.

Verifying the MPLS VPN VRF CLI for IPv4 and IPv6 VPNs Configuration

Perform the following task to verify the MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature configuration, that is, to show that the virtual routing and forwarding (VRF) configuration is upgraded to a multi-AF multiprotocol VRF.

SUMMARY STEPS

1. **enable**
2. **show running-config vrf** [*vrf-name*]
3. **show vrf**
4. **show vrf detail** [*vrf-name*]
5. **exit**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password, if prompted. For example:

Example:

```
Device> enable
Device#
```

Step 2 **show running-config vrf** [*vrf-name*]

Verifies that the upgrade to a multi-AF multiprotocol VRF configuration was successful. The following is sample command output before the upgrade to a multi-AF multiprotocol VRF:

Example:

```
Device# show running-config vrf vpn2

Building configuration...
Current configuration : 604 bytes
ip vrf vpn2
 rd 1:1
 route-target both 1:1
!
!
interface Loopback1
 ip vrf forwarding vpn2
 ip address 10.43.43.43 255.255.255.255
!
```

The following is sample command output after you upgrade to a multi-AF multiprotocol VRF with common policies for all address families:

Example:

```
Device# show running-config vrf vpn1

Building configuration...
Current configuration : 604 bytes
vrf definition vpn1
 rd 1:1
 route-target both 1:1
!
 address-family 1pv4
 exit-address-family
!
!
interface Loopback1
 ip vrf forwarding vpn1
 ip address 10.43.43.43 255.255.255.255
!
```

This configuration contains the **vrf definition** command. The **vrf definition** command replaces the **ip vrf** command in the multi-AF multiprotocol VRF configuration.

Step 3 **show vrf**

Verifies that the upgrade to a multi-AF multiprotocol VRF configuration was successful. The **show vrf** command replaces the **show ip vrf** command when a VRF configuration is updated to a multi-AF multiprotocol VRF configuration. The **show vrf** command displays the protocols defined for a VRF. The following command shows sample output after you upgrade a single-protocol VRF configuration to a multi-AF multiprotocol VRF configuration:

Example:

```
Device# show vrf vpn1

  Name                             Default RD     Protocols          Interfaces
  vpn1                             1:1            ipv4               Lo1/0
```

The following is sample output from the **show ip vrf vp1** command. Compare this to the output of the **show vrf vpn1** command. The protocols under the VRF are not displayed.

Example:

```
Device# show ip vrf vrf1

  Name      Default RD   Interface
  vpn1      1:1          Loopback1
```

The following is sample output from the **show vrf** command for multiprotocol VRFs, one of which contains both IPv4 and IPv6 protocols:

Example:

```
Device# show vrf

  Name                             Default RD     Protocols          Interfaces
  vpn1                             1:1            ipv4               Lo1/0
  vpn2                             100:3          ipv4               Lo23  AT3/0/0.1
  vpn4                             100:2          ipv4,ipv6
```

Step 4 **show vrf detail** [*vrf-name*]

Displays all characteristics of the defined VRF to verify that the configuration is as you expected. For example, if your VRF configuration for VRF vpn1 is as follows:

Example:

```
vrf definition vpn1
 route-target both 100:1
 route-target import 100:2
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 route-target both 100:1
 route-target import 100:3
 exit-address-family
```

This command displays the following:

Example:

```
Device# show vrf detail vpn1

VRF vpn1 (VRF Id = 3); default RD <not set>; default VPNID <not set>
```

```
  No interfaces
Address family ipv4 (Table ID = 3 (0x3)):
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:100:1
  Import VPN route-target communities
    RT:100:1                 RT:100:2
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
Address family ipv6 (Table ID = 503316483 (0x1E000003)):
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:100:1
  Import VPN route-target communities
    RT:100:1                 RT:100:3
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
```

Step 5 **exit**

Returns to user EXEC mode. For example:

Example:

```
Device# exit
Device>
```

Migrating from a Single-Protocol IPv4-Only VRF to a Multiprotocol VRF Configuration

Perform the following task to force migration from a single-protocol IPv4-only virtual routing and forwarding (VRF) configuration to a multiprotocol VRF configuration.

The multiprotocol VRF configuration allows you to define multiple address families under the same VRF. A given VRF, identified by its name and a set of policies, can apply to both an IPv4 VPN and an IPv6 VPN at the same time. This VRF can be activated on a given interface, even though the routing and forwarding tables are different for the IPv4 and IPv6 protocols.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf upgrade-cli multi-af-mode** {**common-policies** | **non-common-policies**} [**vrf** *vrf-name*]
4. **exit**
5. **show running-config vrf** [*vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf upgrade-cli multi-af-mode** {**common-policies** | **non-common-policies**} [**vrf** *vrf-name*] **Example:** `Device(config)# vrf upgrade-cli multi-af-mode common-policies vrf vpn4`	Upgrades a VRF instance or all VRFs configured on the device to support multiple address families under the same VRF. • The **multi-af-mode** keyword specifies an upgrade of a single-protocol VRF or all VRFs to a multiprotocol VRF that supports multi-AFs configuration. • The **common-policies** keyword specifies to copy the route-target policies to the common part of the VRF configuration so that the policies apply to all address families configured in the multi-AF VRF. • The **non-common-policies** keyword specifies to copy the route-target policies to the IPv4 address family part of the VRF configuration so that the policies apply only to IPv4. • The **vrf** keyword specifies a VRF for the upgrade to a multi-AF VRF configuration. • The *vrf-name* argument is the name of the single-protocol VRF to upgrade to a multi-AF VRF configuration.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits to privileged EXEC mode.
Step 5	**show running-config vrf** [*vrf-name*] **Example:** `Device# show running-config vrf vpn4`	Displays the subset of the running configuration of a device that is linked to a specific VRF instance or to all VRFs configured on the device. • The *vrf-name* argument is the name of the VRF of which you want to display the configuration.

	Command or Action	Purpose
		Note The Cisco software image that supports the multiprotocol VRF commands might not support the **show running-config vrf** command. You can use the **show running-config** command instead.

Configuration Examples for MPLS VPN VRF CLI for IPv4 and IPv6 VPNs

Example: Multiprotocol VRF Configuration Single Protocol with Noncommon Policies

The following is an example of a multiprotocol virtual routing and forwarding (VRF) configuration for a single protocol (IPv4) with route-target policies in the address family configuration:

```
vrf definition vrf2
 rd 2:2
 !
 address-family ipv4
 route-target export 2:2
 route-target import 2:2
 exit-address-family
```

The RD (2:2) applies to all address families defined for VRF vrf2.

Example: Multiprotocol VRF Configuration Multiprotocol with Noncommon Policies

The following is an example of a multiprotocol virtual routing and forwarding (VRF) configuration for IPv4 and IPv6 Virtual Private Networks (VPNs) in which the route-target policies are defined in the separate address family configurations:

```
vrf definition vrf2
 rd 2:2
 !
 address-family ipv4
 route-target export 2:2
 route-target import 2:2
 exit-address-family
 !
 address-family ipv6
 route-target export 3:3
 route-target import 3:3
 exit-address-family
```

Example: Multiprotocol VRF Configuration Multiprotocol with Common Policies

The following is an example of a multiprotocol virtual routing and forwarding (VRF) configuration for IPv4 and IPv6 Virtual Private Networks (VPNs) with route-target policies defined in the global part of the VRF:

```
vrf definition vrf2
 rd 2:2
 route-target export 2:2
 route-target import 2:2
  !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
```

The route-target policies are defined outside the address family configurations. Therefore, the policies apply to all address families defined in VRF vrf2.

Example: Multiprotocol VRF Configuration Multiprotocol with Common and Noncommon Policies

The following is an example of a multiprotocol virtual routing and forwarding (VRF) configuration with route-target policies defined in both global and address family areas:

- For IPv6, the route-target definitions are defined under the address family. These definitions are used and the route-target definitions in the global area are ignored. Therefore, the IPv6 Virtual Private Network (VPN) ignores import 100:2.
- For IPv4, no route-target policies are defined under the address family, therefore, the global definitions are used.

```
vrf definition vfr1
 route-target export 100:1
 route-target import 100:1
 route-target import 100:2
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 route-target export 100:1
 route-target import 100:1
 route-target import 100:3
 exit-address-family
```

Examples: Configuring a VRF for IPv4 and IPv6 VPNs

The following example shows how to configure a virtual routing and forwarding (VRF) instance for IPv4 and IPv6 Virtual Private Networks (VPNs):

```
configure terminal
!
vrf definition vrf1
 rd 100:1
```

```
!
 address-family ipv4
 route-target both 100:2
 exit-address-family
!
 address-family ipv6
 route-target both 100:3
 exit-address-family
```

In this example, noncommon policies are defined in the address family configuration.

The following is an example of a VRF for IPv4 and IPv6 that has common policies defined in the global part of the VRF configuration:

```
configure terminal
!
vrf definition vrf2
 rd 200:1
 route-target both 200:2
!
 address-family ipv4
 exit-address-family
!
 address-family ipv6
 exit-address-family
 end
```

Example: Associating a Multiprotocol VRF with an Interface

The following example shows how to associate a multiprotocol virtual routing and forwarding (VRF) instance with an interface:

```
configure terminal
!
interface Ethernet 0/1
 vrf forwarding vrf1
 ip address 10.24.24.24 255.255.255.255
 ipv6 address 2001:0DB8:0300:0201::/64
 end
```

Examples: Migrating from a Single-Protocol IPv4-Only VRF Configuration to a Multiprotocol VRF Configuration

This section contains examples that show how to migrate from a single-protocol IPv4-only virtual routing and forwarding (VRF) configuration to a multiprotocol VRF configuration.

This example shows a single-protocol IPv4-only VRF before the VRF CLI for IPv4 and IPv6 is entered on the device:

```
ip vrf vrf1
 rd 1:1
 route-target both 1:1
interface Loopback1
 ip vrf forwarding V1
 ip address 10.3.3.3 255.255.255.255
```

This example shows how to force the migration of the single-protocol VRF vrf1 to a multiprotocol VRF configuration:

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
!
Device(config)# vrf upgrade-cli multi-af-mode common-policies vrf vrf1
You are about to upgrade to the multi-AF VRF syntax commands.
You will loose any IPv6 address configured on interfaces
belonging to upgraded VRFs.
Are you sure ? [yes]: yes
Number of VRFs upgraded: 1
Device(config)# exit
```

This example shows the multiprotocol VRF configuration after the forced migration:

```
vrf definition vrf1
 rd 1:1
 route-target both 1:1
 !
 address-family ipv4
 exit-address-family
!
interface Loopback1
 vrf forwarding V1
 ip address 10.3.3.3 255.255.255.255
```

The following is another example of a multi-AF multiprotocol VRF configuration:

```
vrf definition vrf2
 rd 100:1
 address family ipv6
 route-target both 200:1
 exit-address-family
!
ip vrf vrf1
 rd 200:1
 route-target both 200:1
!
interface Ethernet0/0
 vrf forwarding vrf2
 ip address 10.50.1.2 255.255.255.0
 ipv6 address 2001:0DB8:0:1::/64
!
interface Ethernet0/1
 ip vrf forwarding vrf1
 ip address 10.60.1.2 255.255.255.0
 ipv6 address 2001:0DB8:1 :1::/64
```

In this example, all addresses (IPv4 and IPv6) defined for interface Ethernet0/0 are in VRF vrf2. For the interface Ethernet0/1, the IPv4 address is defined in VRF vrf1 but the IPv6 address is in the global IPv6 routing table.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco Master Command List, All Releases
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 1771	*A Border Gateway Protocol 4 (BGP-4)*
RFC 4364	*BGP MPLS/IP Virtual Private Networks (VPNs)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS VPN VRF CLI for IPv4 and IPv6 VPNs

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

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

Table 9: Feature Information for MPLS VPN VRF CLI for IPv4 and IPv6 VPNs

Feature Name	Releases	Feature Information
MPLS VPN VRF CLI for IPv4 and IPv6 VPNs	12.2(33)SB 12.2(33)SRB 12.2(33)SXI 12.4(20)T Cisco IOS XE Release 3.1S	This document describes how to configure a multiprotocol Virtual Private Network (VPN) routing and forwarding (VRF) instance for IPv4 and IPv6 VPNs and describes how to upgrade your existing single-protocol IPv4-only VRF to a multiprotocol VRF configuration. The MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature introduces commands that allow you to enable an IPv4 and IPv6 VPN in the same Multiprotocol Label Switching (MPLS) VRF configuration and to simplify the migration from a single-protocol VRF configuration to a multiprotocol VRF configuration. In Cisco IOS Release 12.2(33)SB, this feature was introduced on the Cisco 10000 series router. In Cisco IOS Release 12.2(33)SRB, this feature was implemented on the Cisco 7600 series router. In Cisco IOS Release 12.2(33)SXI, this feature was integrated. In Cisco IOS Release 12.4(2)T, this feature was integrated. In Cisco IOS XE Release 3.1S, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers.
		The following commands were introduced or modified: **show vrf**, **vrf definition**, **vrf forwarding**, **vrf upgrade-cli**.

Glossary

6PE—IPv6 provider edge device or a Multiprotocol Label Switching (MPLS) label switch router (LSR) edge router using IPv6.

6VPE—IPv6 Virtual Private Network (VPN) provider edge device.

AF—address family. Set of related communication protocols in which all members use a common addressing mechanism to identify endpoints. Also called protocol family.

AFI—Address Family Identifier. Carries the identity of the network-layer protocol that is associated with the network address.

BGP—Border Gateway Protocol. A routing protocol used between autonomous systems. It is the routing protocol that makes the internet work. BGP is a distance-vector routing protocol that carries connectivity information and an additional set of BGP attributes. These attributes allow for a set of policies for deciding the best route to use to reach a given destination. BGP is defined by RFC 1771.

CE—customer edge device. A service provider device that connects to Virtual Private Network (VPN) customer sites.

FIB—Forwarding Information Base. Database that stores information about switching of data packets. A FIB is based on information in the Routing Information Base (RIB). It is the optimal set of selected routes that are installed in the line cards for forwarding.

HA—high availability. High availability is defined as the continuous operation of systems. For a system to be available, all components--including application and database servers, storage devices, and the end-to-end network--need to provide continuous service.

IP—Internet Protocol. Network-layer protocol in the TCP/IP stack offering a connectionless internetwork service. IP provides features for addressing, type-of-service specification, fragmentation and reassembly, and security.

IPv4—IP Version 4. Network layer for the TCP/IP protocol suite. IPv4 is a connectionless, best-effort packet switching protocol.

IPv6—IP Version 6. Replacement for IPv4. IPv6 is a next-generation IP protocol. IPv6 is backward compatible with and designed to fix the shortcomings of IPv4, such as data security and maximum number of user addresses. IPv6 increases the address space from 32 to 128 bits, providing for an unlimited number of networks and systems. It also supports quality of service (QoS) parameters for real-time audio and video.

MFI—MPLS Forwarding Infrastructure. In the Cisco MPLS subsystem, the data structure for storing information about incoming and outgoing labels and associated equivalent packets suitable for labeling.

MPLS—Multiprotocol Label Switching. MPLS is a method for forwarding packets (frames) through a network. It enables devices at the edge of a network to apply labels to packets (frames). ATM switches or existing devices in the network core can switch packets according to the labels with minimal lookup overhead.

PE—provider edge device. A device that is part of a service provider’s network and that is connected to a customer edge (CE) device. The PE device function is a combination of an MLS edge label switch router (LSR) function with some additional functions to support Virtual Private Networks (VPNs).

RD (IPv4)—route distinguisher. An 8-byte value that is concatenated with an IPv4 prefix to create a unique VPN IPv4 (VPNv4) prefix.

RD (IPv6)—route distinguisher. A 64-bit value that is prepended to an IPv6 prefix to create a globally unique VPN-IPv6 address.

RIB—Routing Information Base. The set of all available routes from which to choose the Forwarding Information Base (FIB). The RIB essentially contains all routes available for selection. It is the sum of all routes learned by dynamic routing protocols, all directly attached networks (that is-networks to which a given device has interfaces connected), and any additional configured routes, such as static routes.

RT—route target. Extended community attribute used to identify the Virtual Private Network (VPN) routing and forwarding (VRF) routing table into which a prefix is to be imported.

VPN—Virtual Private Network. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses “tunneling” to encrypt all information at the IP level.

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

VRF table—A routing and a forwarding table associated to a Virtual Private Network (VPN) routing and forwarding (VRF) instance. This is a customer-specific table, enabling the provider edge (PE) device to maintain independent routing states for each customer.

CHAPTER 11

MPLS VPN Route Target Rewrite

The MPLS VPN Route Target Rewrite feature allows the replacement of route targets on incoming and outgoing Border Gateway Protocol (BGP) updates. Typically, Autonomous System Border Routers (ASBRs) perform the replacement of route targets at autonomous system boundaries. Route Reflectors (RRs) and provider edge (PE) devices can also perform route target replacement.

The main advantage of the MPLS VPN Route Target Rewrite feature is that it keeps the administration of routing policy local to the autonomous system.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for MPLS VPN Route Target Rewrite

- You should know how to configure Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs).
- You need to configure your network to support interautonomous systems with different route target (RT) values in each autonomous system.

- You need to identify the RT replacement policy and target device for each autonomous system.

Restrictions for MPLS VPN Route Target Rewrite

You can apply multiple replacement rules using the route-map continue clause. The MPLS VPN Route Target Rewrite feature does not support the continue clause on outbound route maps.

Information About MPLS VPN Route Target Rewrite

Route Target Replacement Policy

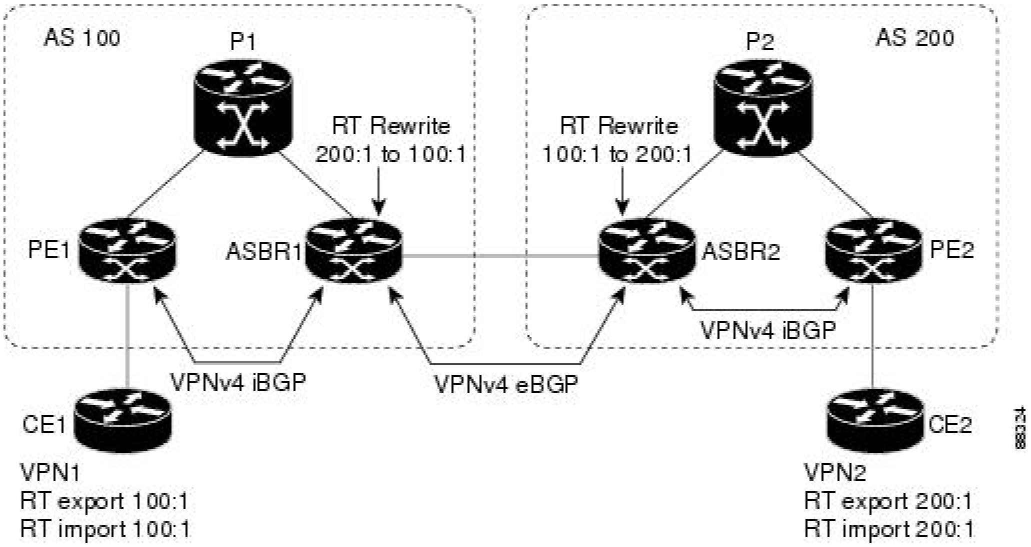
Routing policies for a peer include all configurations that may impact inbound or outbound routing table updates. The MPLS VPN Route Target Rewrite feature can influence routing table updates by allowing the replacement of route targets on inbound and outbound Border Gateway Protocol (BGP) updates. Route targets are carried as extended community attributes in BGP Virtual Private Network IP Version 4 (VPNv4) updates. Route target extended community attributes are used to identify a set of sites and VPN routing and forwarding (VRF) instances that can receive routes with a configured route target.

In general, autonomous system border routers (ASBRs) perform route target replacement at autonomous system borders when the ASBRs exchange VPNv4 prefixes. You can also configure the MPLS VPN Route Target Rewrite feature on provider edge (PE) devices and Route Reflector (RR) devices.

The figure below shows an example of route target replacement on ASBRs in an Multiprotocol Label Switching (MPLS) VPN interautonomous system topology. This example includes the following configurations:

- PE1 is configured to import and export RT 100:1 for VRF VPN1.
- PE2 is configured to import and export RT 200:1 for VRF VPN2.
- ASBR1 is configured to rewrite all inbound VPNv4 prefixes with RT 200:1 to RT 100:1.
- ASBR2 is configured to rewrite all inbound VPNv4 prefixes with RT 100:1 to RT 200:1.

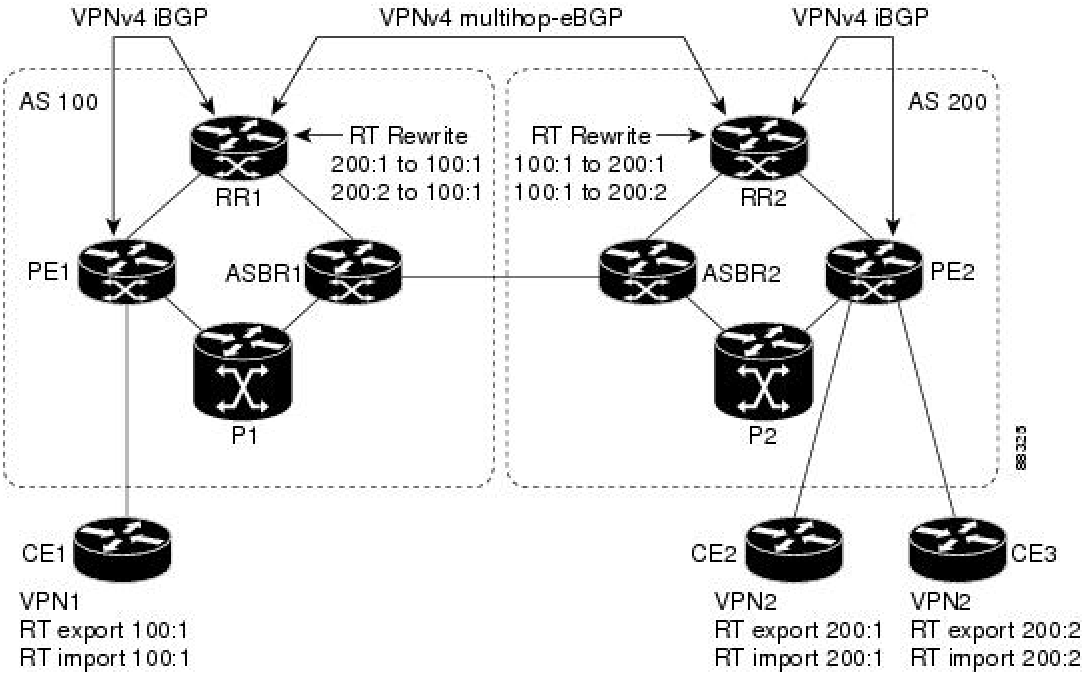
Figure 12: Route Target Replacement on ASBRs in an MPLS VPN Interautonomous System Topology



The figure below shows an example of route target replacement on route reflectors in an MPLS VPN interautonomous system topology. This example includes the following configurations:

- External BGP (EBGP) is configured on the route reflectors.
- EBGP and internal BGP (IBGP) IPv4 label exchange is configured between all BGP devices.
- Peer groups are configured on the route reflectors.
- PE2 is configured to import and export RT 200:1 for VRF VPN2.
- PE2 is configured to import and export RT 200:2 for VRF VPN3.
- PE1 is configured to import and export RT 100:1 for VRF VPN1.
- RR1 is configured to rewrite all inbound VPNv4 prefixes with RT 200:1 or RT 200:2 to RT 100:1.
- RR2 is configured to rewrite all inbound prefixes with RT 100:1 to RT 200:1 and RT 200:2.

Figure 13: Route Target Rewrite on Route Reflectors in an MPLS VPN Interautonomous System Topology



Route Maps and Route Target Replacement

The MPLS VPN Route Target Rewrite feature extends the Border Gateway Protocol (BGP) inbound/outbound route map functionality to enable route target replacement. The **set extcomm-list delete** command entered in route-map configuration mode allows the deletion of a route target extended community attribute based on an extended community list.

How to Configure MPLS VPN Route Target Rewrite

Configuring a Route Target Replacement Policy

Perform this task to configure a route target (RT) replacement policy for your internetwork.

If you configure a provider edge (PE) device to rewrite RT *x* to RT *y* and the PE has a virtual routing and forwarding (VRF) instance that imports RT *x* , you need to configure the VRF to import RT *y* in addition to RT *x* .

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip extcommunity-list** {*standard-list-number* | *expanded-list-number*} {**permit** | **deny**} [*regular-expression*] [**rt** | **soo** *extended-community-value*]
4. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]

5. **match extcommunity** {*standard-list-number* | *expanded-list-number*}
6. **set extcomm-list** *extended-community-list-number* **delete**
7. **set extcommunity** {**rt** *extended-community-value* [**additive**] | **soo** *extended-community-value*}
8. **end**
9. **show route-map** *map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip extcommunity-list** {*standard-list-number* | *expanded-list-number*} {**permit** | **deny**} [*regular-expression*] [**rt** | **soo** *extended-community-value*] **Example:** `Device(config)# ip extcommunity-list 1 permit rt 100:3`	Creates an extended community access list and controls access to it. • The *standard-list-number* argument is an integer from 1 to 99 that identifies one or more permit or deny groups of extended communities. • The *expanded-list-number* argument is an integer from 100 to 500 that identifies one or more permit or deny groups of extended communities. Regular expressions can be configured with expanded lists but not standard lists. • The **permit** keyword permits access for a matching condition. • The **deny** keyword denies access for a matching condition. • The *regular-expression* argument specifies an input string pattern to match against. When you use an expanded extended community list to match route targets, include the pattern RT: in the regular expression. • The **rt** keyword specifies the route target extended community attribute. The **rt** keyword can be configured only with standard extended community lists and not expanded community lists. • The **soo** keyword specifies the site of origin (SOO) extended community attribute. The **soo** keyword can

	Command or Action	Purpose
		be configured only with standard extended community lists and not expanded community lists. • The *extended-community-value* argument specifies the route target or site of origin. The value can be one of the following combinations: • autonomous-system-number:network-number • ip-address:network-number The colon is used to separate the autonomous system number and network number or IP address and network number.
Step 4	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map extmap permit 10`	Defines the conditions for redistributing routes from one routing protocol into another or enables policy routing and enables route-map configuration mode. • The *map-name* argument defines a meaningful name for the route map. The **redistribute** router configuration command uses this name to reference this route map. Multiple route maps can share the same map name. • If the match criteria are met for this route map, and the **permit** keyword is specified, the route is redistributed as controlled by the set actions. In the case of policy routing, the packet is policy routed. If the match criteria are not met, and the **permit** keyword is specified, the next route map with the same map tag is tested. If a route passes none of the match criteria for the set of route maps sharing the same name, it is not redistributed by that set. The **permit** keyword is the default. • If the match criteria are met for the route map and the **deny** keyword is specified, the route is not redistributed. In the case of policy routing, the packet is not policy routed, and no further route maps sharing the same map tag name will be examined. If the packet is not policy routed, the normal forwarding algorithm is used. • The *sequence-number* argument is a number that indicates the position a new route map will have in the list of route maps already configured with the same name. If given with the **no** form of this command, the position of the route map should be deleted.
Step 5	**match extcommunity** {*standard-list-number* | *expanded-list-number*}	Matches the Border Gateway Protocol (BGP) extended community list attributes.

	Command or Action	Purpose

	Example:
	`Device(config-route-map)# match extcommunity 1`
	Example:
	`Device(config-route-map)# match extcommunity 101`
		• The *standard-list-number* argument is a number from 1 to 99 that identifies one or more permit or deny groups of extended community attributes.
		• The *expanded-list-number* argument is a number from 100 to 500 that identifies one or more permit or deny groups of extended community attributes.

Step 6	**set extcomm-list** *extended-community-list-number* **delete** **Example:** `Device(config-route-map)# set extcomm-list 1 delete`	Removes a route target from an extended community attribute of an inbound or outbound BGP Virtual Private Network Version 4 (VPNv4) update. • The *extended-community-list-number* argument specifies the extended community list number.

Step 7	**set extcommunity** {**rt** *extended-community-value* [**additive**] | **soo** *extended-community-value*}
	Example:
	`Device(config-route-map)# set extcommunity rt 100:4 additive`
		Sets BGP extended community attributes.
		• The **rt** keyword specifies the route target extended community attribute.
		• The **soo** keyword specifies the site of origin extended community attribute.
		• The *extended-community-value* argument specifies the value to be set. The value can be one of the following combinations:
		  • autonomous-system-number : network-number
		  • ip-address : network-number
		The colon is used to separate the autonomous system number and network number or IP address and network number.
		• The **additive** keyword adds a route target to the existing route target list without replacing any existing route targets.

Step 8	**end**
	Example:
	`Device(config-route-map)# end`
		(Optional) Returns to privileged EXEC mode.

Step 9	**show route-map** *map-name* **Example:** `Device# show route-map extmap`	(Optional) Verifies that the match and set entries are correct. • The *map-name* argument is the name of a specific route map.

Applying the Route Target Replacement Policy

Perform the following tasks to apply the route target replacement policy to your internetwork:

Associating Route Maps with Specific BGP Neighbors

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
5. **address-family vpnv4** [**unicast**]
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **extended** | **standard**]
8. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 100`	Configures a Border Gateway Protocol (BGP) routing process and places the device in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP devices and tags the routing information passed along. The range is 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:** `Device(config-router)# neighbor 172.10.0.2 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.

	Command or Action	Purpose
		- 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 vpnv4** [**unicast**] **Example:** `Device(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. - The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 172.16.0.2 activate`	Enables the exchange of information with a neighboring BGP device. - 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* | *peer-group-name*} **send-community** [**both** | **extended** | **standard**] **Example:** `Device(config-router-af)# neighbor 172.16.0.2 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. - The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. - The *peer-group-name* argument specifies the name of a BGP peer group. - The **both** keyword sends standard and extended community attributes. - The **extended** keyword sends an extended community attribute. - The **standard** keyword sends a standard community attribute.

Step 8	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
	Example:
	`Device(config-router-af)# neighbor 172.16.0.2 route-map extmap in`
		Apply a route map to incoming or outgoing routes
		- The *ip-address* argument specifies the IP address of the neighbor.
		- The *peer-group-name* argument specifies the name of a BGP or multiprotocol peer group.
		- The *map-name* argument specifies the name of a route map.
		- The **in** keyword applies route map to incoming routes.
		- The **out** keyword applies route map to outgoing routes.

	Command or Action	Purpose

Step 9	**end**
	Example:
	`Device(config-router-af)# end`
		(Optional) Returns to privileged EXEC mode.

Refreshing BGP Session to Apply Route Target Replacement Policy

After you have defined two devices to be Border Gateway Protocol (BGP) neighbors, the devices form a BGP connection and exchange routing information. If you subsequently change a routing policy, you must reset BGP connections for the configuration change to take effect. After configuring the route target (RT) replacement policy and applying it to the target devices in your system, you must refresh the BGP session to put the policy into operation.

SUMMARY STEPS

1. **enable**
2. **clear ip bgp** {***** | *neighbor-address* | *peer-group-name* [**soft** [**in** | **out**]} [**ipv4** {**multicast** | **unicast**} | **vpnv4 unicast** {**soft** | {**in** | **out**}]
3. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip bgp** {***** | *neighbor-address* | *peer-group-name* [**soft** [**in** | **out**]} [**ipv4** {**multicast** | **unicast**} | **vpnv4 unicast** {**soft** | {**in** | **out**}] **Example:** `Device# clear ip bgp vpnv4 unicast 172.16.0.2 in`	Resets a BGP connection using BGP soft reconfiguration. • The ***** keyword resets all current BGP sessions. • The *neighbor-address* argument resets only the identified BGP neighbor. • The *peer-group-name* argument resets the specified BGP peer group. • The **ipv4** keyword resets the specified IPv4 address family neighbor or peer group. The **multicast** or **unicast** keyword must be specified. • The **vpnv4** keyword resets the specified Virtual Private Network Version 4 (VPNv4) address family neighbor or peer group. The **unicast** keyword must be specified. • The **soft** keyword indicates a soft reset. Does not reset the session. The **in** or **out** keywords do not follow the **soft** keyword when a connection is cleared under the

	Command or Action	Purpose
		VPNv4 or IPv4 address family because the **soft** keyword specifies both. • The **in** and **out** keywords trigger inbound or outbound soft reconfiguration, respectively. If the **in** or **out** keyword is not specified, both inbound and outbound soft reset are triggered.
Step 3	**disable** **Example:** `Device# disable`	(Optional) Returns to user EXEC mode.

Troubleshooting Tips

To determine whether a BGP device supports the route refresh capability, use the **show ip bgp neighbors** command. If a device supports the route refresh capability, the following message is displayed:

```
Received route refresh capability from peer.
```

You can issue the **debug ip bgp updates** command on the device where you entered the **clear ip bgp** command to verify that the updates are occurring.

Note Issuing the **debug ip bgp updates** command could impair performance if the device sends or receives a large number of Border Gateway Protocol (BGP) updates.

Verifying the Route Target Replacement Policy

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4 all** *network-address*
3. **exit**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device> enable
Device#
```

Step 2 **show ip bgp vpnv4 all** *network-address*

Verifies that all Virtual Private Network Versio9n 4 (VPNv4) prefixes with a specified route target (RT) extended community attribute are replaced with the proper RT extended community attribute at the autonomous system border routers (ASBRs) or route reflectors and to verify that the provider edge (PE) devices receive the rewritten RT extended community attributes from the ASBRs or route reflectors. The following examples verify route target replacement on ABSR1 and ABSR2.

Verify route target replacement on ABSR1:

Example:

```
Device# show ip bgp vpnv4 all 172.16.17.17
BGP routing table entry for 100:1:172.16.17.17/32, version 6
Paths: (1 available, best #1, no table)
  Advertised to update-groups:
     1
  300
    172.16.11.11 (metric 589) from 172.16.11.11 (172.16.11.11)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:200:1
```

Verify route target replacement on ABSR2:

Example:

```
Device# show ip bgp vpnv4 all 172.16.17.17
BGP routing table entry for 100:1:172.16.17.17/32, version 6
Paths: (1 available, best #1, no table)
  Advertised to update-groups:
     1
  100 300
    192.168.1.1 from 192.168.1.1 (172.16.13.13)
      Origin incomplete, localpref 100, valid, external, best
      Extended Community: RT:100:1
```

The following examples verify route target replacement on PE1 and PE2.

Verify route target on PE1:

Example:

```
Device# show ip bgp vpnv4 all 172.16.17.17
BGP routing table entry for 100:1:172.16.17.17/32, version 13
Paths: (1 available, best #1, table vpn1)
  Advertised to update-groups:
     1
  300
    192.168.2.1 (via vpn1) from 192.168.2.1 (172.16.19.19)
      Origin incomplete, metric 0, localpref 100, valid, external, best
      Extended Community: RT:200:1
```

Verify route target on PE2:

Example:

```
Device# show ip bgp vpnv4 all 172.16.17.17
BGP routing table entry for 100:1:172.16.17.17/32, version 13
Paths: (1 available, best #1, table vpn1)
  Advertised to update-groups:
     3
  100 300
    192.168.1.1 (metric 20) from 172.16.16.16 (172.16.16.16)
```

```
      Origin incomplete, localpref 100, valid, internal, best
      Extended Community: RT:100:1
```

Step 3 **exit**

Returns to user EXEC mode:

Example:

```
Device# exit
Device>
```

Troubleshooting Your Route Target Replacement Policy

SUMMARY STEPS

1. **enable**
2. **debug ip bgp updates**
3. **show ip bgp vpnv4 all** *network-address*
4. **exit**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device> enable
Device#
```

Step 2 **debug ip bgp updates**

Verifies that the Border Gateway Protocol (BGP) updates are occurring on the autonomous system border router (ASBR). The ASBR in this example has the IP address 172.16.16.16.

Example:

```
Device# debug ip bgp updates
BGP(2): no valid path for 100:1:172.16.20.20/32
BGP(2): no valid path for 100:1:10.0.0.0/8
%BGP-5-ADJCHANGE: neighbor 172.16.16.16 Down User reset
BGP(2): nettable_walker 100:1:172.16.20.20/32 no RIB
BGP(2): nettable_walker 100:1:192.168.3.0/8 no RIB
BGP(2): 172.16.11.11 computing updates, afi 2, neighbor version 13,
table version 15, starting at 0.0.0.0
BGP(2): 172.16.11.11 send unreachable 100:1:172.16.20.20/32
BGP(2): 172.16.11.11 send UPDATE 100:1:172.16.20.20/32 -- unreachable
BGP(2): 172.16.11.11 send UPDATE 100:1:192.168.3.0/8 -- unreachable
BGP(2): 1 updates (average = 58, maximum = 58)
BGP(2): 172.16.11.11 updates replicated for neighbors: 172.16.11.11
BGP(2): 172.16.11.11 update run completed, afi 2, ran for 0ms,
neighbor version 15, start version 15, throttled to 15
```

```
BGP: Import walker start version 13, end version 15
BGP: ... start import cfg version = 30
%BGP-5-ADJCHANGE: neighbor 172.16.16.16 Up
BGP(2): 172.16.16.16 computing updates, afi 2, neighbor version 0,
table version 15, starting at 0.0.0.0
BGP(2): 172.16.16.16 send UPDATE (format) 100:1:172.16.0.0/16,
next 172.16.11.11, metric 0, path 300, extended community RT:2:2
RT:7777:222222222 RT:20000:111 RT:65535:999999999
BGP(2): 172.16.16.16 send UPDATE (prepend, chgflags: 0x0)
100:1:172.16.19.19/32, next 172.16.11.11, metric 0, path 300,
extended community RT:2:2 RT:7777:222222222 RT:20000:111
RT:65535:999999999
BGP(2): 172.16.16.16 send UPDATE (format) 100:1:192.168.2.0/8,
next 172.16.11.11, metric 0, path , extended community
RT:2:2 RT:7777:222222222 RT:20000:111 RT:65535:999999999
BGP(2): 2 updates (average = 111, maximum = 121)
BGP(2): 172.16.16.16 updates replicated for neighbors: 172.16.16.16
BGP(2): 172.16.16.16 update run completed, afi 2, ran for 0ms,
neighbor version 15, start version 15, throttled to 15
BGP(2): 172.16.16.16 rcvd UPDATE w/ attr: nexthop 172.16.15.15,
origin ?, path 200, extended community RT:100:1
BGP(2): 172.16.16.16 rcvd 100:1:192.168.3.0/8
BGP(2): 172.16.16.16 rcvd UPDATE w/ attr: nexthop 172.16.15.15,
origin ?, path 200 400, extended community RT:100:1
BGP(2): 172.16.16.16 rcvd 100:1:172.16.0.0/16
BGP(2): 172.16.16.16 rcvd 100:1:172.16.20.20/32
BGP(2): nettable_walker 100:1:172.16.20.20/32 no RIB
BGP(2): nettable_walker 100:1:192.168.3.0/8 no RIB
BGP: Import walker start version 15, end version 17
BGP: ... start import cfg version = 30
BGP(2): 172.16.11.11 computing updates, afi 2,
neighbor version 15, table version 17,
starting at 0.0.0.0
BGP(2): 172.16.11.11 NEXT_HOP part 1 net 100:1:172.16.20.20/32,
next 172.16.15.15
BGP(2): 172.16.11.11 send UPDATE (format) 100:1:172.16.20.20/32,
next 172.16.15.15,metric 0, path 200 400, extended community
RT:1:1 RT:10000:111 RT:33333:888888888
RT:65535:999999999
BGP(2): 172.16.11.11 NEXT_HOP part 1 net 100:1:10.0.0.0/8,
next 172.16.15.15
BGP(2): 172.16.11.11 send UPDATE (format) 100:1:192.168.3.0/8,
next 172.16.15.15, metric 0, path 200, extended community
RT:1:1 RT:10000:111 RT:33333:888888888 RT:65535:999999999
BGP(2): 2 updates (average = 118, maximum = 121)
BGP(2): 172.16.11.11 updates replicated for neighbors: 172.16.11.11
BGP(2): 172.16.11.11 update run completed, afi 2, ran for 0ms,
neighbor version 17, start version 17, throttled to 17
```

You can also reset the BGP connection by using the **clear ip bgp *** command and enter the **debug ip bgp updates** command again to verify that BGP updates are occurring as shown in the output after the **clear ip bgp** command is entered.

Step 3 **show ip bgp vpnv4 all** *network-address*

Verifies that route target (RT) extended community attributes are replaced correctly.

Example:

```
Device# show ip bgp vpnv4 all 172.16.17.17
BGP routing table entry for 100:1:172.16.17.17/32, version 6
Paths: (1 available, best #1, no table)
  Advertised to update-groups:
     1
  100 300
    192.168.1.1 from 192.168.1.1 (172.16.13.13)
      Origin incomplete, localpref 100, valid, external, best
      Extended Community: RT:100:1
```

This example shows Virtual Private Network (VPN) address information from the BGP table and verifies that RT extended community attributes are replaced correctly.

Step 4 **exit**

Returns to user EXEC mode:

Example:

```
Device# exit
Device>
```

Configuration Examples for MPLS VPN Route Target Rewrite

Examples: Configuring Route Target Replacement Policies

This example shows the route target (RT) replacement configuration of an autonomous system border router (ASBR1) that exchanges Virtual Private Network Version 4 (VPNv4) prefixes with another ASBR (ASBR2). The route map extmap is configured to replace RTs on inbound updates. Any incoming update with RT 100:3 is replaced with RT 200:3. Any other prefixes with an RT whose autonomous system number is 100 is rewritten to RT 200:4.

```
!
ip extcommunity-list 1 permit rt 100:3
ip extcommunity-list 101 permit RT:100:*
!
route-map extmap permit 10
match extcommunity 1
set extcomm-list 1 delete
set extcommunity rt 200:3 additive
!
route-map regexp permit 10
match extcommunity 101
set extcomm-list 101 delete
set extcommunity rt 200:4 additive
!
route-map regexp permit 20
```

This example shows the use of the route-map configuration **continue** command when you need to apply more than one replacement rule on an update. In this example, an incoming update with RT 100:3 is replaced with RT 200:3. Without the **continue 20** command, route-map evaluation would stop when a match on sequence 10 is made. With the **continue 20** command, route-map evaluation continues into sequence 20 even if a match occurs in sequence 10. If the incoming update has an RT 100:4, the device replaces it with RT 200:4.

```
!
ip extcommunity-list 1 permit rt 100:3
ip extcommunity-list 2 permit rt 100:4
```

```
!
route-map extmap permit 10
match extcommunity 1
set extcomm-list 1 delete
set extcommunity rt 200:3 additive
continue 20
!
route-map extmap permit 20
match extcommunity 2
set extcomm-list 2 delete
set extcommunity rt 200:4 additive
!
route-map extmap permit 30
```

Note The route-map configuration **continue** command is not supported on outbound route maps.

Examples: Applying Route Target Replacement Policies

This section contains the following examples:

Examples: Associating Route Maps with Specific BGP Neighbor

This example shows the association of route map extmap with a Border Gateway Protocol (BGP) neighbor. The BGP inbound route map is configured to replace route targets (RTs) on incoming updates.

```
router bgp 100
.
.
.
 neighbor 172.16.0.2 remote-as 100
.
.
.
!
address family vpnv4
 neighbor 172.16.0.2 activate
 neighbor 172.16.0.2 send-community extended
 neighbor 172.16.0.2 route-map extmap in
```

This example shows the association of the same route map with the outbound BGP neighbor. The route map is configured to replace RTs on outgoing updates.

```
router bgp 100
.
.
.
 neighbor 172.16.0.2 remote-as 100
.
.
.
!
address family vpnv4
 neighbor 172.16.0.2 activate
 neighbor 172.16.0.2 send-community extended
 neighbor 172.16.0.2 route-map extmap out
```

Example: Refreshing the BGP Session to Apply the Route Target Replacement Policy

The following example shows the **clear ip bgp** command used to initiate a dynamic reconfiguration in the Border Gateway Protocol (BGP) peer 172.16.0.2. This command requires that the peer supports the route refresh capability.

```
Device# clear ip bgp 172.16.0.2 vpnv4 unicast in
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco Master Command List, All Releases
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference
MPLS, MPLS VPN, and MPLS VPN interautonomous systems configuration tasks	*MPLS Layer 3 Inter-AS and CSC Configuration Guide*
BGP configuration tasks	*IP Routing: BGP Configuration Guide*
Commands to configure and monitor BGP	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS VPN Route Target Rewrite

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 10: Feature Information for MPLS VPN Route Target Rewrite

Feature Name	Releases	Feature Information
MPLS VPN Route Target Rewrite	12.4(20)T	The MPLS VPN Route Target Rewrite feature allows the replacement of route targets on incoming and outgoing Border Gateway Protocol (BGP) updates. Typically, Autonomous System Border Routers (ASBRs) perform the replacement of route targets at autonomous system boundaries. Route Reflectors (RRs) and provider edge (PE) devices can also perform route target replacement. The main advantage of the MPLS VPN Route Target Rewrite feature is that it keeps the administration of routing policy local to the autonomous system. In Cisco IOS Release 12.4(20)T, this feature was integrated. The following command was modified: **set extcomm-list delete.**

Table 11: Feature Information for MPLS VPN Route Target Rewrite

Feature Name	Releases	Feature Information
MPLS VPN Route Target Rewrite	12.0(26)S 12.2(25)S 12.2(33)SRA 12.2(33)SXH 12.4(20)T	The MPLS VPN Route Target Rewrite feature allows the replacement of route targets on incoming and outgoing Border Gateway Protocol (BGP) updates. Typically, Autonomous System Border Routers (ASBRs) perform the replacement of route targets at autonomous system boundaries. Route Reflectors (RRs) and provider edge (PE) devices can also perform route target replacement. The main advantage of the MPLS VPN Route Target Rewrite feature is that it keeps the administration of routing policy local to the autonomous system. In Cisco IOS Release 12.0(26)S, this feature was introduced for the Cisco 7200, 7500, and 12000 series routers. In Cisco IOS Release 12.2(25)S, this feature was integrated to support Cisco 7500 series routers. In Cisco IOS Release 12.2(33)SRA, this feature was integrated. In Cisco IOS Release 12.2(33)SXH, this feature was integrated to support the Catalyst 6500 series routers. In Cisco IOS Release 12.4(20)T, this feature was integrated. The following command was modified: **set extcomm-list delete**.

Glossary

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

ASBR—autonomous system border router. A device 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 devices 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 device—customer edge device. The customer device that connects to the provider edge (PE) device.

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

IBGP—Internal Border Gateway Protocol. A BGP session between devices 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 Internal Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).

LDP—Label Distribution Protocol. A standard protocol between MPLS-enabled devices to negotiate the labels (addresses) used to forward packets. The Cisco proprietary version of this protocol is the Tag Distribution Protocol (TDP).

LER—label edge router. The edge device 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.

MPLS—Multiprotocol Label Switching. A switching method that forwards IP traffic using a label. This label instructs the devices and the switches in the network where to forward the packets based on preestablished IP routing information.

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 device—provider device. The core device in the service provider network that connects to provider edge (PE) devices. In a packet-switched star topology, a device 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 device—provider edge device. The label edge router (LER) in the service provider network that connects to the customer edge (CE) device.

RD—route distinguisher. An 8-byte value that is concatenated with an IPv4 prefix to create a unique VPN IPv4 (VPNv4) prefix.

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

RT—route target. Extended community attribute used to identify the VRF routing table into which a prefix is to be imported.

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 prefix—IPv4 prefix preceded by an 8-byte route distinguisher. The VPN addresses are made unique by adding a route distinguisher to the front of the address.

VRF—VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a provider edge (PE) device.

CHAPTER 12

MPLS VPN Per VRF Label

The MPLS VPN Per VRF Label feature allows you to configure a single Virtual Private Network (VPN) label for all local routes in the entire VPN routing and forwarding (VRF) domain. This MPLS VPN Per VRF Label feature incorporates a single (per VRF) VPN label that for all local routes in the VRF table.

You can enable (or disable) the MPLS VPN Per VRF Label feature in global configuration mode.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for MPLS VPN Per VRF Label

- If your virtual routing and forwarding (VRF) domain has the external/internal Border Gateway Protocol (EIBGP) multipath feature or the Carrier Supporting Carrier (CSC) feature enabled, disable those features before you configure the MPLS VPN Per VRF Label feature.
- Before configuring Multiprotocol Label Switching (MPLS) Layer 3 Virtual Private Networks (VPNs), you must install MPLS, Label Distribution Protocol (LDP), and Cisco Express Forwarding in your network. All devices in the core, including the provider edge (PE) devices, must be able to support Cisco Express Forwarding and MPLS forwarding.

Restrictions for MPLS VPN Per VRF Label

- Enabling the MPLS VPN Per VRF Label feature causes Border Gateway Protocol (BGP) reconvergence, which can result in data loss for traffic coming from the Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) core.

Note You can minimize network disruption by enabling this feature during a scheduled MPLS maintenance window. Also, if possible, avoid enabling this feature on a live device

- There is no performance degradation when you configure up to 511 VRFs; however, when you add more than 511 VRFs, your network might experience some minor performance degradation (similar to the normal degradation experienced by any of the directly connected VRF prefixes present in the device).
- Per-prefix MPLS counters for VPN prefixes are lost when you enable the MPLS VPN Per VRF Label feature.
- You cannot use this feature with Carrier Supporting Carrier (CSC) and external/internal Border Gateway Protocol (EIBGP) multipath features.

Information About MPLS VPN Per VRF Label

MPLS VPN Per VRF Label Functionality

The provider edge (PE) stores both local and remote routes and includes a label entry for each route. For distributed platforms, the per-prefix labels consume memory. When there are many virtual routing and forwarding (VRF) domains and routes, the amount of memory that the per-prefix labels consume can become an issue.

The MPLS VPN Per VRF Label feature allows the advertisement of a single Virtual Private Network (VPN) label for local routes throughout the entire VRF. The device uses a new VPN label for the VRF decoding and IP-based lookup to learn where to forward packets for the PE or customer edge (CE) interfaces.

The following conditions apply when you configure the Per VRF Label feature:

- The VRF uses one label for all local routes.
- When you *enable* the MPLS VPN Per VRF Label feature, any existing Per VRF Aggregate label is used. If no Per VRF Aggregate label is present, the software creates a new Per VRF label.
- When you *enable* the MPLS VPN Per VRF Label feature, the CE device's learned local routes will experience some data loss.

The CE does not lose data when you disable the MPLS VPN Per VRF Label feature because when you disable the feature, the configuration reverts to the default labeling configuration, which uses the Per VRF Aggregate label from the local nonCE-sourced routes.

- When you *disable* the MPLS VPN Per VRF Label feature, the configuration reverts to the default configuration.

- A Per VRF label forwarding entry is deleted only if the VRF or the Border Gateway Protocol (BGP) configuration is removed.

Summarization of Label Allocation Modes

The table below defines the label allocations used with various route types.

Table 12: Label Allocation Modes

Route Types	Label Mode Default	Label Mode: Per VRF Label Feature
Local to the PE (connected, static route to NULL0, BGP aggregates), redistributed to BGP	Per VRF Aggregate label	Per VRF label
Locally learned from CE (through EBGP or other PE or CE protocols)	Per Prefix label	Per VRF label

How to Configure MPLS VPN Per VRF Label

Configuring the Per VRF Label Feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label mode** {**vrf** *vrf-name* | **all-vrfs**} **protocol bgp-vpnv4** {**per-prefix** | **per-vrf**}
4. **end**
5. **show ip vrf detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls label mode** {**vrf** *vrf-name* | **all-vrfs**} **protocol bgp-vpnv4** {**per-prefix** | **per-vrf**} **Example:**	Configures the MPLS VPN Per VRF Label feature.

	Command or Action	Purpose
	`Device(config)# mpls label mode all-vrfs protocol bgp-vpnv4 per-vrf`	

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 5	**show ip vrf detail** **Example:** `Device# show ip vrf detail`	Displays the VRF label mode.

Examples

The following command example shows how to verify the MPLS VPN Per VRF Label configuration:

In this example output, the **bold** text indicates the label modes:

```
Device# show ip vrf detail
VRF vpn1; default RD 1:1; default VPNID <not set>
VRF Table ID = 1
  Interfaces:
    Ethernet0/0              Serial5/0                Loopback1
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:1:1
  Import VPN route-target communities
    RT:1:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-vrf (Label 19)
VRF vpn2; default RD 2:1; default VPNID <not set>
VRF Table ID = 2
  Interfaces:
    Ethernet2/0              Loopback2
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:2:1
  Import VPN route-target communities
    RT:2:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-vrf (Label 20)
VRF vpn3; default RD 3:1; default VPNID <not set>
VRF Table ID = 3
  Interfaces:
    Ethernet3/0              Loopback3
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:3:1
  Import VPN route-target communities
    RT:3:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-vrf (Label 23)
Device# show ip bgp vpnv4 all labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 1:1 (vpn1)
   127.0.0.1/32     192.168.1.1     IPv4 VRF Aggr:19/nolabel
   127.0.0.5/32     127.0.0.4       nolabel/19
   192.168.1.0/24   192.168.1.1     IPv4 VRF Aggr:19/nolabel
                    0.0.0.0         IPv4 VRF Aggr:19/aggregate(vpn1)
   192.168.4.0/24   127.0.0.4       nolabel/20
   172.16.0.0/16    0.0.0.0         IPv4 VRF Aggr:19/aggregate(vpn1)
   172.16.128.0/32  192.168.1.1     IPv4 VRF Aggr:19/nolabel
Route Distinguisher: 2:1 (vpn2)
   127.0.2.2/32     0.0.0.0         IPv4 VRF Aggr:20/aggregate(vpn2)
   127.0.0.6/32     192.168.5.1     IPv4 VRF Aggr:20/nolabel
   192.168.5.0/24   0.0.0.0         IPv4 VRF Aggr:20/aggregate(vpn2)
   172.17.128.0/32  192.168.5.1     IPv4 VRF Aggr:20/nolabel
Route Distinguisher: 3:1 (vpn3)
   127.0.3.2/32     0.0.0.0         IPv4 VRF Aggr:23/aggregate(vpn3)
   127.0.0.8/32     192.168.7.1     IPv4 VRF Aggr:23/nolabel
   192.168.7.0/24   0.0.0.0         IPv4 VRF Aggr:23/aggregate(vpn3)
   172.16.128.0/32  192.168.7.1     IPv4 VRF Aggr:23/nolabel
Device# show mpls forwarding-table

Local  Outgoing    Prefix              Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
16     Pop tag     192.168.3.0/24    0          Et1/0      192.168.2.3
17     Pop tag     127.0.0.3/32      0          Et1/0      192.168.2.3
18     17          127.0.0.4/32      0          Et1/0      192.168.2.3
19     Pop Label   IPv4 VRF[V]       0          aggregate/vpn1
20     Pop Label   IPv4 VRF[V]       0          aggregate/vpn2
23     Pop Label   IPv4 VRF[V]       0          aggregate/vpn3
PE1#
```

Configuration Examples for MPLS VPN Per VRF Label

Example: No Label Mode Default Configuration

The following example shows the default label mode configuration (no label mode).

In this example output, the **bold** text indicates the label modes:

```
Device# show ip vrf detail
VRF vpn1; default RD 1:1; default VPNID <not set>
VRF Table ID = 1
  Interfaces:
    Ethernet0/0              Serial5/0                Loopback1
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:1:1
  Import VPN route-target communities
    RT:1:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-prefix
    per-vrf-aggr for connected and BGP aggregates (Label 19)
```

```
VRF vpn2; default RD 2:1; default VPNID <not set>
VRF Table ID = 2
  Interfaces:
    Ethernet2/0              Loopback2
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:2:1
  Import VPN route-target communities
    RT:2:1
  No import route-map
  No export route-map
CSC is not configured.

VRF label allocation mode: per-prefix

    per-vrf-aggr for connected and BGP aggregates (Label 20)
VRF vpn3; default RD 3:1; default VPNID <not set>
VRF Table ID = 3
  Interfaces:
    Ethernet3/0              Loopback3
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:3:1
  Import VPN route-target communities
    RT:3:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-prefix
    per-vrf-aggr for connected and BGP aggregates (Label 23)
Device# show ip bgp vpnv4 all labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 1:1 (vpn1)
   127.0.0.1/32     192.168.1.1     27/nolabel
   127.0.0.5/32     127.0.0.4       nolabel/19
   192.168.1.0/24   192.168.1.1     IPv4 VRF Aggr:19/nolabel
                    0.0.0.0         IPv4 VRF Aggr:19/aggregate(vpn1)
   192.168.4.0/24   127.0.0.4       nolabel/20
   172.16.0.0/16    0.0.0.0         IPv4 VRF Aggr:19/aggregate(vpn1)
   172.16.128.0/32  192.168.1.1     28/nolabel
Route Distinguisher: 2:1 (vpn2)
   127.0.2.2/32     0.0.0.0         IPv4 VRF Aggr:20/aggregate(vpn2)
   127.0.0.6/32     192.168.5.1     21/nolabel
   192.168.5.0/24   0.0.0.0         IPv4 VRF Aggr:20/aggregate(vpn2)
   172.17.128.0/32  192.168.5.1     22/nolabel
Route Distinguisher: 3:1 (vpn3)
   127.0.3.2/32     0.0.0.0         IPv4 VRF Aggr:23/aggregate(vpn3)
   127.0.0.8/32     192.168.7.1     24/nolabel
   192.168.7.0/24   0.0.0.0         IPv4 VRF Aggr:23/aggregate(vpn3)
   172.16.128.0/32  192.168.7.1     25/nolabel
Device# show mpls forwarding-table
Local  Outgoing    Prefix              Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
16     Pop tag     192.168.3.0/24    0          Et1/0      192.168.2.3
17     Pop tag     127.0.0.3/32      0          Et1/0      192.168.2.3
18     17          127.0.0.4/32      0          Et1/0      192.168.2.3
19     Pop Label   IPv4 VRF[V]       0          aggregate/vpn1
20     Pop Label   IPv4 VRF[V]       0          aggregate/vpn2
21     Untagged    127.0.0.6/32[V]   0          Et2/0      192.168.5.1
22     Untagged    172.17.128.0/32[V]0          Et2/0      192.168.5.1
23     Pop Label   IPv4 VRF[V]       0          aggregate/vpn3
24     Untagged    127.0.0.8/32[V]   0          Et3/0      192.168.7.1
25     Untagged    172.16.128.0/32[V]0          Et3/0      192.168.7.1
```

```
27     Untagged    127.0.0.1/32[V]   0          Et0/0      192.168.1.1
28     Untagged    172.16.128.0/32[V]0          Et0/0      192.168.1.1
```

Example: Mixed Mode with Global Per-Prefix

For this example, the following commands set VPN 1 for per-vrf label mode, VPN 2 for per-prefix label mode, and all remaining VPNs for per-prefix (globally).

In this example output, the **bold** text indicates the label modes:

```
Device# mpls label mode vrf vpn1 protocol bgp-vpnv4 per-vrf
Device# mpls label mode vrf vpn2 protocol bgp-vpnv4 per-prefix
```

Use the following show commands to display the label mode settings:

```
Device# show ip vrf detail
VRF vpn1; default RD 1:1; default VPNID <not set>
VRF Table ID = 1
  Interfaces:
    Ethernet0/0              Serial5/0                Loopback1
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:1:1
  Import VPN route-target communities
    RT:1:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-vrf (Label 26)
VRF vpn2; default RD 2:1; default VPNID <not set>
VRF Table ID = 2
  Interfaces:
    Ethernet2/0              Loopback2
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:2:1
  Import VPN route-target communities
    RT:2:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-prefix
    per-vrf-aggr for connected and BGP aggregates (Label 27)
VRF vpn3; default RD 3:1; default VPNID <not set>
VRF Table ID = 3
  Interfaces:
    Ethernet3/0              Loopback3
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:3:1
  Import VPN route-target communities
    RT:3:1
  No import route-map
  No export route-map
CSC is not configured.

VRF label allocation mode: per-prefix
    per-vrf-aggr for connected and BGP aggregates (Label 28)
Device# show ip bgp vpnv4 all label
   Network          Next Hop      In label/Out label
Route Distinguisher: 1:1 (vpn1)
```

```
   127.0.0.1/32     192.168.1.1     IPv4 VRF Aggr:26/nolabel
   127.0.0.5/32     127.0.0.4       nolabel/19
   192.168.1.0/24   0.0.0.0         IPv4 VRF Aggr:26/aggregate(vpn1)
                    192.168.1.1     IPv4 VRF Aggr:26/nolabel
   192.168.4.0/24   127.0.0.4       nolabel/20
   172.16.0.0/16    0.0.0.0         IPv4 VRF Aggr:26/aggregate(vpn1)
   172.16.128.0/32  192.168.1.1     IPv4 VRF Aggr:26/nolabel
Route Distinguisher: 2:1 (vpn2)
   127.0.2.2/32     0.0.0.0         IPv4 VRF Aggr:27/aggregate(vpn2)
   127.0.0.6/32     192.168.5.1     20/nolabel
   192.168.5.0/24   0.0.0.0         IPv4 VRF Aggr:27/aggregate(vpn2)
   172.17.128.0/32  192.168.5.1     21/nolabel
Route Distinguisher: 3:1 (vpn3)
   127.0.3.2/32     0.0.0.0         IPv4 VRF Aggr:28/aggregate(vpn3)
   127.0.0.8/32     192.168.7.1     22/nolabel
   192.168.7.0/24   0.0.0.0         IPv4 VRF Aggr:28/aggregate(vpn3)
   172.16.128.0/32  192.168.7.1     23/nolabel
Device# show mpls forwarding-table

Local  Outgoing    Prefix              Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
16     Pop tag     192.168.3.0/24    0          Et1/0      192.168.2.3
17     Pop tag     127.0.0.3/32      0          Et1/0      192.168.2.3
18     17          127.0.0.4/32      0          Et1/0      192.168.2.3
20     Untagged    127.0.0.6/32[V]   0          Et2/0      192.168.5.1
21     Untagged    172.17.128.0/32[V]0          Et2/0      192.168.5.1
22     Untagged    127.0.0.8/32[V]   0          Et3/0      192.168.7.1
23     Untagged    172.16.128.0/32[V]0          Et3/0      192.168.7.1
26     Pop Label   IPv4 VRF[V]       0          aggregate/vpn1
27     Pop Label   IPv4 VRF[V]       0          aggregate/vpn1
28     Pop Label   IPv4 VRF[V]       0          aggregate/vpn1
```

Example: Mixed Mode with Global Per-VRF

For this example, the following commands set VPN 1 for per-vrf label mode, VPN 2 for per-prefix label mode, and all remaining VPNs for per-vrf (globally).

In this example output, the **bold** text indicates the label modes:

```
Device# mpls label mode vrf vpn1 protocol bgp-vpnv4 per-vrf
Device# mpls label mode vrf vpn2 protocol bgp-vpnv4 per-prefix
Device# mpls label mode all-vrfs protocol bgp-vpnv4 per-vrf
Device# show ip vrf detail
VRF vpn1; default RD 1:1; default VPNID <not set>
VRF Table ID = 1
  Interfaces:
    Ethernet0/0              Serial5/0                Loopback1
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:1:1
  Import VPN route-target communities
    RT:1:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-vrf (Label 26)
VRF vpn2; default RD 2:1; default VPNID <not set>
VRF Table ID = 2
  Interfaces:
    Ethernet2/0              Loopback2
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:2:1
  Import VPN route-target communities
    RT:2:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-prefix
    per-vrf-aggr for connected and BGP aggregates (Label 27)
VRF vpn3; default RD 3:1; default VPNID <not set>
VRF Table ID = 3
  Interfaces:
    Ethernet3/0              Loopback3
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:3:1
  Import VPN route-target communities
    RT:3:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-vrf (Label 28)
Device# show ip bgp vpnv4 all label

   Network          Next Hop      In label/Out label
Route Distinguisher: 1:1 (vpn1)
   127.0.0.1/32     192.168.1.1     IPv4 VRF Aggr:26/nolabel
   127.0.0.5/32     127.0.0.4       nolabel/19
   192.168.1.0/24   0.0.0.0         IPv4 VRF Aggr:26/aggregate(vpn1)
                    192.168.1.1     IPv4 VRF Aggr:26/nolabel
   192.168.4.0/24   127.0.0.4       nolabel/20
   172.16.0.0/16    0.0.0.0         IPv4 VRF Aggr:26/aggregate(vpn1)
   172.16.128.0/32  192.168.1.1     IPv4 VRF Aggr:26/nolabel
Route Distinguisher: 2:1 (vpn2)
   127.0.2.2/32     0.0.0.0         IPv4 VRF Aggr:27/aggregate(vpn2)
   127.0.0.6/32     192.168.5.1     20/nolabel
   192.168.5.0/24   0.0.0.0         IPv4 VRF Aggr:27/aggregate(vpn2)
   172.17.128.0/32  192.168.5.1     21/nolabel
Route Distinguisher: 3:1 (vpn3)
   127.0.3.2/32     0.0.0.0         IPv4 VRF Aggr:28/aggregate(vpn3)
   127.0.0.8/32     192.168.7.1     IPv4 VRF Aggr:28/nolabel
   192.168.7.0/24   0.0.0.0         IPv4 VRF Aggr:28/aggregate(vpn3)
   172.16.128.0/32  192.168.7.1     IPv4 VRF Aggr:28/nolabel
Device# show mpls forwarding-table

Local  Outgoing    Prefix              Bytes tag  Outgoing      Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
16     Pop tag     192.168.3.0/24    0          Et1/0           192.168.2.3
17     Pop tag     127.0.0.3/32      0          Et1/0           192.168.2.3
18     17          127.0.0.4/32      0          Et1/0           192.168.2.3
20     Untagged    127.0.0.6/32[V]   0          Et2/0           192.168.5.1
21     Untagged    172.17.128.0/32[V]0          Et2/0           192.168.5.1
26     Pop Label   IPv4 VRF[V]       0          aggregate/vpn1
27     Pop Label   IPv4 VRF[V]       0
aggregate/vpn2
28     Pop Label   IPv4 VRF[V]       0          aggregate/vpn3
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco Master Command List, All Releases
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference
MPLS VPNs	*MPLS Layer 3 VPNs Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 2547	*BGP/MPLS*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS VPN Per VRF Label

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 13: Feature Information for MPLS VPN Per VRF Label

Feature Name	Releases	Feature Information
MPLS VPN Per VRF Label	12.2(18)SXF2 12.2(33)SRA 12.2(33)SXH 12.4(6)T Cisco IOS XE Release 2.2	The MPLS VPN Per VRF Label feature allows a user to configure a single VPN label for all local routes in the entire VPN routing and forwarding (VRF) domain. The feature incorporates a single (per VRF) VPN label for all local routes in the VRF table. In Cisco IOS Release 12.2(18)SXF2, this feature was introduced. In Cisco IOS Releases 12.2(33)SRA, 12.2(33)SRD, and 12.4(6)T, this feature was integrated. In Cisco IOS XE Release 2.2, support was added for the Cisco ASR 1000 Series Routers. The following commands were introduced or modified: **debug ip bgp vpnv4 unicast**, **mpls label mode**.

CHAPTER 13

MPLS VPN SNMP Notifications

This document describes the Simple Network Management Protocol (SNMP) agent support in Cisco IOS for Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) event notifications as implemented in the notifications section of the draft *MPLS/BGP Virtual Private Network Management Information Base Using SMIv2* (*draft-ietf-ppvpn-mpls-vpn-mib-03.txt*).

The MPLS VPN technology allows service providers to offer intranet and extranet VPN services that directly connect their customers' remote offices to a public network with the same security and service levels that a private network offers. The Provider-Provisioned VPN (PPVPN)-MPLS-VPN MIB notifications provide SNMP notification for critical MPLS VPN events.

The MPLS VPN SNMP Notifications feature provides the following benefits:

- A standards-based SNMP interface for retrieving information about critical MPLS VPN events.
- The generation and queuing of notifications that call attention to major changes in the operational status of MPLS VPN enabled interfaces; the forwarding of notification messages to a designated NMS for evaluation and action by network administrators.
- Advanced warning when VPN routing tables are approaching or exceed their capacity.
- Warnings about the reception of illegal labels on a VRF enabled interface. Such receptions may indicate misconfiguration or an attempt to violate security.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for MPLS VPN SNMP Notifications

The MPLS VPN SNMP Notifications feature requires the following:

- SNMP is installed and enabled on the label switching routers.
- Multiprotocol Label Switching (MPLS) is enabled on the label switching routers.
- Multiprotocol Border Gateway Protocol (BGP) is enabled on the label switching routers.
- Cisco Express Forwarding is enabled on the label switching routers.

Restrictions for MPLS VPN SNMP Notifications

- The MPLS-VPN-MIB agent is not implemented in this release.
- Configuration of the MIB using the SNMP SET command is not supported in this release.
- The retrieval of MPLS-VPN-MIB objects using SNMP GET is not supported in this release.

Information About MPLS VPN SNMP Notifications

Cisco Implementation of MPLS VPN MIB

SNMP agent code operating with the notifications of the MPLS VPN SNMP Notifications feature enables a standardized, SNMP-based approach to monitoring the MPLS VPN MIB notifications that aid in the management of Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) in Cisco software.

The MPLS VPN SNMP Notifications feature is based on the IETF draft specification *draft-ietf-ppvpn-mpls-vpn-mib-02.txt*, which includes notification objects that support MPLS VPN notification events. This IETF draft MIB, which undergoes revisions from time to time, is being evolved toward becoming a standard. Accordingly, the Cisco implementation of features of the MPLS VPN MIB is expected to track the evolution of the IETF draft MIB, and may change accordingly.

Some slight differences between the IETF draft MIB and the actual implementation of MPLS VPNs within Cisco software require some minor translations between the MPLS VPN MIB and the internal data structures of Cisco software. These translations are accomplished by means of the SNMP agent code. Also, while running as a low priority process, the SNMP agent provides a management interface to Cisco software. SNMP adds little overhead on the normal functions of the device.

The SNMP objects defined in the MPLS VPN MIB notifications can be viewed by any standard SNMP utility. The network administrator can retrieve information in the MPLS VPN MIB using standard SNMP **get** and **getnext** operations for SNMP v1, v2, and v3.

All MPLS VPN MIB objects are based on the IETF draft MIB; thus, no specific Cisco SNMP application is required to support the functions and operations pertaining to the MPLS VPN SNMP Notifications feature.

This section contains the following information about the Cisco implementation of the MPLS VPN MIB:

Capabilities Supported by MPLS VPN SNMP Notifications

The following functionality is supported in this release for the MPLS VPN SNMP Notifications feature. This feature provides you with the ability to do the following:

- Create and send notification messages that signal changes when critical Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) events occur.
- Enable, disable, and configure notification messages for MPLS VPN events by using extensions to existing SNMP CLI commands.
- Specify the IP address of a network management system (NMS) in the operating environment to which notification messages are sent.
- Write notification configurations into nonvolatile memory.

Notification Generation Events for the MPLS VPN MIB

The following notifications of the MPLS VPN MIB are implemented for this release:

- **mplsVrfIfUp**—Sent to an NMS when an interface comes up and is assigned a VPN routing/forwarding table instance (VRF).
- **mplsVrfIfDown**—Generated and sent to the NMS when a VRF is removed from an interface or the interface transitions from an operationally “up” state to a “down” state.

Note For the mplsVrfIfUp or mplsVrfIfDown notifications to be issued on ATM or Frame Relay subinterfaces, you must configure the **snmp-server traps atm subif** command or the **snmp-server traps frame-relay subif** command on the subinterfaces, respectively.

- **mplsNumVrfRouteMidThreshExceeded**—Generated and sent when the middle (warning) threshold is crossed. You can configure this threshold in the CLI by using the following commands:

```
Router(config)# ip vrf vrf-name
Router(config-vrf)# maximum routes max-thresh
mid-thresh (% of max)
```

This notification is sent to the NMS only at the time the threshold is exceeded. Whenever the number of routes falls below this threshold and exceeds the threshold again, a notification is sent to the NMS. (See the figure below for a comparison of the warning and maximum thresholds.)

- **mplsNumVrfRouteMaxThreshExceeded**—Generated and sent when you attempt to create a route on a VRF that already contains the maximum number of routes as defined by the following CLI commands:

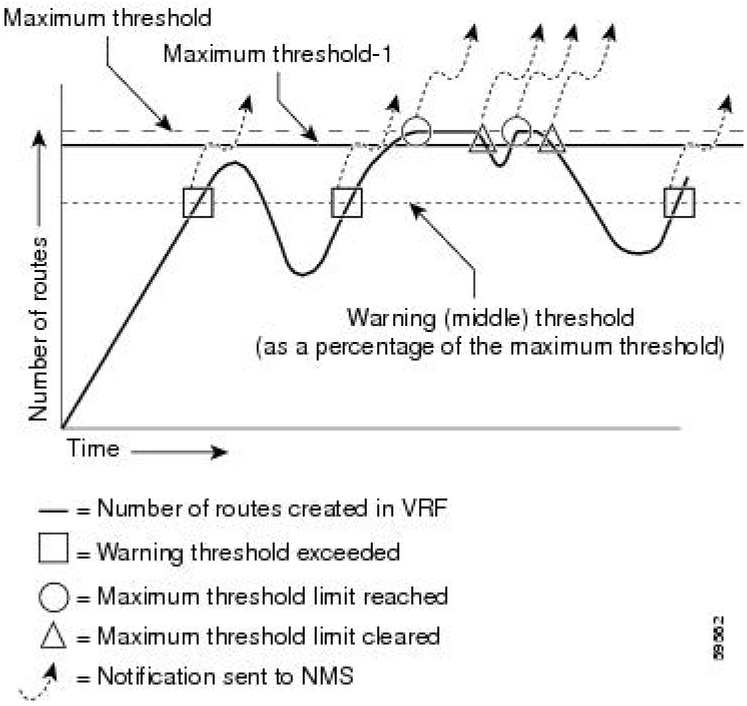
```
Router(config)# ip vrf vrf-name
Router(config-vrf)# maximum routes max-thresh
mid-thresh (% of max)
```

A trap notification is sent to the NMS when you attempt to exceed the maximum threshold. Another notification is not sent until the number of routes falls below the maximum threshold and reaches the maximum threshold again. (See the figure below for an example of how this notification works and for a comparison of the maximum and warning thresholds.)

Note The **maximum routes** command sets the number of routes for a VRF. You *cannot* exceed the number of routes in the VRF that you set with the **maximum routes** *max-thresh* CLI command. Prior to this implementation of the MPLS-VPN-MIB, you were not notified when this threshold (or the warning threshold) was reached.

- **mplsNumVrfSecIllegalLabelThreshExceeded**—Generated and sent when the amount of illegal labels received on a VRF interface exceeds the threshold *mplsVpnVrfSecIllegalLabelRcvThresh*. This threshold is defined with a value of 0. Therefore, a notification is sent when the first illegal label is received on a VRF. Labels are considered illegal if they are outside of the valid label range, do not have a Label Forwarding Information Base (LFIB) entry, or the table ID of the message does not match the table ID for the label in the LFIB.

Figure 14: Comparison of Warning and Maximum Thresholds



Notification Specification for MPLS-VPN-MIB

In an SNMPv1 notification, each VPN notification has a generic type identifier and an enterprise-specific type identifier for identifying the notification type.

- The generic type for all VPN notifications is “enterpriseSpecific” as this is not one of the generic notification types defined for SNMP.

- The enterprise-specific type is identified as follows:
  - 1 for *mplsVrfIfUp*
  - 2 for *mplsVrfIfDown*
  - 3 for *mplsNumVrfRouteMidThreshExceeded*
  - 4 for *mplsNumVrfRouteMaxThreshExceeded*
  - 5 for *mplsNumVrfSecIllegalLabelThreshExceeded*

In SNMPv2, the notification type is identified by an **SnmpTrapOID** varbind (variable binding consisting of an object identifier (OID) type and value) included within the notification message.

Each notification also contains two additional objects from the MPLS-VPN-MIB. These objects provide additional information about the event, as follows:

- The VRF interface up/down notifications provide additional variables--*mplsVpnInterfaceConfIndex* and *mplsVpnVrfName*-- in the notification. These variables describe the SNMP interface index and the VRF name, respectively.
- The mid and max threshold notifications include the *mplsVpnVrfName* variable (VRF name) as well as the *mplsVpnVrfPerfCurrNumRoutes* variable that indicates the current number of routes within the VRF.
- The illegal label notification includes the *mplsVpnVrfName* variable (VRF name) and the *mplsVpnVrfSecIllegalLabelViolations* variable that maintains the current count of illegal labels on a VPN.

Monitoring the MPLS VPN SNMP Notifications

When MPLS-VPN-MIB notifications are enabled, notification messages relating to specific Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) events within Cisco software are generated and sent to a specified network management system (NMS) in the network. Any utility that supports SNMPv1 or SNMPv2 notifications can receive notification messages.

To monitor MPLS-VPN-MIB notification messages, log in to an NMS that supports a utility that displays SNMP notifications, and start the display utility.

How to Configure the MPLS VPN SNMP Notifications

Configuring an SNMP Community

An SNMP community string defines the relationship between the SNMP manager and the agent. The community string acts like a password to regulate access to the agent on the device.

Perform this task to configure an SNMP community.

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **configure terminal**
4. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*acl-number*]

5. **do copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** **Example:** `Device# show running-config`	Displays the running configuration to determine if an SNMP agent is already running. If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify the information or change it as needed.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*acl-number*] **Example:** `Device(config)# snmp-server community comaccess ro`	Sets up the community access string to permit access to the Simple Network Management Protocol (SNMP). • The *string* argument acts like a password and permits access to the SNMP protocol. • The **view***view-name* keyword and argument specifies the name of a previously defined view. The view defines the objects available to the community. • The **ro** keyword specifies read-only access. Authorized management stations are only able to retrieve MIB objects. • The **rw** keyword specifies read-write access. Authorized management stations are able to both retrieve and modify MIB objects. • The *acl-number* argument is an integer from 1 to 99 that specifies an access list of IP addresses that are allowed to use the community string to gain access to the SNMP agent.

Step 5	**do copy running-config startup-config**
	Example:
	`Device(config)# do copy running-config startup-config`
		Saves the modified configuration to nonvolatile memory (NVRAM) as the startup configuration file. (The **do** command allows you to perform Exec level commands in configuration mode.)

Configuring the Device to Send SNMP Traps

Perform this task to configure the device to send traps to a host.

The **snmp-server host** command specifies which hosts receive traps. The **snmp-server enable traps** command globally enables the trap production mechanism for the specified traps.

Note Although you can set the *community-string* argument using the **snmp-server host** command by itself, we recommend you define this string using the **snmp-server community** command prior to using the **snmp-server host** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** *host-addr* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]
4. Do one of the following:
   - **snmp-server enable traps atm** [**pvc** | **subif**]
   - **snmp-server enable traps frame-relay** [**subif**]
5. **snmp-server enable traps mpls vpn**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server host** *host-addr* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] **Example:** `Device(config)# snmp-server host 172.20.2.160 traps comaccess mpls-vpn`	Specifies the recipient of an SNMP notification operation. • The *host-addr* argument specifies the name or Internet address of the host (the targeted recipient). • The **traps** keyword sends SNMP traps to this host. This is the default. • The **informs** keyword sends SNMP informs to this host.

	Command or Action	Purpose
		• The **version** keyword specifies the version of the SNMP used to send the traps. Version 3 is the most secure model, as it allows packet encryption with the **priv** keyword. If you use the **version** keyword, you must specify one of the following: • **1**—SNMPv1. This option is not available with informs. • **2c**—SNMPv2C. • **3**—SNMPv3. The following three optional keywords can follow the **version 3** keyword (**auth**, **noauth**, **priv**). • The *community-string* argument is a password-like community string sent with the notification operation. • The **udp-port** *port* keyword and argument names the UDP port of the host to use. The default is 162. • The *notification-type* argument specifies the type of notification to be sent to the host. If no type is specified, all notifications are sent. MPLS VPN notifications are specified with the **mpls-vpn** keyword.

Step 4	Do one of the following:
	• **snmp-server enable traps atm** [**pvc** | **subif**]
	• **snmp-server enable traps frame-relay** [**subif**]
	Example:
	`Device(config)# snmp-server enable traps atm subif`
	Example:
	`Device(config)# snmp-server enable traps frame-relay subif`
		(For ATM subinterfaces only) Enables the sending of ATM SNMP notifications.
		• The **pvc** keyword enables SNMP ATM permanent virtual circuit (PVC) traps.
		• The **subif** keyword enables SNMP ATM subinterface traps.
		or
		(For Frame Relay subinterfaces only) Enables Frame Relay DLCI link status SNMP notifications.
		• The **subif** keyword enables SNMP Frame Relay subinterface traps.
		Note For mplsVrfIfUp or mplsVrfIfDown notifications to be issued on ATM or Frame Relay subinterfaces, you must configure the appropriate **snmp-server enable traps** command with the **subif** keyword.

Step 5	**snmp-server enable traps mpls vpn** **Example:** `Device(config)# snmp-server enable traps mpls vpn vrf-up vrf-down`	Enables the device to send MPLS VPN SNMP notifications.

	Command or Action	Purpose

Step 6	**end**
	Example:
	`Device(config)# end`
		(Optional) Returns to user EXEC mode.

Configuring Threshold Values for MPLS VPN SNMP Notifications

Perform this task to configure threshold values for MPLS VPN SNMP notifications.

The **mplsNumVrfRouteMidThreshExceeded** notification event is generated and sent when the middle (warning) threshold is crossed. You can configure this threshold in the CLI by using the **maximum routes** command in VRF configuration mode. This notification is sent to the NMS only at the time the threshold is exceeded. Whenever the number of routes falls below this threshold and exceeds the threshold again, a notification is sent to the NMS.

The **mplsNumVrfRouteMaxThreshExceeded** notification event is generated and sent when you attempt to create a route on a VRF that already contains the maximum number of routes as defined by the **maximum routes** command in VRF configuration mode. A trap notification is sent to the NMS when you attempt to exceed the maximum threshold. Another notification is not sent until the number of routes falls below the maximum threshold and reaches the maximum threshold again.

(See the figure above for an example of how this notification works and for a comparison of the maximum and warning thresholds.)

Note The **maximum routes** command sets the number of routes for a VRF. You *cannot* exceed the number of routes in the VRF that you set with the **maximum routes** *max-thresh* CLI command. Prior to this implementation of the MPLS-VPN-MIB, you were not notified when this threshold (or the warning threshold) was reached.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **maximum routes** *limit* {*warn-threshold* | **warning-only**}
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf vpn1`	Configures a VRF routing table. • The *vrf-name* argument specifies the name assigned to a VRF.

Step 4	**maximum routes** *limit* {*warn-threshold* | **warning-only**}
	Example:
	`Device(config-vrf)# maximum routes 10000 80`
		Limits the maximum number of routes in a VRF to prevent a PE device from importing too many routes.
		• The *limit* argument specifies the maximum number of routes allowed in a VRF. You may select from 1 to 4,294,967,295 routes to be allowed in a VRF.
		• The *warn-threshold* argument specifies when the threshold limit is reached and routes are rejected. The threshold limit is a percentage of the *limit* specified, from 1 to 100 percent.
		• The **warning-only** keyword specifies that a SYSLOG error message is issued when the maximum number of routes allowed for a VRF exceeds the threshold. However, additional routes are still allowed.

Step 5	**end**
	Example:
	`Device(config-vrf)# end`
		(Optional) Returns to privileged EXEC mode.

Configuration Examples for MPLS VPN SNMP Notifications

Example: Configuring the Community

The following example shows enabling a simple SNMP community group. This configuration permits any SNMP client to access all MPLS-VPN-MIB objects with read-only access using the community string comaccess.

```
Device# configure terminal
Device(config)# snmp-server community comaccess ro
```

Verify that the SNMP master agent is enabled for the MPLS VPN SNMP Notifications feature:

```
Device# show running-config | include snmp-server
Building configuration...
```

```
....
snmp-server community comaccess RO
....
```

Note If you do not see any “snmp-server” statements, SNMP has not been enabled on the device.

Example: Configuring the Device to Send SNMP Traps

The following example shows you how to enable the device to send MPLS VPN notifications to host 172.20.2.160 using the comaccess community string if a VRF transitions from a down state to an up state or from an up state to a down state.

```
Device# configure terminal
Device(config)# snmp-server host 172.20.2.160 traps comaccess mpls-vpn
Device(config)# snmp-server enable traps mpls vpn vrf-up vrf-down
```

Example: Configuring Threshold Values for MPLS VPN SNMP Notifications

The following example shows how to set a maximum threshold or 10000 routes and a warning threshold that is 80 percent of the maximum threshold for a VRF named vpn1 on a device:

```
Device(config)# ip vrf vpn1
Device(config)# maximum routes 10000 80
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference
MPLS Virtual Private Network (VPN) configuration tasks	“MPLS Virtual Private Networks” module in the *MPLS Layer 3 VPNs Configuration Guide*

MIBs

MIBs	MIBs Link
MPLS/BGP Virtual Private Network Management Information Base Using SMIv2 (*draft-ietf-ppvpn-mpls-vpn-mib-03.txt*) MPLS-VPN-MIB.my	To obtain lists of supported MIBs by platform and Cisco software release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 2233	*The Interfaces Group MIB using SMIv2*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS VPN SNMP Notifications

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 14: Feature Information for MPLS VPN SNMP Notifications

Feature Name	Releases	Feature Information
MPLS VPN SNMP Notifications	12.0(21)ST 12.0(22)S 12.2(13)T	The MPLS VPN SNMP Notifications feature provides Simple Network Management Protocol (SNMP) agent support in Cisco software for Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) event notifications. In Cisco IOS Release 12.0(21)ST, this feature was introduced. In Cisco IOS Release 12.0(22)S, this feature was integrated. In Cisco IOS Release 12.2(13)T, this feature was integrated. Supported platforms: • Cisco IOS 12.0 S and ST Releases: Cisco 7500 series, Cisco 12000 series. • Cisco IOS 12.2 T Releases: Cisco 3620, Cisco 3640, Cisco 7200 series, Cisco 7500 series, Cisco MGX 8850-RPM. **Note** In Cisco IOS Releases 12.0(21)ST and 12.0(22)S, the PPVPN MPLS-VPN-MIB notifications are described in the *MPLS VPN--SNMP MIB Support* feature module. The following commands were introduced or modified: **snmp-server enable traps mpls vpn**, **snmp-server host**.

Glossary

ASN.1—Abstract Syntax Notation One. OSI language for describing data types independent of particular computer structures and representation techniques. Described by ISO International Standard 8824.

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.

CEF—Cisco Express Forwarding. An advanced Layer 3 IP switching technology. CEF optimizes network performance and scalability for networks with large and dynamic traffic patterns.

CE device—customer edge router. A router on the border between a VPN provider and a VPN customer that belongs to the customer.

community—In SNMP, a logical group of managed devices and NMSs in the same administrative domain.

community name—*See* community string.

community string—Text string that acts as a password and is used to authenticate messages sent between a managed station and a router containing an SNMP agent. The community string is sent in every packet between the manager and the client. Also called a community name.

IETF—Internet Engineering Task Force. Task force consisting of over 80 working groups responsible for developing Internet standards. The IETF operates under the auspices of ISOC. *See also* ISOC.

informs—A type of notification message that is more reliable than a conventional trap notification message, because the informs message notification requires acknowledgment, and a trap notification does not.

ISOC—Internet Society. International nonprofit organization, founded in 1992, that coordinates the evolution and use of the Internet. In addition, ISOC delegates authority to other groups related to the Internet, such as the IAB. ISOC is headquartered in Reston, Virginia (United States).

label—A short, fixed-length data construct that tells switching nodes how to forward data (packets or cells).

label distribution protocol—*See* LDP.

label forwarding information base—*See* LFIB.

label switch router—*See* LSR.

LDP —label distribution protocol. A standard protocol between MPLS-enabled routers that is used for the negotiation of the labels (addresses) used to forward packets.

LFIB —label forwarding information base. In the Cisco Label Switching system, the data structure for storing information about incoming and outgoing tags (labels) and associated equivalent packets suitable for labeling.

LSR —label switch router. A device that forwards MPLS packets based on the value of a fixed-length label encapsulated in each packet.

MIB —Management Information Base. Database of network management information that is used and maintained by a network management protocol such as SNMP or CMIP. The value of a MIB object can be changed or retrieved using SNMP or CMIP commands, usually through a GUI network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

MPLS —Multiprotocol Label Switching. A method for forwarding packets (frames) through a network. It enables routers at the edge of a network to apply labels to packets (frames). ATM switches or existing routers in the network core can switch packets according to the labels with minimal lookup overhead.

MPLS interface—An interface on which MPLS traffic is enabled.

MPLS VPN—Multiprotocol Label Switching Virtual Private Network. Using MPLS VPNs in a Cisco network provide the capability to deploy and administer scalable Layer 3 VPN backbone services including applications, data hosting network commerce, and telephony services, to business customers. A VPN is a secure IP-based network that shares resources on one or more physical networks. A VPN contains geographically dispersed sites that can communicate securely over a shared backbone.

For an MPLS VPN Solution, an MPLS VPN is a set of PEs that are connected by means of a common “backbone” network to supply private IP interconnectivity between two or more customer sites for a given customer. Each VPN has a set of provisioning templates and policies and can span multiple provider administrative domains (PADs).

Multiprotocol label Switching—*See* MPLS.

notification —A message sent by an SNMP agent to a network management station, console, or terminal to indicate that a significant event within Cisco software has occurred. *See also* trap.

NMS —network management system. A powerful, well-equipped computer (typically an engineering workstation) that is used by a network administrator to communicate with other devices in the network. An NMS is typically used to manage network resources, gather statistics, and perform a variety of network administration and configuration tasks.

PE device—provider edge router. A router on the border between a VPN provider and a VPN customer that belongs to the provider.

PPVPN —Provider-Provisioned VPN. The name of the IETF working group that is developing the PPVPN-MPLS-VPN MIB (MPLS-VPN-MIB).

QoS —quality of service. Measure of performance for a transmission system that reflects its transmission quality and service availability.

RSVP —Resource Reservation Protocol. Protocol for reserving network resources to provide Quality of Service guarantees to application flows.

Simple Network Management Protocol—*See* SNMP.

SNMP —Simple Network Management Protocol. Network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices, and to manage configurations, statistics collection, performance, and security. See *also* SNMP2.

SNMP2 —SNMP Version 2. Version 2 of the popular network management protocol. SNMP2 supports centralized as well as distributed network management strategies, and includes improvements in the SMI, protocol operations, management architecture, and security. *See also* SNMP.

traffic engineering—The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

trap —A message sent by an SNMP agent to a network management station, console, or terminal, indicating that a significant event occurred. Traps (notifications) are less reliable than inform requests, because the receiver does not send an acknowledgment when it receives a trap. The sender cannot determine if the trap was received. *See also* notification.

VPN —Virtual Private Network. A group of sites that, as the result of a set of administrative policies, are able to communicate with each other over a shared backbone network. *See* MPLS VPN.

VRF —VPN routing/forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE device.

CHAPTER 14

Multi-VRF Selection Using Policy-Based Routing

The Multi-VRF Selection Using Policy-Based Routing (PBR) feature allows a specified interface on a provider edge (PE) device to route packets to Virtual Private Networks (VPNs) based on packet length or match criteria defined in an IP access list.

You can enable VPN routing and forwarding (VRF) selection by policy routing packets through a route map, through the global routing table, or to a specified VRF.

You can enable policy-routing packets for VRF instances by using route map commands with **set** commands.

On supported hardware, you can configure both the Multi-VRF Selection Using Policy-Based Routing feature and the MPLS VPN VRF Selection Based on a Source IP Address feature on the same interface.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for Multi-VRF Selection Using Policy-Based Routing

- The device must support policy-based routing (PBR) in order for you to configure this feature. For platforms that do not support PBR, use the MPLS VPN VRF Selection Based on a Source IP Address feature.
- A Virtual Private Network (VPN) virtual routing and forwarding (VRF) instance must be defined before you configure this feature. An error message is displayed on the console if no VRF exists.

RestrictionsforMulti-VRFSelectionUsingPolicy-BasedRouting

- All commands that aid in routing also support hardware switching, except for the **set ip next-hop verify availability** command because Cisco Discovery Protocol information is not available in the line cards.
- Protocol Independent Multicast (PIM) and multicast packets do not support policy-based routing (PBR) and cannot be configured for a source IP address that is a match criterion for this feature.
- The **set vrf** and **set ip global next-hop** commands can be configured with the **set default interface**, **set interface** , **set ip default next-hop**, and **set ip next-hop** commands. But the **set vrf** and **set ip global next-hop** commands take precedence over the **set default interface**, **set interface** , **set ip default next-hop**, and **set ip next-hop** commands. No error message is displayed if you attempt to configure the **set vrf** command with any of these three **set** commands.
- The Multi-VRF Selection Using Policy-Based Routing feature cannot be configured with IP prefix lists.
- The **set global** and **set vrf** commands cannot be simultaneously applied to a route map.
- The Multi-VRF Selection Using Policy-Based Routing feature supports VRF-lite; that is, only IP routing protocols run on the device. Multiprotocol Label Switching (MPLS) and Virtual Private Networks (VPNs) cannot be configured. However, the **set vrf** command will work in MPLS VPN scenarios.
- If you delete one VRF using **no vrf definition** *vrf-name* command, then other VRFs in the VRF routing table are also removed unexpectedly; when **ip vrf receive** command is configured with receive entries above 400, and IPv4 and IPv6 routes above 2000. This is applicable only for Cisco ASR 1000 platform.
- In a VRF receive scenario, the memory requirements are proportional to the number of VRF receives that are configured multiplied by the number of directly connected neighbours (Cisco Express Forwarding adjacencies). When the **ip vrf receive** command is configured, Cisco Express Forwarding adjacency prefixes are copied to the VRF. Network resources might be exhausted based on number of bytes per each adjacency prefix, number of adjacency prefixes, number of VRF receives configured, and the platform-specific route processor memory restrictions applicable to Cisco Express Forwarding entries.

Information About Multi-VRF Selection Using Policy-Based Routing

Policy Routing of VPN Traffic Based on Match Criteria

The Multi-VRF Selection Using Policy-Based Routing feature is an extension of the MPLS VPN VRF Selection Based on a Source IP Address feature. The Multi-VRF Selection Using Policy-Based Routing feature allows you to policy route Virtual Private Network (VPN) traffic based on match criteria. Match criteria are defined in an IP access list and/or are based on packet length. The following match criteria are supported in Cisco software:

- IP access lists—Define match criteria based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. All IP access list configuration options in Cisco software can be used to define match criteria.
- Packet lengths—Define match criteria based on the length of a packet, in bytes. The packet length filter is defined in a route map with the **match length** route-map configuration command.

Policy routing is defined in the route map. The route map is applied to the incoming interface with the **ip policy route-map** interface configuration command. An IP access list is applied to the route map with the **match ip address** route-map configuration command. Packet length match criteria are applied to the route map with the **match length** route-map configuration command. The **set** action is defined with the **set vrf** route-map configuration command. The match criteria are evaluated, and the appropriate VRF is selected by the **set** command. This combination allows you to define match criteria for incoming VPN traffic and policy route VPN packets out to the appropriate virtual routing and forwarding (VRF) instance.

Policy-Based Routing set Commands

Policy-routing Packets for VRF Instances

To enable policy-routing packets for virtual routing and forwarding (VRF) instances, you can use route map commands with the following **set** commands. They are listed in the order in which the device uses them during the routing of packets.

- **set tos**—Sets the Type of Service (TOS) bits in the header of an IP packet.
- **set df**—Sets the Don't Fragment (DF) bit in the header of an IP packet.
- **set vrf**—Routes packets through the specified interface. The destination interface can belong only to a VRF instance.
- **set global**—Routes packets through the global routing table. This command is useful for routing ingress packets belonging to a specific VRF through the global routing table.
- **set ip vrf next-hop**—Indicates where to output IPv4 packets that pass a match criteria of a route map for policy routing when the IPv4 next hop must be under a specified VRF.
- **set ipv6 vrf next-hop**—Indicates where to output IPv6 packets that pass a match criteria of a route map for policy routing when the IPv6 next hop must be under a specified VRF.

- **set ip global next-hop**—Indicates where to forward IPv4 packets that pass a match criterion of a route map for policy routing and for which the Cisco software uses the global routing table. The global keyword explicitly defines that IPv4 next-hops are under the global routing table.
- **set ipv6 global next-hop**—Indicates where to forward IPv6 packets that pass a match criterion of a route map for policy routing and for which the Cisco software uses the global routing table. The global keyword explicitly defines that IPv6 next-hops are under the global routing table.
- **set interface**—When packets enter a VRF, routes the packets out of the egress interface under the same VRF according to the set interface policy, provided that the Layer 2 rewrite information is available.
- **set ip default vrf**—Provides IPv4 inherit-VRF and inter-VRF routing. With inherit-VRF routing, IPv4 packets arriving at a VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, IPv4 packets arriving at a VRF interface are routed through any other outgoing VRF interface.
- **set ipv6 default vrf**—Provides IPv6 inherit-VRF and inter-VRF routing. With inherit-VRF routing, IPv6 packets arriving at a VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, IPv6 packets arriving at a VRF interface are routed through any other outgoing VRF interface.
- **set ip default global**—Provides IPv4 VRF to global routing.
- **set ipv6 default global**—Provides IPv6 VRF to global routing.
- **set default interface**—Indicates where to output packets that pass a match criterion of a route map for policy routing and have no explicit route to the destination. The interface can belong to any VRF.
- **set ip default next-hop**—Indicates where to output IPv4 packets that pass a match criterion of a route map for policy routing and for which the Cisco software has no explicit route to a destination.
- **set ipv6 default next-hop**—Indicates where to IPv6 output packets that pass a match criterion of a route map for policy routing and for which the Cisco software has no explicit route to a destination.

Change of Normal Routing and Forwarding Behavior

When you configure policy-based routing (PBR), you can use the following six **set** commands to change normal routing and forwarding behavior. Configuring any of these **set** commands, with the potential exception of the **set ip next-hop** command, overrides the routing behavior of packets entering the interface if the packets do not belong to a virtual routing and forwarding (VRF) instance. The packets are routed from the egress interface across the global routing table.

- **set default interface**—Indicates where to output packets that pass a match criterion of a route map for policy routing and have no explicit route to the destination.
- **set interface**—When packets enter a VRF interface, routes the packets out of the egress interface under the same VRF according to the set interface policy, provided that the Layer 2 rewrite information is available.

  Note The interface must be a peer-to-peer (P2P) interface.

- **set ip default next-hop**—Indicates where to output IPv4 packets that pass a match criterion of a route map for policy routing and for which the Cisco software has no explicit route to a destination.

- **set ipv6 default next-hop**—Indicates where to output IPv6 packets that pass a match criterion of a route map for policy routing and for which the Cisco software has no explicit route to a destination.
- **set ip next-hop**—Indicates where to output IPv4 packets that pass a match criterion of a route map for policy routing. If an IPv4 packet is received on a VRF interface and is transmitted from another interface within the same VPN, the VRF context of the incoming packet is inherited from the interface.
- **set ipv6 next-hop**—Indicates where to output IPv6 packets that pass a match criterion of a route map for policy routing. If an IPv6 packet is received on a VRF interface and is transmitted from another interface within the same Virtual Private Network (VPN), the VRF context of the incoming packet is inherited from the interface.

Support of Inherit-VRF Inter-VRF and VRF-to-Global Routing

The Multi-VRF Selection Using Policy-Based Routing (PBR) feature supports inherit-VRF and inter-VRF. With inherit-VRF routing, packets arriving at a virtual routing and forwarding (VRF) interface are routed by the same outgoing VRF interface. With inter-VRF routing, packets arriving at a VRF interface are routed through any other outgoing VRF interface.

VRF-to-global routing causes packets that enter any VRF interface to be routed through the global routing table. When a packet arrives on a VRF interface, the destination lookup normally is done only in the corresponding VRF table. If a packet arrives on a global interface, the destination lookup is done in the global routing table.

The Multi-VRF Selection Using Policy-Based Routing feature modifies the following **set** commands to support inherit-VRF, inter-VRF, and VRF-to-global routing. The commands are listed in the order in which the device uses them during the routing of packets.

- **set global**—Routes packets through the global routing table. This command is useful for routing ingress packets belonging to a specific VRF through the global routing table.
- **set ip global next-hop**—Indicates where to forward IPv4 packets that pass a match criterion of a route map for policy routing and for which the Cisco software uses the global routing table.
- **set ipv6 global next-hop**—Indicates where to forward IPv6 packets that pass a match criterion of a route map for policy routing and for which the Cisco software uses the global routing table.
- **set ip vrf next-hop**—Causes the device to look up the IPv4 next hop in the VRF table. If an IPv4 packet arrives on an interface that belongs to a VRF and the packet needs to be routed through a different VRF, you can use the **set ip vrf next-hop** command.
- **set ipv6 vrf next-hop**—Causes the device to look up the IPv6 next hop in the VRF table. If an IPv6 packet arrives on an interface that belongs to a VRF and the packet needs to be routed through a different VRF, you can use the **set ipv6 vrf next-hop** command.
- **set ip default vrf**—Provides IPv4 inherit-VRF and inter-VRF routing. With IPv4 inherit-VRF routing, IPv4 packets arriving at a VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, IPv4 packets arriving at a VRF interface are routed through any other outgoing VRF interface.
- **set ipv6 default vrf**—Provides IPv6 inherit-VRF and inter-VRF routing. With IPv6 inherit-VRF routing, IPv6 packets arriving at a VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, IPv6 packets arriving at a VRF interface are routed through any other outgoing VRF interface.
- **set interface**—When packets enter a VRF, routes the packets out of the egress interface under the same VRF, according to the set interface policy, provided that the Layer 2 rewrite information is available.

- **set default interface**—Indicates where to output packets that pass a match criterion of a route map for policy routing and have no explicit route to the destination. The interface can belong to any VRF.
- **set ip next-hop**—Routes IPv4 packets through the global routing table in an IPv4-to-IPv4 routing and forwarding environment.
- **set ipv6 next-hop**—Routes IPv6 packets through the global routing table in an IPv6-to-IPv6 routing and forwarding environment.
- **set vrf**—Selects the appropriate VRF after a successful match occurs in the route map. VRS-aware PSV allows only inter-VRF (or VRF-to-VRF) switching.

How to Configure Multi-VRF Selection Using Policy-Based Routing

Defining the Match Criteria for Multi-VRF Selection Using Policy-Based Routing

Define the match criteria for the Multi-VRF Selection using Policy-Based Routing (PBR) feature so that you can selectively route the packets instead of using their default routing and forwarding.

The match criteria for the Multi-VRF Selection using Policy-Based Routing are defined in an access list. Standard, named, and extended access lists are supported.

You can define the match criteria based on the packet length by configuring the **match length** route-map configuration command. This configuration option is defined entirely within a route map.

The following sections explain how to configure PBR route selection:

Configuring Multi-VRF Selection Using Policy-Based Routing with a Standard Access List

Before you begin

The tasks in the following sections assume that the virtual routing and forwarding (VRF) instance and associated IP address are already defined.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} [**source** *source-wildcard*] [**log**]

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} [**source** *source-wildcard*] [**log**] **Example:** `Device(config)# access-list 40 permit source 10.1.1.0/24 0.0.0.255`	Creates an access list and defines the match criteria for the route map. • Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. You can use all IP access list configuration options to define match criteria. • The example creates a standard access list numbered 40. This filter permits traffic from any host with an IP address in the 10.1.1.0/24 subnet.

Configuring Multi-VRF Selection Using Policy-Based Routing with a Named Extended Access List

To configure Multi-VRF Selection using Policy-Based Routing (PBR) with a named extended access list, complete the following steps.

Before you begin

The tasks in the following sections assume that the virtual routing and forwarding (VRF) instance and associated IP address are already defined.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*]
4. [*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos***tos*] [**ttl** *operator-vaue*] [**log**] [**time-range** *time-range-name*] [**fragments**]

DETAILED STEPS

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

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] **Example:** `Device(config)# ip access-list extended NAMEDACL`	Specifies the IP access list type and enters the corresponding access list configuration mode. • You can specify a standard, extended, or named access list.
Step 4	[*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos***tos*] [**ttl** *operator-vaue*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# permit ip any any option any-options`	Defines the criteria for which the access list will permit or deny packets. • Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. You can use all IP access list configuration options to define match criteria. • The example creates a named access list that permits any configured IP option.

Configuring Multi-VRF Selection in a Route Map

Incoming packets are filtered through the match criteria that are defined in the route map. After a successful match occurs, the **set** command configuration determines the VRF through which the outbound Virtual Private Network (VPN) packets will be policy routed.

Before you begin

You must define the virtual routing and forwarding (VRF) instance before you configure the route map; otherwise an error message appears on the console.

A receive entry must be added to the VRF selection table with the **ip vrf receive** command. If a match and set operation occurs in the route map but there is no receive entry in the local VRF table, the packet will be dropped if the packet destination is local.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **named-ordering-route-map enable**]
4. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [
5. Do one of the following :
   - **set ip vrf** *vrf-name* **next-hop** *global-ipv4-address* [*...global-ipv4-address*]
   - **set ipv6 vrf** *vrf-name* **next-hop** *global-ipv6-address* [*...global-ipv6-address*]
   - **set ip next-hop recursive vrf** *global-ipv4-address* [*...global-ipv4-address*]
   - **set ip global next-hop** *global-ipv4-address* [*...global-ipv4-address*]
   - **set ipv6 global next-hop** *global-ipv6-address* [*...global-ipv6-address*]

6. Do one of the following:
   - **match ip address** {*acl-number* [*acl-name* | *acl-number*]}
   - **match length** *minimum-lengthmaximum-length*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**named-ordering-route-map enable**] **Example:** `Device(config)# named-ordering-route-map enable`	Enables ordering of route-maps based on a string provided by the user.
Step 4	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**Example:** `Device(config)# route-map alpha permit ordering-seq`	Configures a route map and specifies how the packets are to be distributed. .
Step 5	Do one of the following : • **set ip vrf** *vrf-name* **next-hop** *global-ipv4-address* [*...global-ipv4-address*] • **set ipv6 vrf** *vrf-name* **next-hop** *global-ipv6-address* [*...global-ipv6-address*] • **set ip next-hop recursive vrf** *global-ipv4-address* [*...global-ipv4-address*] • **set ip global next-hop** *global-ipv4-address* [*...global-ipv4-address*] • **set ipv6 global next-hop** *global-ipv6-address* [*...global-ipv6-address*] **Example:** `Device(config-route-map)# set ip vrf myvrf next-hop 10.0.0.0` **Example:**	Indicates where to forward packets that pass a match criterion of a route map for policy routing when the IPv4 next hop must be under a specified VRF. Indicates where to forward packets that pass a match criterion of a route map for policy routing when the IPv6 next hop must be under a specified VRF. Indicates the IPv4 address to which destination or next hop is used for packets that pass the match criterion configured in the route map. Indicates the IPv4 address to forward packets that pass a match criterion of a route map for policy routing and for which the software uses the global routing table. Indicates the IPv6 address to forward packets that pass a match criterion of a route map for policy routing and for which the software uses the global routing table.

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td></td><td><code>Device(config-route-map)# set ipv6 vrf myvrf next-hop 2001.DB8:4:1::1/64</code>
<b>Example:</b>
<code>Device(config-route-map)# set ip next-hop recursive vrf 10.0.0.0</code>
<b>Example:</b>
<code>Device(config-route-map)# set ip global next-hop 10.0.0.0</code>
<b>Example:</b>
<code>Device(config-route-map)# set ipv6 global next-hop 2001.DB8:4:1::1/64</code></td><td></td></tr>
<tr><td><b>Step 6</b></td><td>Do one of the following:
• <b>match ip address</b> {<i>acl-number</i> [<i>acl-name</i> | <i>acl-number</i>]}
• <b>match length</b> <i>minimum-lengthmaximum-length</i>
<b>Example:</b>
<code>Device(config-route-map)# match ip address 1 or</code>
<b>Example:</b>
<code>Device(config-route-map)# match length 3 200</code></td><td>Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets. IP access lists are supported.
• The example configures the route map to use standard access list 1 to define match criteria.
Specifies the Layer 3 packet length in the IP header as a match criterion in a class map.
• The example configures the route map to match packets that are 3 to 200 bytes in length.</td></tr>
<tr><td><b>Step 7</b></td><td><b>end</b>
<b>Example:</b>
<code>Device(config-route-map)# end</code></td><td>Returns to privileged EXEC mode.</td></tr>
</table>

Configuring Multi-VRF Selection Using Policy-Based Routing and IP VRF Receive on the Interface

The route map is attached to the incoming interface with the **ip policy route-map** interface configuration command.

The source IP address must be added to the virtual routing and forwarding (VRF) selection table. VRF selection is a one-way (unidirectional) feature. It is applied to the incoming interface. If a **match** and **set** operation occurs in the route map but there is no receive entry in the local VRF table, the packet is dropped if the packet destination is local.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip policy route-map** *map-tag*
5. **ip vrf receive** *vrf-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Device(config)# interface FastEthernet 0/1/0`	Configures an interface and enters interface configuration mode.
Step 4	**ip policy route-map** *map-tag* **Example:** `Device(config-if)# ip policy route-map map1`	Identifies a route map to use for policy routing on an interface. • The configuration example attaches the route map named map1 to the interface.
Step 5	**ip vrf receive** *vrf-name* **Example:** `Device(config-if)# ip vrf receive VRF-1`	Adds the IP addresses that are associated with an interface into the VRF table. • This command must be configured for each VRF that will be used for VRF selection.

Step 6	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Verifying the Configuration of Multi-VRF Selection Using Policy-Based Routing

To verify the configuration of the Multi-VRF Selection Using Policy-Based Routing (PBR) feature, perform the following steps. You can enter the commands in any order.

SUMMARY STEPS

1. **show ip access-list** [*access-list-number* | *access-list-name*]
2. **show route-map** [*map-name*]
3. **show ip policy**

DETAILED STEPS

Step 1 **show ip access-list** [*access-list-number* | *access-list-name*]

Verifies the configuration of match criteria for Multi-VRF Selection Using Policy-Based Routing. The command output displays three subnet ranges defined as match criteria in three standard access lists:

Example:

```
Device# show ip access-list

Standard IP access list 40
 10 permit 10.1.0.0, wildcard bits 0.0.255.255
Standard IP access list 50
 10 permit 10.2.0.0, wildcard bits 0.0.255.255
Standard IP access list 60
 10 permit 10.3.0.0, wildcard bits 0.0.255.255
```

Step 2 **show route-map** [*map-name*]

Verifies **match** and **set** commands within the route map:

Example:

```
Device# show route-map
```

The output displays the match criteria and set action for each route-map sequence. The output also displays the number of packets and bytes that have been policy routed per each route-map sequence.

Example:

```
Device# show route-map map1

route-map map1, permit, sequence 10
Match clauses:
Set clauses:
 ip next-hop vrf myvrf 10.5.5.5  10.6.6.6  10.7.7.7
 ip next-hop global 10.8.8.8  10.9.9.9
Policy routing matches: 0 packets, 0 bytes
Device# show route-map map2
route-map map2, permit, sequence 10
Match clauses:
Set clauses:
 vrf myvrf
Policy routing matches: 0 packets, 0 bytes
Device# show route-map map3
route-map map3, permit, sequence 10
Match clauses:
Set clauses:
 global
Policy routing matches: 0 packets, 0 bytes
```

The following **show route-map** command displays output from the **set ip vrf next-hop** command:

Example:

```
Device(config)# route-map test

Device(config-route-map)# set ip vrf myvrf next-hop
Device(config-route-map)# set ip vrf myvrf next-hop 192.168.3.2
Device(config-route-map)# match ip address 255 101
Device(config-route-map)# end
Device# show route-map

route-map test, permit, sequence 10
 Match clauses:
  ip address (access-lists): 101
 Set clauses:
  ip vrf myvrf next-hop 192.168.3.2
 Policy routing matches: 0 packets, 0 bytes
```

The following **show route-map** command displays output from the **set ip global** command:

Example:

```
Device(config)# route-map test
Device(config-route-map)# match ip address 255 101
Device(config-route-map)# set ip global next-hop 192.168.4.2
Device(config-route-map)# end
Device# show route-map

*May 25 13:45:55.551: %SYS-5-CONFIG_I: Configured from console by consoleout-map
route-map test, permit, sequence 10
 Match clauses:
  ip address (access-lists): 101
 Set clauses:
  ip global next-hop 192.168.4.2
 Policy routing matches: 0 packets, 0 bytes
```

Step 3 **show ip policy**

Verifies the Multi-VRF Selection Using Policy-Based Routing policy.

Example:

```
Device# show ip policy
```

The following **show ip policy** command output displays the interface and associated route map that is configured for policy routing:

Example:

```
Device# show ip policy

Interface                Route map
FastEthernet0/1/0        PBR-VRF-Selection
```

Configuration Examples for Multi-VRF Selection Using Policy-Based Routing

Example: Defining the Match Criteria for Multi-VRF Selection Using Policy-Based Routing

In the following example, three standard access lists are created to define match criteria for three different subnetworks. Any packets received on FastEthernet interface 0/1/0 will be policy routed through the PBR-VRF-Selection route map to the virtual routing and forwarding (VRF) that is matched in the same route-map sequence. If the source IP address of the packet is part of the 10.1.0.0/24 subnet, VRF1 will be used for routing and forwarding.

```
access-list 40 permit source 10.1.0.0 0.0.255.255
access-list 50 permit source 10.2.0.0 0.0.255.255
access-list 60 permit source 10.3.0.0 0.0.255.255
route-map PBR-VRF-Selection permit 10
 match ip address 40
 set vrf VRF1
 !
route-map PBR-VRF-Selection permit 20
 match ip address 50
 set vrf VRF2
 !
route-map PBR-VRF-Selection permit 30
 match ip address 60
 set vrf VRF3
 !
interface FastEthernet 0/1/0
 ip address 192.168.1.6 255.255.255.252
 ip policy route-map PBR-VRF-Selection
 ip vrf receive VRF1
 ip vrf receive VRF2
 ip vrf receive VRF3
```

Example: Configuring Multi-VRF Selection in a Route Map

The following example shows a **set ip vrf next-hop** command that applies policy-based routing to the virtual routing and forwarding (VRF) interface named myvrf and specifies that the IP address of the next hop is 10.0.0.2:

```
Device(config)# route-map map1 permit
Device(config)# set vrf myvrf
Device(config-route-map)# set ip vrf myvrf next-hop 10.0.0.2
Device(config-route-map)# match ip address 101
Device(config-route-map)# end
```

The following example shows a **set ip global** command that specifies that the device should use the next hop address 10.0.0.1 in the global routing table:

```
Device(config-route-map)# set ip global next-hop 10.0.0.1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco Master Command List, All Releases
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference
IP access list commands	*Cisco IOS Security Command Reference*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Multi-VRF Selection Using Policy-Based Routing

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 15: Feature Information for Multi-VRF Selection Using Policy-Based Routing

Feature Name	Releases	Feature Information
Multi-VRF Selection Using Policy-Based Routing (PBR)	12.2(33)SRB1 12.2(33)SXH1 12.4(24)T Cisco IOS XE Release 2.2	The Multi-VRF Selection Using Policy-Based Routing (PBR) feature allows a specified interface on a provider edge (PE) router to route packets to Virtual Private Networks (VPNs) based on packet length or match criteria defined in an IP access list. This feature and the MPLS VPN VRF Selection Based on Source IP Address feature can be configured together on the same interface In Cisco IOS Release 12.2(33)SRB1, this feature was introduced. In Cisco IOS Release 12.2(33)SXH1, support was added. In Cisco IOS Release 12.4(24)T, this feature was integrated. In Cisco IOS XE Release 2.2, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were modified: **set ip global next-hop** and **set ip vrf next-hop**.
IPv6 VRF-Aware PBR Next-hop Enhancement	15.2(2)S Cisco IOS XE Release 3.6S	In Cisco IOS Release 15.2(2)S, this feature was introduced. In Cisco IOS XE Release 3.6S, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced: **set ipv6 default next-hop**, **set ipv6 next-hop (PBR)**

Glossary

CE device—customer edge device. A device that is part of a customer network and that interfaces to a provider edge (PE) device.

Inherit-VRF routing—Packets arriving at a VRF interface are routed by the same outgoing VRF interface.

Inter-VRF routing—Packets arriving at a VRF interface are routed via any other outgoing VRF interface.

IP—Internet Protocol. Network layer protocol in the TCP/IP stack offering a connectionless internetwork service. IP provides features for addressing, type-of-service specification, fragmentation and reassembly, and security. Defined in RFC 791.

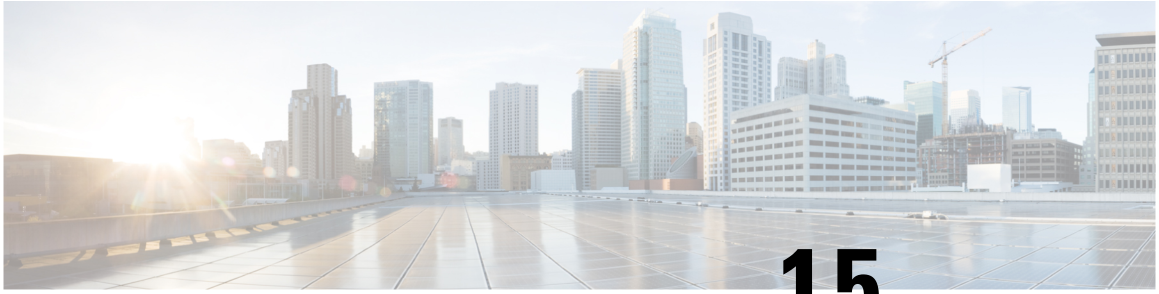
PBR—policy-based routing. PBR allows a user to manually configure how received packets should be routed.

PE device—provider edge device. A device that is part of a service provider's network and that is connected to a CE device. It exchanges routing information with CE devices by using static routing or a routing protocol such as BGP, RIPv1, or RIPv2.

VPN—Virtual Private Network. A collection of sites sharing a common routing table. A VPN provides a secure way for customers to share bandwidth over an ISP backbone network.

VRF—A VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table.

VRF-lite—A feature that enables a service provider to support two or more VPNs, where IP addresses can be overlapped among the VPNs.



CHAPTER 15

MPLS VPN VRF Selection Based on a Source IP Address

The MPLS VPN VRF Selection Based on a Source IP Address feature allows a specified interface on a provider edge (PE) device to route packets to different Virtual Private Networks (VPNs) based on the source IP address of the packet. This feature is an improvement over using a policy-based device to route packets to different VPNs.

The MPLS VPN VRF Selection Based on a Source IP Address feature allows packets arriving on an interface to be switched into the appropriate virtual routing and forwarding (VRF) table based upon the source IP address of the packets. Once the packets have been “selected” into the correct VRF routing table, they are processed normally, based on the destination address and forwarded through the rest of the Multiprotocol Label Switching (MPLS) VPN.

In most cases, this feature is a "one way" feature; it works on packets coming from the end users to the PE device

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for MPLS VPN VRF Selection Based on a Source IP Address

- Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) must be enabled in the provider network.
- The Cisco 12000 Internet Router Series must contain one of the following line cards:

Engine 0:

- 1-port OC-12 POS
- 4-port OC-3 POS
- 6- and 12- port DS3

Engine 2:

- 1-port OC-48 POS
- 4-port OC-12 POS
- 8- and 16-port OC-3 POS
- 3-port Gigabit Ethernet

Engine 3:

- 4-port OC-12c/STM-4c POS ISE
- 4-port CHOC-12 ISE
- 1-port OC-48c POS ISE
- 1-port CHOC-48 ISE
- 4-, 8-, and 16-port OC-3c POS ISE

Engine 4:

- 4-port OC-48 POS
- OC-192 E4+ POS
- 1-port 10-Gigabit Ethernet (E4+)
- Modular Gigabit Ethernet (E4+)

Restrictions for MPLS VPN VRF Selection Based on a Source IP Address

- The MPLS VPN VRF Selection Based on a Source IP Address feature is supported only in Service Provider (-p-) images.
- The Cisco software must support Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs), and the provider network must have MPLS Label Distribution Protocol (LDP) installed and running.
- The MPLS VPN VRF Selection Based on a Source IP Address feature is a unidirectional feature and can only be used from a customer (IP-based) network for a connection to a provider (MPLS-based) network and cannot be used from a provider network to a customer network.
- Subnet masks should be kept as short as possible for the MPLS VPN VRF Selection Based on a Source IP Address criteria for Engine 2 line cards. The performance of this feature can degrade with longer subnet masks (/24 or /32, for example).
- Cisco Express Forwarding must be enabled on any interfaces that have the MPLS VPN VRF Selection Based on a Source IP Address feature enabled. Distributed Cisco Express Forwarding is enabled by default on Cisco 12000 Series Internet routers.
- An IP traceroute command from an MPLS VPN VRF Selection customer edge (CE) device to a typical MPLS VPN VRF CE device works as expected. However, an IP traceroute command from a typical MPLS VPN VRF CE device to an MPLS VPN VRF Selection CE device might fail to show all the relevant hop information across the core.

Information About MPLS VPN VRF Selection Based on a Source IP Address

VRF Selection Process

The MPLS VPN VRF Selection Based on a Source IP Address feature uses the process described in this section to route packets from the customer networks to the provider edge (PE) device and into the provider network.

A two-table lookup mechanism is used at the ingress interface of the PE device to determine the routing and forwarding of packets coming from the customer networks, which use IP protocols, to the Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs), which use MPLS protocols.

- The first table, the VRF Selection table, is used to compare the source IP address of the packet with a list of IP addresses in the table. Each IP address in the table is associated with an MPLS VPN. If a match is found between the source IP address of the packet and an IP address in the VRF Selection table, the packet is routed to the second table (the VRF table) or the routing table for the appropriate VPN.

  If no match is found in the table for the source IP address of the packet, the packet is either routed via the global routing table used by the PE device (this is the default behavior), or is dropped.

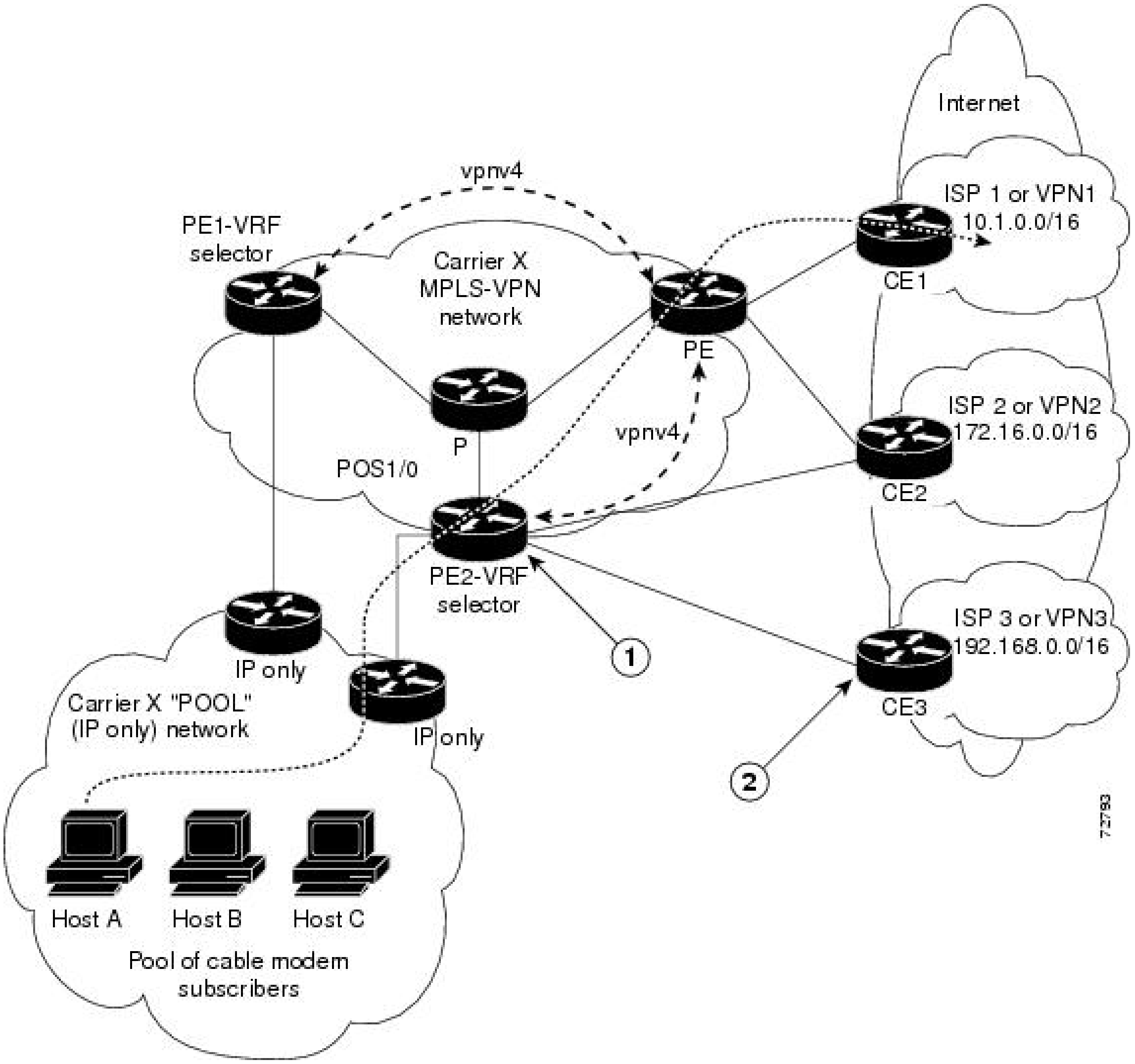
- The second table, the VRF table (also known as the VPN routing table), contains the virtual routing and forwarding information for the specified VPN and is used to forward the selected VPN traffic to the correct MPLS label switched path (LSP) based upon the destination IP address of the packet.

The VRF Selection process removes the association between the VPN and the interface and allows more than one MPLS VPN to be associated with the interface.

VRF Selection Examples

Here is an example of the MPLS VPN VRF Selection Based on a Source IP Address feature. It is based on a network carrier that allows subscribers to the carrier to choose from multiple Internet service providers (ISPs) for Internet access. The figure below provides an example of the MPLS VPN VRF Selection Based on a Source IP Address feature with an IP-based host network, an Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) , and three ISPs connected to the MPLS VPN network.

Figure 15: VRF Selection Implementation Example


Internet
vpnv4
PE1-VRF selector
Carrier X MPLS-VPN network
ISP 1 or VPN1 10.1.0.0/16
CE1
PE
P
vpnv4
ISP 2 or VPN2 172.16.0.0/16
CE2
POS1/0
PE2-VRF selector
ISP 3 or VPN3 192.168.0.0/16
CE3
IP only
Carrier X "POOL" (IP only) network
IP only
1
2
72793
Host A
Host B
Host C
Pool of cable modem subscribers

1	PE2 is acting as a VRF selector and as a typical MPLS VPN PE device to CE2 and CE3.	2	ISPs 1 to 3 provide a list of IP addresses to Carrier X so that each host in the “POOL” network can be properly addressed. This host addressing would most likely be done by means of the DHCP or DNS services of Carrier X.

In the figure above, Carrier X represents the network carrier; Host A, Host B and Host C represent the carrier subscribers; and ISP 1, ISP 2 and ISP 3 represent the ISPs.

The figure illustrates a packet traveling from Host A to ISP 1. The dashed line represents the travel of the packet.

Host A chooses ISP 1 to use as its ISP. Carrier X will provide an IP address to Host A that falls within the range of the ISP 1 registered network addresses. Based upon this IP address allocation, the VRF Selection criteria are set.

The POOL network, by using default routes, forwards traffic from the Carrier X IP-based (POOL) network to the Carrier X MPLS-based VPN network. The MPLS VPN network forwards (shunts) the traffic from Host A into the correct VPN, which is VPN 1 (ISP 1), by using the VRF Selection-enabled device PE2.

To enable the MPLS VPN VRF Selection Based on a Source IP Address feature on the devices PE1 and PE2, enter the following commands:

```
Device(config)# vrf selection source 10.1.0.0 255.255.0.0 vrf vpn1
Device(config)# vrf selection source 172.16.0.0 255.255.0.0 vrf vpn2
Device(config)# interface POS1/0
Device(config-if)# description Link to CE POS1/0
Device(config-if)# ip vrf select source
```

Traffic coming from the ISPs to the hosts (in the example, traffic traveling from the ISPs on the right to the hosts on the left) is not affected by the MPLS VPN VRF Selection Based on a Source IP Address feature and does not have to be returned via an MPLS path. This traffic can return via the shortest available IP path.

Another example of MPLS VPN VRF Selection Based on a Source IP Address feature in use might involve a cable modem termination system (CMTS). If the owner of the CMTS wants to allow cable modem subscribers to choose their ISP from a group of ISPs, the MPLS VPN VRF Selection Based on a Source IP Address feature provides a fast and scalable solution.

VRF Selection is a Unidirectional Feature

In the figure above, the end users are typical Internet home users. If the MPLS VPN VRF Selection Based on a Source IP Address feature were a two-way (bidirectional) feature, traffic coming from the Internet service providers (ISPs) to the hosts would be required to use only the provider edge (PE) devices that have MPLS VPN VRF Selection Based on a Source IP Address feature enabled, which might cause performance issues.

When traffic from the POOL network goes through the carrier network to the ISP networks for Internet access, the traffic in the carrier network must be forwarded by means of Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) paths, because the VRF Selection-enabled device has “selected” the traffic into the correct MPLS VPN.

Traffic from the ISP networks to the POOL network does not have to use MPLS VPN paths in the carrier network and can use the path that seems most efficient to return to the POOL network. This traffic can use a path that uses either MPLS or IP for routing and forwarding and does not have to travel via an MPLS VPN.

Traffic from the ISP networks to the POOL networks can be forwarded by the global routing table used by every interface. One way to accomplish this is to enter virtual routing and forwarding (VRF) static routes on the PE device interfaces connected to the ISPs. The VRF static routes would route traffic from the ISPs to the carrier network.

Establishing static VRF routes allows traffic from the ISPs to enter the carrier network as traffic that can only be routed by using the global routing table toward the POOL network.

If the ISPs are not providing global host address space, or the MPLS VPN VRF Selection Based on a Source IP Address feature is not being used to route Internet traffic, the PE interfaces connected to the ISPs must be placed into a VRF. If the PE interfaces are using VRFs for routing traffic from the ISPs, all traffic from the ISPs to the hosts through the carrier network would be forwarded by MPLS VPN paths, and performance would not be as good as if IP forwarding were used.

Normal IP-based VPN operations, such as populating the routing information base (RIB) and forwarding information base (FIB) from a routing protocol such as Border Gateway Protocol (BGP), are used to route and forward packets within the various VPNs in the customer networks. The provider network uses MPLS-based routing protocols to perform VPN routing and forwarding inside the provider network.

Conditions Under Which VRF Selection Becomes Bidirectional

Forwarding of traffic from the carrier network to the POOL network by using the global routing table is possible only if the Internet service providers (ISPs) have provided registered IP address space for all of the subscribed users within the POOL network.

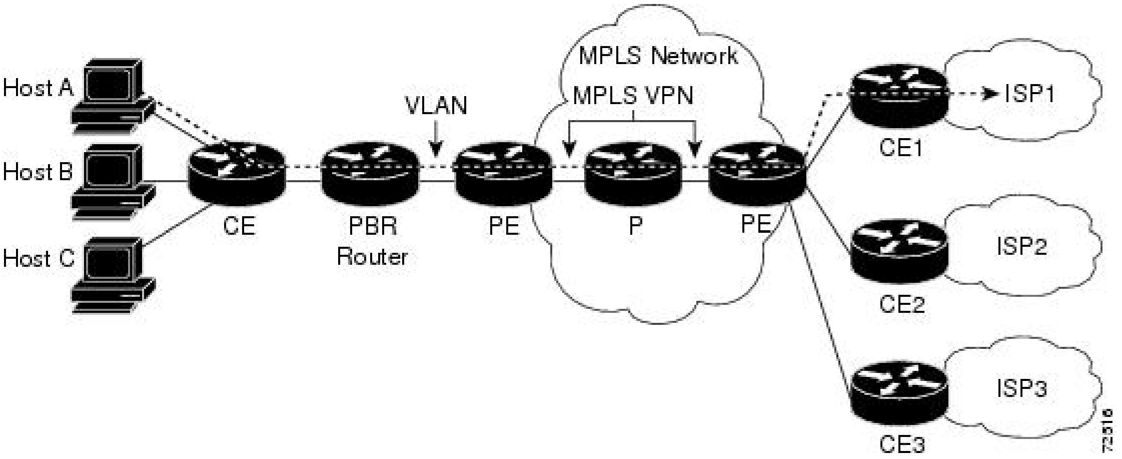
If the POOL network uses IP addresses that are not globally routeable and are designed for a nonconnected enterprise (defined by RFC1918), the MPLS VPN VRF Selection Based on a Source IP Address feature becomes bidirectional. All traffic being sent and received by the host would have to travel via a device that has the MPLS VPN VRF Selection Based on a Source IP Address feature enabled. The POOL network cannot be addressed with overlapping address space, regardless of the type of address space being used.

Advantages of VRF Selection over Per-Interface IP VPN Configuration

The MPLS VPN VRF Selection Based on a Source IP Address feature removes the association between a Virtual Private Network (VPN) and an interface. Before the MPLS VPN VRF Selection Based on a Source IP Address feature was introduced, the following implementation was used to route outgoing Multiprotocol Label Switching (MPLS) VPN packets to different destinations:

- A policy-based device (PBR) is attached to the customer edge (CE) device.
- The egress side of the PBR device side has VLANs connected to a provider edge (PE) device.
- The PBR device uses a policy-based route map to select the correct output (VLAN) interface, and each VLAN is under a specific VRF. The figure below illustrates a sample configuration in which a PBR device is used for routing MPLS packets to different destinations.

Figure 16: Implementation of Multiple VPNs Before VRF Selection



The following limitations apply to PBR-based solutions that use this implementation:

- Policy routing and MPLS VPN functions cannot be performed on the same platform. Integration into a single platform is critical for manageability and support.
- Each VRF is limited to one VPN per interface, which limits scalability.
- There is no network redundancy.
- The PBR is the only point of connection for all the networks attached to the PBR. The capacity and the performance capabilities of the PBR device are critical.
- There is no diversity in the connectivity to the networks.
- Every network is required to connect to every PBR. If every network is not connected to every PBR, packets from the end user to the PBR are dropped because the PBR has no way of switching the IP traffic properly.
- Adding multiple PBRs that are interconnected introduces more network policy-routed hops.

The MPLS VPN VRF Selection Based on a Source IP Address feature addresses the limitations of and problems with using a PBR for packet routing and forwarding.

Benefits of MPLS VPN VRF Selection Based on a Source IP Address

The following are benefits to using the MPLS VPN VRF Selection Based on a Source IP Address method of VPN routing and forwarding.

- Association of VPN to interface is removed—The MPLS VPN VRF Selection Based on a Source IP Address feature removes the association between a Virtual Private Network (VPN) and an interface, thus allowing packets from the host network to the provider network to have more than one VPN available per interface.
- Access to every customer network is possible from every provider edge (PE) device in the provider network-Access points to each network can be established at any MPLS PE device and can be made redundant by connections to multiple PE devices (for example, the CE2 device in the figure above).

- Multiple points in the provider network can be used for VPN routing and forwarding—MPLS VPNs, like IP, are connectionless. Any PE device, whether MPLS VPN VRF Selection Based on a Source IP Address feature is enabled or not, is capable of carrying VRF Selection traffic from the MPLS network out to the customer edge (CE) devices.

How to Configure MPLS VPN VRF Selection Based on a Source IP Address

Configuring VRF Selection

To add a source IP address to a VRF Selection table, complete the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf selection source** *source-ip-address source-ip-mask* **vrf** *vrf-name*
4. **ip vrf select source**
5. **ip vrf receive** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf selection source** *source-ip-address source-ip-mask* **vrf** *vrf-name* **Example:** `Device(config)# vrf selection source 172.16.0.0 255.255.0.0 vrf test`	Populates a single source IP address, or range of source IP addresses, to a VRF Selection table.
Step 4	**ip vrf select source** **Example:** `Device(config-if)# ip vrf select source`	Enables the MPLS VPN VRF Selection Based on a Source IP Address feature on an interface.

	Command or Action	Purpose
Step 5	**ip vrf receive** *vrf-name* **Example:** `Device(config-if)# ip vrf receive red`	Adds all the IP addresses that are associated with an interface into a VRF table.

Establishing IP Static Routes for a VRF Instance

Traffic coming from the Internet service providers (ISPs) to the hosts does not require the use of the Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) paths; this traffic can use the shortest IP route back to the host.

VPN static routes for traffic returning to the customer networks are only necessary if VPN traffic returning to the customer networks is being forwarded from the MPLS VPN VRF Selection Based on a Source IP Address interface. The remote provider edge (PE) device could also be configured to route return traffic to the customer networks directly by using the global routing table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route vrf** *vrf-name prefix mask* [*next-hop-address*] {[**interface** *interface-number*}] [**global**] [**distance**] [**permanent**] [**tag**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route vrf** *vrf-name prefix mask* [*next-hop-address*] {[**interface** *interface-number*}] [**global**] [**distance**] [**permanent**] [**tag**] **Example:** `Device(config-if)# ip route vrf vpn1 172.16.0.0 255.255.0.0 POS1/0`	Establishes static routes for a VRF.

Verifying VRF Selection

Enter the **show ip route vrf** command in privileged EXEC mode to display the IP routing table associated with a virtual routing and forwarding (VRF) instance. This example shows the IP routing table associated with the VRF vrf1:

```
Device#  show ip route vrf vpn1

Routing Table: vpn1
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       I - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
Gateway of last resort is not set
B    10.0.0.0/8 [200/0] via 10.10.10.10, 00:00:37
     172.16.0.0/16 is subnetted, 1 subnets
B       10.19.0.0 [200/0] via 10.10.10.10, 00:00:37
     10.0.0.0/32 is subnetted, 1 subnets
B       10.14.14.14 [200/0] via 10.10.10.10, 00:00:37
     10.0.0.0/32 is subnetted, 1 subnets
S       10.15.15.15 [1/0] via 10.0.0.1, POS1/1
```

Troubleshooting Tips

Enter the **debug vrf select** command to enable debugging for the MPLS VPN VRF Selection Based on a Source IP Address feature.

Note The **debug vrf select** command can cause many messages to be logged when you change the configuration and when switching occurs.

If you attempt to configure a nonexisting MPLS VPN VRF Selection Based on a Source IP Address table, the following error appears:

```
Device(config)# vrf selection source 172.16.0.0 255.255.0.0 vrf VRF_NOEXIST
VRF Selection: VRF table VRF_NOEXIST does not exist.
```

If you attempt to remove an MPLS VPN VRF Selection Based on a Source IP Address entry that does not exist, the following error appears:

```
Device(config)# no vrf selection source 172.16.0.0 255.255.0.0 vrf VRF1
VRF Selection: Can't find the node to remove.
```

If you attempt to configure a duplicate IP address and subnet mask for an MPLS VPN VRF Selection Based on a Source IP Address entry, the following error appears:

```
Device(config)# vrf selection source 172.16.0.0 255.0.0.0 vrf VRF_AOL
Device(config)# vrf selection source 172.16.0.0 255.0.0.0 vrf VRF_AOL

VRF Selection: duplicate address and mask configured.
```

If an inconsistent IP address and mask are used for an MPLS VPN VRF Selection Based on a Source IP Address entry, the following error appears:

```
Device(config)# vrf selection source 172.16.2.1 255.255.255.0 vrf red
% Inconsistent address and mask
```

If you attempt to configure a VRF instance on an interface that has MPLS VPN VRF Selection Based on a Source IP Address already configured, the following error appears:

```
Device(config-if)# ip vrf select source
Device(config-if)# ip vrf forward red
% Can not configure VRF if VRF Select is already configured
```

To enable VRF, first remove MPLS VPN VRF Selection Based on a Source IP Address from the interface. If you attempt to configure an MPLS VPN VRF Selection Based on a Source IP Address entry on an interface that has VRF already configured, the following error appears:

```
Device(config-if)# ip vrf forward red
Device(config-if)# ip vrf select red
% Can not configure VRF Select if interface is under a non-global VRF
```

To enable the MPLS VPN VRF Selection Based on a Source IP Address feature, first remove VRF from the interface

Configuration Examples for MPLS VPN VRF Selection Based on a Source IP Address

Example: Enabling MPLS VPNs

The following example shows how to enable the device to accept MPLS VPNs:

```
Device(config)# mpls label protocol ldp
Device(config)# interface loopback0
Device(config-if)# ip address 10.13.13.13 255.255.255.255
Device(config-if)# no ip directed-broadcast
```

Example: Creating a VRF Routing Table

The following example shows how to create two VRF Selection tables (vpn1 and vpn2):

```
Device(config)# ip vrf vpn1
Device(config-vrf)# rd 1000:1
Device(config-vrf)# route-target export 1000:1
Device(config-vrf)# route-target import 1000:1
Device(config-vrf)# exit
Device(config)# ip vrf vpn2
Device(config-vrf)# rd 1000:2
Device(config-vrf)# route-target export 1000:2
Device(config-vrf)# route-target export 1000:2
```

Example: Defining VRF Selection Entries

The following example shows two entries (vpn1 and vpn2) being defined in the VRF Selection table. In this example, packets with the source address of 10.16.0.0 will be routed to the VRF vpn1, and packets with the source address of 10.17.0.0 will be routed to the VRF vpn2:

```
Device(config)# vrf selection source 10.16.0.0 255.255.0.0 vrf vpn1
Device(config)# vrf selection source 10.17.0.0 255.255.0.0 vrf vpn2
```

Example: Defining IP Static Routes for a VRF

The following example shows IP static routes being created for two VRFs (vpn1 and vpn2) for the POS1/0 interface:

```
Device(config)# ip route vrf vpn1 10.16.0.0 255.255.0.0 POS1/0
Device(config)# ip route vrf vpn2 10.17.0.0 255.255.0.0 POS1/0
```

Example: Configuring an Interface for VRF Selection

The following example shows the POS1/0 interface being configured for the MPLS VPN VRF Selection Based on a Source IP Address feature and the configured IP address (31.0.0.1) being added to the VRFs vpn1 and vpn2 as connected routes:

```
Device(config)# interface POS1/0
Device(config-if)# description Link to CE1 POS1/0 (eng2)
Device(config-if)# ip vrf select source
Device(config-if)# ip vrf receive vpn1
Device(config-if)# ip vrf receive vpn2
Device(config-if)# ip address 10.0.0.1 255.0.0.0
Device(config-if)# no ip directed-broadcast
Device(config-if)# load-interval 30
Device(config-if)# crc 32
Device(config-if)# end
```

Example: Configuring a BGP Device for VRF Selection

A device that is enable with the MPLS VPN VRF Selection Based on a Source IP Address feature requires an MPLS VPN BGP configuration. The following example configures a device that is using BGP for the MPLS VPN VRF Selection Based on a Source IP Address feature:

```
Device(config)# router bgp 1000
Device(config-router)# no bgp default ipv4-unicast
Device(config-router)# bgp log-neighbor-changes
Device(config-router)# timers bgp 10 30
Device(config-router)# neighbor 10.11.11.11 remote-as 1000
Device(config-router)# neighbor 10.11.11.11 update-source Loopback0
Device(config-router)# no auto-summary
Device(config-router)# address-family vpnv4
Device(config-router-af)# neighbor 10.11.11.11 activate
Device(config-router-af)# neighbor 10.11.11.11 send-community extended
Device(config-router-af)# exit-address-family
Device(config-router)# address-family ipv4 vrf vpn2
Device(config-router-af)# redistribute static
Device(config-router-af)# no auto-summary
Device(config-router-af)# no synchronization
Device(config-router-af)# exit-address-family
Device(config-router)# address-family ipv4 vrf vpn1
Device(config-router-af)# redistribute static
Device(config-router-af)# no auto-summary
Device(config-router-af)# no synchronization
Device(config-router-af)# exit-address-family
```

Example: Configuring a VRF to Eliminate Unnecessary Packet Forwarding

If a packet arrives at an interface that has the MPLS VPN VRF Selection Based on a Source IP Address feature enabled, and the packet source IP address does not match any VRF selection definition, that packet will be forwarded by means of the global routing table. This default behavior could cause problems if IP address spoofing is being implemented. Unnecessary traffic could be forwarded by the global routing table. To eliminate this unnecessary routing of packets, create a VRF selection definition that will forward all unknown incoming traffic to a null interface.

The following configuration causes all traffic not matching a more specific VRF selection definition to be routed to the Null0 interface, thus causing the packets to be dropped.

```
Device(config)# ip vrf VRF_DROP
Device(config-vrf)# rd 999:99
Device(config-vrf)# route-target export 999:99
Device(config-vrf)# route-target import 999:99
Device(config-vrf)# exit
Device(config)# vrf selection source 0.0.0.0 0.0.0.0 vrf VRF_DROP
Device(config)# ip route vrf VRF_DROP 0.0.0.0 0.0.0.0 Null0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco Master Command List, All Releases
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS VPN VRF Selection Based on a Source IP Address

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 16: Feature Information for MPLS VPN VRF Selection Based on a Source IP Address

Feature Name	Releases	Feature Information
MPLS VPN VRF Selection Based on a Source IP Address	12.0(22)S 12.0(23)S 12.0(24)S 12.0(26)S 12.2(18)S 12.2(27)SBC	The MPLS VPN VRF Selection Based on a Source IP Address feature allows a specified interface on a provider edge (PE) device to route packets to different Virtual Private Networks (VPNs) based on the source IP address of the packet. This feature is an improvement over using a policy-based device to route packets to different VPNs. In Cisco IOS Release 12.0(22)S, this feature was introduced on the Cisco 12000 Series Internet Router. In Cisco IOS Release 12.0(23)S, this feature was updated to support the 1-port 10-Gigabit Ethernet (E4+), 3-port Gigabit Ethernet , and the Modular Gigabit Ethernet (E4+) line cards. In Cisco IOS Release 12.0(24)S, support for the Cisco 12000 Series Internet Router engine 3 was added. In Cisco IOS Release 12.0(26)S, this feature was implemented on the Cisco 7200 and 7500 series routers. In Cisco IOS Release 12.2(18)S, this feature was implemented on the Cisco 7304 router. In Cisco IOS Release 12.2(27)SBC, this feature was integrated. The following commands were introduced or modified: **ip vrf receive**, **ip vrf select source**, **vrf selection source**.

CHAPTER 16

MPLS VPN VRF Selection Using Policy-Based Routing

The MPLS VPN VRF Selection Using Policy-Based Routing feature is an extension of the MPLS VPN VRF Selection Based on Source IP Address feature. This feature introduces a policy-based routing (PBR) mechanism to classify and forward Virtual Private Network (VPN) traffic based on multiple VPN routing and forwarding (VRF) selection match criteria.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for MPLS VPN VRF Selection Using Policy-Based Routing

- The device must support policy-based routing (PBR). For platforms that do not support PBR, use the “MPLS VPN VRF Selection Based on Source IP Address” feature.

- A virtual routing and forwarding (VRF) instance must be defined prior to the configuration of this feature. An error message is displayed on the console if no VRF exists.

- Before you configure the MPLS VPN VRF Selection Using Policy-Based Routing feature, make sure that the VRF and associated IP address are already defined.
- This document assumes that multiprotocol Border Gateway Protocol (mBGP), Multiprotocol Label Switching (MPLS), and Cisco Express Forwarding are enabled in your network.

Restrictions for MPLS VPN VRF Selection Using Policy-Based Routing

- The MPLS VPN VRF Selection Using Policy-Based Routing feature is supported only in service provider (-p-) images.
- The MPLS VPN VRF Selection Using Policy-Based Routing feature can coexist with the MPLS VPN VRF Selection Based on Source IP address feature on the same device, but these features cannot be configured together on the same interface. This is designed behavior to prevent virtual routing and forwarding (VRF) table selection conflicts that could occur if these features were misconfigured together. An error message is displayed on the console if you attempt to configure the **ip vrf select source** and the **ip policy route-map** commands on the same interface.
- Protocol Independent Multicast (PIM) and multicast packets do not support policy-based routing (PBR) and cannot be configured for a source IP address that is a match criterion for this feature.
- The MPLS VPN VRF Selection Using Policy-Based Routing feature cannot be configured with IP prefix lists.

Information About MPLS VPN VRF Selection Using Policy-Based Routing

Introduction to MPLS VPN VRF Selection Using Policy-Based Routing

The MPLS VPN VRF Selection Using Policy-Based Routing feature is an extension of the MPLS VPN VRF Selection Based on Source IP Address feature. The policy-based routing (PBR) implementation of the virtual routing and forwarding (VRF) selection feature allows you to policy route Virtual Private Network (VPN) traffic based on match criteria. Match criteria are defined in an IP access list or based on packet length. The following match criteria are supported in Cisco software:

- IP access lists—Define match criteria based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. All IP access-list configuration options in Cisco software can be used to define match criteria.
- Packet lengths—Define match criteria based on the length of a packet in bytes. The packet length filter is defined in a route map with the **match length** route-map configuration command.

Policy routing is defined in the route map. The route map is applied to the incoming interface with the **ip policy route-map** interface configuration command. An IP access list is applied to the route map with the **match ip address** route-map configuration command. Packet length match criteria are applied to the route map with the **match length** route-map configuration command. The set action is defined with the **set vrf** route-map configuration command. The match criteria are evaluated, and the appropriate VRF is selected by the set clause. This combination allows you to define match criteria for incoming VPN traffic and policy route VPN packets out to the appropriate VRF.

Policy-Based Routing Set Clauses Overview

When you are configuring policy-based routing (PBR), the following four set clauses can be used to change normal routing and forwarding behavior:

- **set default interface**
- **set interface**
- **set ip default next-hop**
- **set ip next-hop**

Configuring any of the set clauses will overwrite normal routing forwarding behavior of a packet.

The MPLS VPN VRF Selection Using Policy-Based Routing feature introduces the fifth set clause that can be used to change normal routing and forwarding behavior. The **set vrf** command is used to select the appropriate virtual routing and forwarding (VRF) instance after the successful match occurs in the route map.

Match Criteria for Policy-Based Routing VRF Selection Based on Packet Length

The match criteria for policy-based routing (PBR) virtual routing and forwarding (VRF) route selection are defined in an access list. Standard and named access lists are supported. Match criteria can also be defined based on the packet length using the **match length** route-map configuration command. This configuration option is defined entirely within a route map.

How to Configure MPLS VPN VRF Selection Using Policy-Based Routing

Configuring Policy-Based Routing VRF Selection with a Standard Access List

Use the following commands to create a standard access list and define the policy-based routing (PBR) virtual routing and forwarding (VRF) route selection match criteria in it in order to permit or deny the transmission of VPN traffic data packets.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} *source-addr* [*source-wildcard*] [**log**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} *source-addr* [*source-wildcard*] [**log**] **Example:** `Device(config)# access-list 40 permit 10.1.0.0/24 0.0.0.255`	Creates an access list and defines the match criteria for the route map. • Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access-list filtering options. Named, numbered, standard, and extended access lists are supported. All IP access list configuration options in Cisco software can be used to define match criteria. • The example creates a standard access list numbered 40. This filter will permit traffic from any host with an IP address in the 10.1.0.0/24 subnet.

Configuring Policy-Based Routing VRF Selection with a Named Access List

Use the following commands to define the policy-based routing (PBR) virtual routing and forwarding (VRF) route selection match criteria in a named access list in order to permit or deny the transmission of Virtual Private Network (VPN) traffic data packets.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*]
4. [*sequence-number*] {**permit** | **deny**} *protocol source-addr source-wildcard destination-addr destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]

DETAILED STEPS

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

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] **Example:** `Device(config)# ip access-list extended NAMEDACL`	Specifies the IP access list type and enters the corresponding access-list configuration mode. • A standard, extended, or named access list can be used.
Step 4	[*sequence-number*] {**permit** | **deny**} *protocol source-addr source-wildcard destination-addr destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# permit ip any any option any-options`	Defines the criteria for which the access list will permit or deny packets. • Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access-list filtering options. Named, numbered, standard, and extended access lists are supported. All IP access-list configuration options in Cisco software can be used to define match criteria. • The example creates a named access list that permits any configured IP option.

Configuring Policy-Based Routing VRF Selection in a Route Map

Use the following commands to configure the VRF through which the outbound Virtual Private Network (VPN) packets will be policy routed in order to permit or deny the transmission of VPN traffic data packets.

Incoming packets are filtered through the match criteria that are defined in the route map. After a successful match occurs, the **set vrf** command configuration determines the VRF through which the outbound VPN packets will be policy routed.

Before you begin

- The virtual routing and forwarding (VRF) instance must be defined prior to the configuration of the route map; otherwise, an error message is displayed on the console.
- A receive entry must be added to the VRF selection table with the **ip vrf receive** command. If a match and set operation occurs in the route map but there is no receive entry in the local VRF table, the packet will be dropped if the packet destination is local.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]

4. Do one of the following:
   - **match ip address** {*acl-number* [*acl-number* ... | *acl-name* ...] | *acl-name* [*acl-name* ... | *acl-number* ...]}
   -
   - **match length** *minimum-length maximum-length*
5. **set vrf** *vrf-name*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map map1 permit 10`	Enters route map configuration mode. Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing.

Step 4	Do one of the following:
	• **match ip address** {*acl-number* [*acl-number* ... | *acl-name* ...] | *acl-name* [*acl-name* ... | *acl-number* ...]}
	•
	• **match length** *minimum-length maximum-length*
	Example:
	`Device(config-route-map)# match ip address 1`
	Example:
	`Device(config-route-map)# match length 3 200`
		Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets.
		• IP access lists are supported.
		• The example configures the route map to use standard access list 1 to define match criteria.
		or
		Specifies the Layer 3 packet length in the IP header as a match criterion in a class map.
		• The example configures the route map to match packets that are 3 to 200 bytes in size.

Step 5	**set vrf** *vrf-name* **Example:** `Device(config-route-map)# set vrf map1`	Defines which VRF to route VPN packets that are successfully matched in the same route map sequence for policy-based routing (PBR) VRF selection. • The example policy routes matched packets out to the VRF named map1.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-route-map)# exit`	Returns to global configuration mode.

Configuring Policy-Based Routing on the Interface

Use the following commands to filter incoming Virtual Private Network (VPN) traffic data packets. Incoming packets are filtered through the match criteria that are defined in the route map.

The route map is applied to the incoming interface. The route map is attached to the incoming interface with the **ip policy route-map** global configuration command.

Note

- The MPLS VPN VRF Selection Using Policy-Based Routing feature can coexist with the MPLS VPN VRF Selection Based on Source IP address feature on the same device, but the two features cannot be configured together on the same interface. This is designed behavior to prevent virtual routing and forwarding (VRF) table selection conflicts that could occur if these features were misconfigured together. An error message is displayed on the console if you attempt to configure the **ip vrf select source** and the **ip policy route-map** commands on the same interface.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip policy route-map** *map-tag*
5. **ip vrf receive** *vrf-name*
6. **exit**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**interface** *type number* [*name-tag*] **Example:** `Device(config)# interface FastEthernet 0/1/0`	Configures an interface and enters interface configuration mode.
Step 4	**ip policy route-map** *map-tag* **Example:** `Device(config-if)# ip policy route-map map1`	Identifies a route map to use for policy routing on an interface. • The configuration example attaches the route map named map1 to the interface.
Step 5	**ip vrf receive** *vrf-name* **Example:** `Device(config-if)# ip vrf receive VRF1`	Adds the IP addresses that are associated with an interface into the VRF table. • This command must be configured for each VRF that will be used for VRF selection.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.

Configuring IP VRF Receive on the Interface

Use the following commands to insert the IP address of an interface as a connected route entry in a virtual routing and forwarding (VRF) routing table. This will prevent dropped packets.

The source IP address must be added to the VRF selection table. VRF selection is a one-way (unidirectional) feature. It is applied to the incoming interface. If a match and set operation occurs in the route map but there is no VRF receive entry in the local VRF table, the packet will be dropped if the packet destination is local.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip policy route-map** *map-tag*
5. **ip vrf receive** *vrf-name*
6. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Device(config)# interface FastEthernet 0/1/0`	Configures an interface and enters interface configuration mode.
Step 4	**ip policy route-map** *map-tag* **Example:** `Device(config-if)# ip policy route-map map1`	Identifies a route map to use for policy routing on an interface. • The configuration example attaches the route map named map1 to the interface.
Step 5	**ip vrf receive** *vrf-name* **Example:** `Device(config-if)# ip vrf receive VRF1`	Adds the IP addresses that are associated with an interface into the VRF table. • This command must be configured for each VRF that will be used for VRF selection.

Step 6	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Verifying the Configuration of the MPLS VPN VRF Selection Using Policy-Based Routing

SUMMARY STEPS

1. **enable**
2. **show ip access-list** [*access-list-number* | *access-list-name*]
3. **show route-map** [*map-name*]
4. **show ip policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip access-list** [*access-list-number* | *access-list-name*]	Displays the contents of all current IP access lists.

	Command or Action	Purpose
	Example: `Device# show ip access-list`	- This command is used to verify the match criteria that are defined in the access list. Both named and numbered access lists are supported.
Step 3	**show route-map** [*map-name*] **Example:** `Device# show route-map`	Displays all route maps configured or only the one specified. - This command is used to verify match and set clauses within the route map.
Step 4	**show ip policy** **Example:** `Device# show ip policy`	Displays the route map used for policy routing. - This command can be used to display the route map and the associated interface.

Configuration Examples for MPLS VPN VRF Selection Using Policy-Based Routing

Example: Defining Policy-Based Routing VRF Selection in an Access List

In the following example, three standard access lists are created to define match criteria for three different subnets. Any packets received on the FastEthernet 0/1/0 interface will be policy routed through the policy-based routing (PBR) VRF selection route map to the virtual routing and forwarding (VRF) instancer that is matched in the same route map sequence. If the source IP address of the packet is part of the 10.1.0.0/24 subnet, VRF1 will be used for routing and forwarding.

```
access-list 40 permit 10.1.0.0 0.0.255.255
access-list 50 permit 10.2.0.0 0.0.255.255
access-list 60 permit 10.3.0.0 0.0.255.255
route-map PBR-VRF-Selection permit 10
 match ip address 40
 set vrf VRF1
 !
route-map PBR-VRF-Selection permit 20
 match ip address 50
 set vrf VRF2
 !
route-map PBR-VRF-Selection permit 30
 match ip address 60
 set vrf VRF3
 !
interface FastEthernet0/1/0
 ip address 10.1.0.0/24 255.255.255.252
 ip policy route-map PBR-VRF-Selection
 ip vrf receive VRF1
 ip vrf receive VRF2
 ip vrf receive VRF3
```

Examples: Verifying VRF Selection Using Policy-Based Routing

The following verification examples show defined match criteria and route-map policy configuration.

Example: Verifying Match Criteria

To verify the configuration of match criteria for policy-based routing (PBR) VRF selection, use the **show ip access-list** command.

The following **show ip access-list** command output displays three subnet ranges defined as match criteria in three standard access lists:

```
Device# show ip access-list

Standard IP access list 40
    10 permit 10.1.0.0, wildcard bits 0.0.255.255
Standard IP access list 50
    10 permit 10.2.0.0, wildcard bits 0.0.255.255
Standard IP access list 60
    10 permit 10.3.0.0, wildcard bits 0.0.255.255
```

Example: Verifying Route-Map Configuration

To verify route-map configuration, use the **show route-map** command. The output displays the match criteria and set action for each route-map sequence. The output also displays the number of packets and bytes that have been policy routed per each route-map sequence.

```
Device# show route-map

route-map PBR-VRF-Selection, permit, sequence 10
  Match clauses:
    ip address (access-lists): 40
  Set clauses:
    vrf VRF1
  Policy routing matches: 0 packets, 0 bytes
route-map PBR-VRF-Selection, permit, sequence 20
  Match clauses:
    ip address (access-lists): 50
  Set clauses:
    vrf VRF2
  Policy routing matches: 0 packets, 0 bytes
route-map PBR-VRF-Selection, permit, sequence 30
  Match clauses:
    ip address (access-lists): 60
  Set clauses:
    vrf VRF3
  Policy routing matches: 0 packets, 0 bytes
```

Example: Verifying Policy-Based Routing VRF Selection Policy

The following **show ip policy** command output displays the interface and associated route map that is configured for policy routing:

```
Device# show ip policy

Interface                Route map
FastEthernet0/1/0        PBR-VRF-Selection
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco Master Command List, All Releases
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

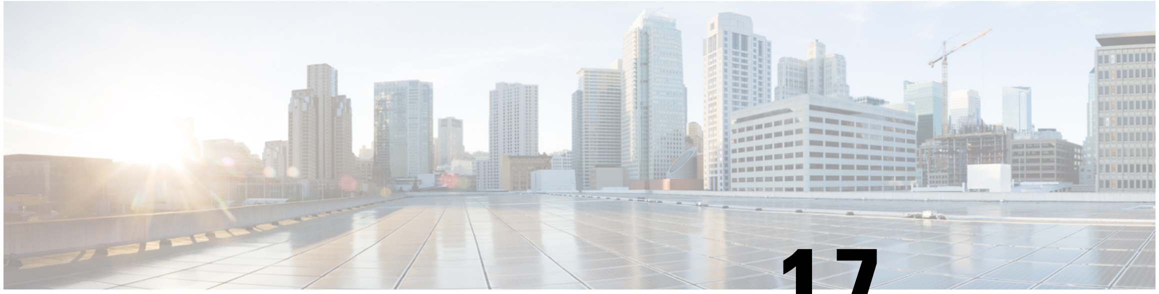
Feature Information for MPLS VPN VRF Selection Using Policy-Based Routing

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 17: Feature Information for MPLS VPN VRF Selection Using Policy-Based Routing

Feature Name	Releases	Feature Information
MPLS VPN VRF Selection Using Policy-Based Routing	12.3(7)T 12.2(25)S 12.2(33)SRB 12.2(33)SXI Cisco IOS XE Release 2.2	The MPLS VPN VRF Selection Using Policy-Based Routing feature is an extension of the MPLS VPN VRF Selection Based on Source IP Address feature. This feature introduces a policy-based routing (PBR) mechanism to classify and forward Virtual Private Network (VPN) traffic based on multiple VPN routing and forwarding (VRF) selection match criteria. In Cisco IOS Release 12.3(7)T, this feature was introduced. In Cisco IOS Releases 12.2(25)S, 12.2(33)SRB, and 12.2(33)SXI, this feature was integrated. In Cisco IOS XE Release 2.2, this feature was implemented on Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **ip vrf receive**, **set vrf**.



CHAPTER 17

MPLS VPN BGP Local Convergence

This document provides information about reducing the downtime of a provider edge (PE) to customer edge (CE) link failure. It describes how to reroute PE-egress traffic onto a backup path to the CE before the Border Gateway Protocol (BGP) has reconverged. The MPLS VPN BGP Local Convergence feature is also referred to as “local protection.” This document explains how to use PE-CE local convergence.

Note The MPLS VPN BGP Local Convergence feature affects only traffic exiting the Virtual Private Network (VPN). Therefore, it cannot fully protect traffic end-to-end by itself.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for MPLS VPN BGP Local Convergence

- Before MPLS VPN BGP Local Convergence link protection can be enabled, the customer site must be connected to the provider site by more than one path.

- Both the main forwarding path and the redundant backup path must have been installed within Border Gateway Protocol (BGP), and BGP must support lossless switchover between operational paths.
- Any of the supported routing protocols can be used between the provider edge (PE) and customer edge (CE) as long as the path is redistributed into BGP. The supported protocols for IPv4 are External BGP (eBGP), Routing Information Protocol (RIP), Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), and static routing. The supported protocols for IPv6 are eBGP and static routing.
- All PE devices that are serving as backup to the link must have assigned a unique route distinguisher to each virtual routing and forwarding (VRF) table involved with the link to ensure that the route reflectors advertise all available paths.
- Although not required the backup PE (shown as “PE2” in the figure below) should run the same Cisco software release that is running on the PE (“PE 1”) whose link with the CE is protected.

Restrictions for MPLS VPN BGP Local Convergence

- The MPLS VPN BGP Local Convergence feature affects only traffic exiting the Virtual Private Network (VPN). Therefore, it cannot fully protect traffic end-to-end by itself.
- This link protection cannot be initiated *during* a high availability (HA) stateful switchover (SSO). But links already configured with this protection *before* the switchover begins will remain protected after the switchover.
- If you perform an in-service software downgrade from an image that does include this link protection to an image that does not support this feature, active protection will be halted when Border Gateway Protocol (BGP) routes are refreshed.
- Any next-hop core tunneling technology that is supported by BGP is also supported for protection, including Multiprotocol Label Switching (MPLS), IP/Layer 2 Tunneling Protocol version 3 (L2TPv3), and IP/generic routing encapsulation (GRE). Enabling a Carrier Supporting Carrier (CSC) protocol between the provider edge (PE) and customer edge (CE) is also supported. Interautonomous system option A (back-to-back VRF) is supported because it is essentially the same as performing the PE-CE link protection in both autonomous systems. However, interautonomous system options B and C protection are not supported.
- The MPLS VPN BGP Local Convergence feature for IPv4 supports the External BGP (eBGP), Routing Information Protocol (RIP), Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), and static routing protocols only.
- The MPLS VPN BGP Local Convergence feature for IPv6 supports the eBGP and static routing protocols only.

Information About MPLS VPN BGP Local Convergence

How Link Failures Are Handled with BGP

Within a Layer 3 Virtual Private Network (VPN) network, the failure of a provider edge (PE)-customer edge (CE) link can cause a loss of connectivity (LoC) to a customer site, which is detrimental to time-sensitive applications. Several factors contribute to the duration of such an outage:

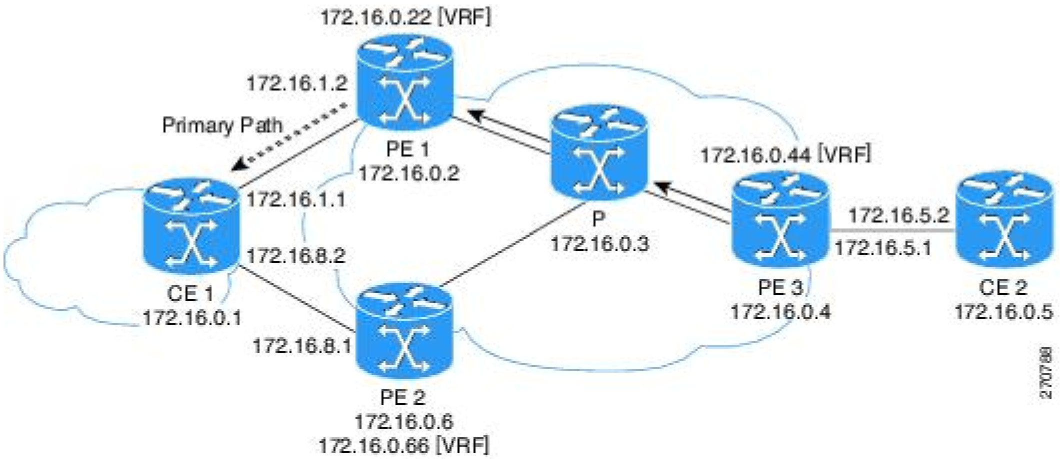
- The time to detect the failure
- The programming of the forwarding
- The convergence of the Border Gateway Protocol (BGP) (in large networks, the restored traffic arrival time at its destination varies according to the prefix)

When BGP detects a PE-CE link failure, it removes all of the BGP paths through the failing link. BGP runs the best-path algorithm on the affected prefixes and selects alternate paths for each prefix. These new paths (which typically include a remote PE) are installed into forwarding. The local labels are removed and BGP withdrawals are sent to all BGP neighbors. As each BGP neighbor receives the withdrawal messages (typically indirectly using route reflectors), the best-path algorithm is called and the prefixes are switched to an alternate path. Only then is connectivity restored.

How Links Are Handled with the MPLS VPN BGP Local Convergence Feature

The MPLS VPN BGP Local Convergence feature requires that the prefixes to be protected on a provider edge (PE)-customer edge (CE) link have at least one backup path that does not include that link. (See the figure below.) The customer site must have backup paths to the provider site.

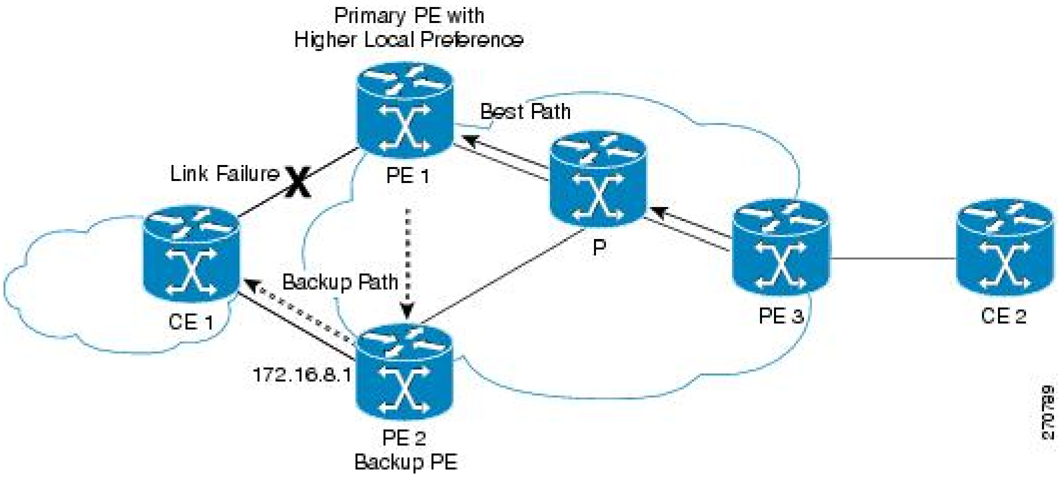
Figure 17: Network Configured with Primary and Backup Paths



The MPLS VPN BGP Local Convergence feature reduces loss of connectivity time by sending the broken link’s traffic over a backup path (as shown in the figure below) instead of waiting for total network convergence. The local label is maintained for 5 minutes while prefixes switch from the failing local path to the backup path. Because the label is not freed as had been the usual practice, forwarding continues to take place.

The best-path algorithm selects the backup path. Thus, the local label has been applied in place of the failed Border Gateway Protocol (BGP) best-path label (which is sometimes called “label swapping”). Traffic is restored locally while the network propagation of the BGP withdrawal messages takes place. Eventually, the egress PE device converges and bypasses the local repair.

Figure 18: Network Using the Backup Path After a PE-CE Link Failure on the Primary Path



Note After the 5-minute label preservation, the local labels are freed. Any BGP prefix that is remote and is not part of a Carrier Supporting Carrier (CSC) network does not have a local label and is removed. The delay in local label deletion does not modify normal BGP addition and deletion of BGP paths. Rather, BGP reprograms the new backup best path into forwarding as usual.

How Link Failures Are Detected

Local protection relies on the Border Gateway Protocol (BGP) being notified of the interface failure. Detection can occur using either the interface drivers or the routing tables. If an interface or route goes down, the corresponding path in the routing table is removed and BGP will be notified using the routing application programming interfaces (APIs).

However, when the routing table cannot detect the failure (as when a Layer 2 switch goes down), BGP determines that a neighbor is down through use of its hold-down timer. However, that determination can be extremely slow because of the 3-minute default for BGP session timeout.

You can reduce the detection delay by either reducing the BGP session timeout interval (as described in the Configuring Internal BGP Features document) or by enabling the Bidirectional Forwarding Detection (BFD) protocol within External BGP (eBGP) between the provider edge (PE) and customer edge (CE).

How to Configure MPLS VPN BGP Local Convergence

Note To configure a VPN routing and forwarding (VRF) instance for IPv4 and IPv6 VPNs or to upgrade your existing single-protocol IPv4-only VRF to a multiprotocol VRF configuration, see the "MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs" module.

Configuring MPLS VPN BGP Local Convergence with IPv4

Before you begin

Ensure that the customer edge (CE) device is already connected to the provider edge (PE) device by a minimum of two paths.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **protection local-prefixes**
6. **do show ip vrf detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf vpn1`	Enters VRF configuration mode. • If no VRF routing table and Cisco Express Forwarding table had been previously created for this named VRF, then this command also creates them, giving both tables the specified value for the *vrf-name* argument (in this example, the name is vpn1).

	Command or Action	Purpose
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:3`	(Optional) Establishes the route distinguisher for the named VRF. • If no route distinguisher had been previously established for the named VRF, then you must enter this command. • The route distinguisher value can be either an: • Autonomous system number followed by a colon and an arbitrary number (for example, 100:3) or • • IP address followed by a colon and an arbitrary number (for example, 192.168.122.15:1)
Step 5	**protection local-prefixes** **Example:** `Device(config-vrf)# protection local-prefixes`	Allows a preconfigured backup path to carry traffic if the PE-CE link breaks by preserving the local prefixes while BGP reconverges.
Step 6	**do show ip vrf detail** **Example:** `Device(config-vrf)# do show ip vrf detail`	(Optional) Verifies that the MPLS VPN BGP Local Convergence feature has been configured.

Configuring MPLS VPNBGP Local Convergence with IPv6

Before you begin

Ensure that the customer edge (CE) device is already connected to the provider edge (PE) device by a minimum of two paths.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **address-family** [**ipv4** | **ipv6**]
6. **protection local-prefixes**
7. **do show ip vrf detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vrf2`	Enters VRF configuration mode. • If no virtual routing and forwarding (VRF) routing table and Cisco Express Forwarding table had been previously created for this named VRF, then this command also creates them, giving both tables the specified value for the *vrf-name* argument (in this example, the name is vrf2).
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:3`	(Optional) Establishes the route distinguisher for the named VRF. • If no route distinguisher had been previously established for the named VRF, then you must enter this command. • The route distinguisher value can be either an: • Autonomous system number followed by a colon and an arbitrary number (for example, 100:3) or • • IP address followed by a colon and an arbitrary number (for example, 192.168.122.15:1)
Step 5	**address-family** [**ipv4** | **ipv6**] **Example:** `Device(config-vrf)# address-family ipv6`	Enters VRF address family configuration mode and specifies the IPv4 or IPv6 protocol.
Step 6	**protection local-prefixes** **Example:** `Device(config-vrf-af)# protection local-prefixes`	Allows a preconfigured backup path to carry traffic if the PE-CE link breaks by preserving the local prefixes while the Border Gateway Protocol (BGP) reconverges.
Step 7	**do show ip vrf detail** **Example:** `Device(config-vrf-af)# do show ip vrf detail`	(Optional) Verifies that the MPLS VPN BGP Local Convergence feature has been configured.

Examples

To verify that local link protection has been enabled, enter the **show ip vrf detail** command. If the protection is enabled, the status message “Local prefix protection enabled” will be shown in the display:

```
Device# show ip vrf detail

VRF vpn1 (VRF Id = 1); default RD 100:1; default VPNID <not set>
Interfaces:
    AT1/0/1.1
VRF Table ID = 1
  Export VPN route-target communities
    RT:100:1
  Import VPN route-target communities
    RT:100:1                 RT:100:2
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
     Local prefix protection enabled
```

Troubleshooting Tips

- Ensure that a minimum of two paths are present for the protected prefix in the Border Gateway Protocol (BGP) in steady state condition on the provider edge (PE) device. The path using the protected PE should be the BGP best-path before failover occurs. To display the configuration, enter the **show ip bgp vpnv4 vrf** *vpn ip-prefix* command.
- Ensure that local protection has been enabled in the protected PE by entering the **show ip vrf detail** command.
- When route reflectors exist in the topology, ensure that each virtual routing and forwarding (VRF) instance has a unique route distinguisher.

Configuration Examples for MPLS VPN BGP Local Convergence

Examples: MPLS VPN BGP Local Convergence

The following examples show how MPLS VPN BGP local convergence can prevent traffic loss after a link failure. You can display a detailed view of local link protection before, during, and after the Border Gateway Protocol (BGP) convergence by using the **show bgp vpnv4** and **show mpls forwarding-table vrf** commands as shown in the following three-stage example.

Note The **show bgp vpnv4 unicast** command is equivalent to the **show ip bgp vpnv4** command.

Example 1: Before the Link Failure

Both a primary path and a backup path have been configured:

```
Device# show bgp vpnv4 unicast all 172.16.0.1
BGP routing table entry for 100:1:172.16.0.1/32, version 2
Paths: (2 available, best #2, table v1)
Flag: 0x820
  Advertised to update-groups:
     1
  100, imported path from 100:2:172.16.0.1/32
    172.16.0.6 (metric 21) from 172.16.0.7 (172.16.0.7)
      Origin incomplete, metric 0, localpref 100, valid, internal
      Extended Community: RT:100:0
      Originator: 172.16.0.6, Cluster list: 172.16.0.7
      mpls labels in/out 16/17
  100
    172.16.1.1 from 172.16.1.1 (172.16.0.1)
      Origin incomplete, metric 0, localpref 100, valid, external, best
      Extended Community: RT:100:0
      mpls labels in/out 16/nolabel
BGP routing table entry for 100:2:172.16.0.1/32, version 9
Paths: (1 available, best #1, no table)
Flag: 0x820
  Not advertised to any peer
  100
    172.16.0.6 (metric 21) from 172.16.0.7 (172.16.0.7)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:0
      Originator: 172.16.0.6, Cluster list: 172.16.0.7
      mpls labels in/out nolabel/17
```

Label information for both paths can be displayed:

```
Device# show bgp vpnv4 unicast all labels
Network          Next Hop      In label/Out label
Route Distinguisher: 100:1 (v1)
   172.16.0.1/32    172.16.0.6      16/17
                    172.16.1.1      16/nolabel
   172.16.0.5/32    172.16.0.4      nolabel/23
   172.16.0.22/32   0.0.0.0         17/nolabel(v1)
   172.16.0.44/32   172.16.0.4      nolabel/24
   172.16.0.66/32   172.16.0.6      nolabel/21
   172.16.1.0/24    172.16.1.1      18/nolabel
                    0.0.0.0         18/nolabel(v1)
   172.16.5.0/24    172.16.0.4      nolabel/25
   172.16.8.0/24    172.16.0.6      19/23
                    172.16.1.1      19/nolabel
Route Distinguisher: 100:2
   172.16.0.1/32    172.16.0.6      nolabel/17
   172.16.0.66/32   172.16.0.6      nolabel/21
   172.16.8.0/24    172.16.0.6      nolabel/23
```

The PE1 (see the first figure above) forwarding table contains BGP best-path information:

```
Device# show mpls forwarding-table vrf v1 172.16.0.1 detail
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
16         No Label   172.16.0.1/32[V] 570           Et0/0      172.16.1.1
        MAC/Encaps=14/14, MRU=1504, Label Stack{}
        AABBCC000B00AABBCC000C000800
        VPN route: v1
        No output feature configured
```

Example 2: After the Link Failure and Before BGP Convergence

After the link failure on only one path, the backup path remains available (see the second figure above):

```
Device# show bgp vpnv4 unicast all 172.16.0.1

BGP routing table entry for 100:1:172.16.0.1/32, version 19
Paths: (1 available, best #1, table v1)
  Not advertised to any peer
  100, imported path from 100:2:172.16.0.1/32
    172.16.0.6 (metric 21) from 172.16.0.7 (172.16.0.7)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:0
      Originator: 172.16.0.6, Cluster list: 172.16.0.7
      mpls labels in/out 16/17
BGP routing table entry for 100:2:172.16.0.1/32, version 9
Paths: (1 available, best #1, no table)
  Not advertised to any peer
  100
    172.16.0.6 (metric 21) from 172.16.0.7 (172.16.0.7)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:0
      Originator: 172.16.0.6, Cluster list: 172.16.0.7
      mpls labels in/out nolabel/17
```

The label information for the backup path label can be displayed:

```
Device# show bgp vpnv4 unicast all labels

Network          Next Hop      In label/Out label
Route Distinguisher: 100:1 (v1)
   172.16.0.1/32    172.16.0.6      16/17
   172.16.0.5/32    172.16.0.4      nolabel/23
   172.16.0.22/32   0.0.0.0         17/nolabel(v1)
   172.16.0.44/32   172.16.0.4      nolabel/24
   172.16.0.66/32   172.16.0.6      nolabel/21
   172.16.1.0/24    172.16.0.6      nolabel/22
   172.16.5.0/24    172.16.0.4      nolabel/25
   172.16.8.0/24    172.16.0.6      19/23
Route Distinguisher: 100:2
   172.16.0.1/32    172.16.0.6      nolabel/17
   172.16.0.66/32   172.16.0.6      nolabel/21
   172.16.1.0/24    172.16.0.6      nolabel/22
   172.16.8.0/24    172.16.0.6      nolabel/23
```

The PE 1 (see the first figure above) forwarding table contains new label and next-hop information to direct traffic onto the backup path:

```
Device# show mpls forwarding-table vrf v1 172.16.0.1 detail

Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
16         17         172.16.0.1/32[V] 0             Et1/0      172.16.3.2
        MAC/Encaps=14/22, MRU=1496, Label Stack{21 17}
        AABBCC000D00AABBCC000C018847 0001500000011000
        VPN route: v1
        No output feature configured
```

Example 3: After Local Label Expiration and BGP Reconvergence

Because the local label preservation window has expired, the replacement local label is now gone from the PE 1 forwarding table information:

```
Device# show mpls forwarding-table vrf v1 172.16.0.1 detail
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
None       17         172.16.0.1/32[V] 0             Et1/0      172.16.3.2
        MAC/Encaps=14/22, MRU=1496, Label Stack{21 17}
        AABBCC000D00AABBCC000C018847 0001500000011000
        VPN route: v1
        No output feature configured
```

The new BGP information reverts to the configuration shown in the first figure above:

```
Device# show bgp vpnv4 unicast all 172.16.0.1
BGP routing table entry for 100:1:172.16.0.1/32, version 23
Paths: (1 available, best #1, table v1)
  Not advertised to any peer
  100, imported path from 100:2:172.16.0.1/32
    172.16.0.6 (metric 21) from 172.16.0.7 (172.16.0.7)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:0
      Originator: 172.16.0.6, Cluster list: 172.16.0.7
      mpls labels in/out nolabel/17
BGP routing table entry for 100:2:172.16.0.1/32, version 9
Paths: (1 available, best #1, no table)
  Not advertised to any peer
  100
    172.16.0.6 (metric 21) from 172.16.0.7 (172.16.0.7)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:0
      Originator: 172.16.0.6, Cluster list: 172.16.0.7
      mpls labels in/out nolabel/17
Device# show bgp vpnv4 unicast all labels

Network          Next Hop      In label/Out label
Route Distinguisher: 100:1 (v1)
   172.16.0.1/32    172.16.0.6      nolabel/17
   172.16.0.5/32    172.16.0.4      nolabel/23
   172.16.0.22/32   0.0.0.0         17/nolabel(v1)
   172.16.0.44/32   172.16.0.4      nolabel/24
   172.16.0.66/32   172.16.0.6      nolabel/21
   172.16.1.0/24    172.16.0.6      nolabel/22
   172.16.5.0/24    172.16.0.4      nolabel/25
   172.16.8.0/24    172.16.0.6      nolabel/23
Route Distinguisher: 100:2
   172.16.0.1/32    172.16.0.6      nolabel/17
   172.16.0.66/32   172.16.0.6      nolabel/21
   172.16.1.0/24    172.16.0.6      nolabel/22
   172.16.8.0/24    172.16.0.6      nolabel/23
```

Examples: MPLS VPN BGP Local Convergence for 6VPE 6PE

You can display a detailed view of local link protection before, during, and after the Border Gateway Protocol (BGP) local convergence for Cisco VPN IPv6 provider edge devices (6VPE) and Cisco IPv6 provider edge devices (6PE) over Multiprotocol Label Switching (MPLS) by using the **show bgp vpnv6** and **show mpls forwarding-table vrf** commands as shown in the following three-stage example.

The figure below shows an MPLS VPN with BGP local convergence configured. The PE-to-CE routing protocol is External BGP (eBGP), and the PE to route reflector (RR) sessions are BGP VPNv6. The protected prefix is the CE 1 loopback (2001:0DB8::/128). The primary path is from PE 1 to CE 1. The secondary path is from PE 1, through P and PE3, to CE 1.

Figure 19: MPLS VPN BGP Local Convergence

Example 1: Before the Link Failure

Both a primary path and a backup path have been configured for the prefix 2001:0DB8::/128. The inlabel/outlabel settings for the two paths are 28/28 and 28/nolabel.

```
Device# show bgp vpnv6 unicast all 2001:0DB8::/128
BGP routing table entry for [1:1]2001:0DB8::/128, version 5
Paths: (2 available, best #2, table v1)
 Advertised to update-groups:
     2
  100, imported path from [2:2]2001:0DB8::/128
    ::FFFF:10.6.6.6 (metric 21) from 10.7.7.7 (10.7.7.7)
      Origin incomplete, metric 0, localpref 100, valid, internal
      Extended Community: RT:1:1
      Originator: 10.6.6.6, Cluster list: 10.7.7.7
      mpls labels in/out 28/28
  100
    2001:0DB8:0:ABCD::1 (FE80::A8BB:CCFF:FE00:B00) from 2001:0DB8:0:ABCD::1 (10.1.1.1)
      Origin incomplete, metric 0, localpref 100, valid, external, best

      Extended Community: RT:1:1
      mpls labels in/out 28/nolabel
BGP routing table entry for [2:2]2001:0DB8::/128, version 11
Paths: (1 available, best #1, no table)
  Not advertised to any peer
  100
    ::FFFF:10.6.6.6 (metric 21) from 10.7.7.7 (10.7.7.7)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:1:1
      Originator: 10.6.6.6, Cluster list: 10.7.7.7
      mpls labels in/out nolabel/28
```

The PE 1 forwarding table contains new label and next-hop information to direct traffic onto the backup path:

```
Device#
show mpls forwarding-table vrf v1 2001:0DB8::/128 detail
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
28        No Label   2001:0DB8::/128[V]    804          Et0/0     FE80::A8BB:CCFF:FE00:B00

        MAC/Encaps=14/14, MRU=1504, Label Stack{}
        AABBCC000B00AABBCC000C0086DD
        VPN route: v1
        No output feature configured
```

Example 2: After the Link Failure

After the link failure, the backup path is still available, the original path is removed from BGP, and the backup path is activated:

```
Device# show mpls forwarding-table vrf v1 2001:0DB8::/128 detail
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
28         28         2001:0DB8::/128[V]     0             Et1/0      10.3.0.2
        MAC/Encaps=14/22, MRU=1496, Label Stack{23 28}
        AABBCC000D00AABBCC000C018847 000170000001C000
        VPN route: v1
        No output feature configured
```

After a configured length of time, the local label expires. The output from the **show mpls forwarding-table**command also verifies that the local label has expired:

```
Device# show mpls forwarding-table vrf v1 2001:0DB8::/128 detail
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
None       28         2001:0DB8::/128[V]      0             Et1/0      10.3.0.2
        MAC/Encaps=14/22, MRU=1496, Label Stack{23 28}
        AABBCC000D00AABBCC000C018847 000170000001C000
        VPN route: v1
        No output feature configured
Example 3: After the Link Is Restored
```

When the link is restored the original path is added to BGP and the traffic switches back to this path:

```
Device# show bgp vpnv6 unicast all 2001:0DB8::/128
BGP routing table entry for [1:1]2001:0DB8::/128, version 28
Paths: (2 available, best #1, table v1)
  Advertised to update-groups:
     2
  100
    2001:0DB8:0:ABCD::1 (FE80::A8BB:CCFF:FE00:B00) from 2001:0DB8:0:ABCD::1 (10.1.1.1)
      Origin incomplete, metric 0, localpref 100, valid, external, best
      Extended Community: RT:1:1
      mpls labels in/out 16/nolabel
  100, imported path from [2:2]2001:0DB8::/128
    ::FFFF:10.6.6.6 (metric 21) from 10.7.7.7 (10.7.7.7)
      Origin incomplete, metric 0, localpref 100, valid, internal
      Extended Community: RT:1:1
      Originator: 10.6.6.6, Cluster list: 10.7.7.7
      mpls labels in/out 16/28
BGP routing table entry for [2:2]2001:0DB8::/128, version 11
Paths: (1 available, best #1, no table)
  Not advertised to any peer
  100
    ::FFFF:10.6.6.6 (metric 21) from 10.7.7.7 (10.7.7.7)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:1:1
      Originator: 10.6.6.6, Cluster list: 10.7.7.7
      mpls labels in/out nolabel/28
Device# show mpls for vrf v1 2001:0DB8::/128 detail
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
16         No Label   2001:0DB8::/128[V] 0          Et0/0      FE80::A8BB:CCFF:FE00:B00
        MAC/Encaps=14/14, MRU=1504, Label Stack{}
        AABBCC000B00AABBCC000C0086DD
        VPN route: v1
        No output feature configured
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco Master Command List, All Releases
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference

Related Topic	Document Title
BGP configuration	“Configuring a Basic BGP Network” module in the *IP Routing: BGP Configuration Guide*
Protocol for quickly detecting failed forwarding paths	“Bidirectional Forwarding Detection” module in the *IP Routing: BFD Configuration Guide*
Configuration of BGP PIC Edge for IP and MPLS-VPN	“BGP PIC Edge for IP and MPLS VPN” module in the *MPLS Layer 3 VPNs Configuration Guide*
Configuration of internal BGP features	“Configuring Internal BGP Features” module in the *IP Routing: BGP Configuration Guide*
Configuration of VRF under the specific cases of IPv4 and IPv6 situations	“MPLS VPN VRF CLI for IPv4 and IPv6 VPNs” module in the *MPLS Layer 3 VPNs Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 2547	BGP/MPLS VPNs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS VPN BGP Local Convergence

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 18: Feature Information for MPLS VPN BGP Local Convergence

Feature Name	Releases	Feature Information
MPLS VPN BGP Local Convergence	12.2(33)SRC 12.2(33)SB 15.0(1)M Cisco IOS XE Release 3.1S	The MPLS VPN BGP Local Convergence feature reduces the downtime of a PE-CE link failure by rerouting PE-egress traffic onto a backup path to the CE before BGP has reconverged. In 12.2(33)SRC, this feature was introduced on the Cisco 7200 and the Cisco 7600. In 12.2(33)SB, this feature became available on the Cisco 7300 series and the Cisco 10000 Series Routers. This feature was integrated into Cisco IOS Release 15.0(1)M. In Cisco IOS XE Release 3.1S, this feature was implemented on Cisco ASR 1000 Series Routers. The following command was modified: **protection local-prefixes**.
MPLS VPN BGP Local Convergence for 6VPE/6PE	15.0(1)S Cisco IOS XE Release 3.1S	The MPLS VPN BGP Local Convergence for 6VPE/6PE feature implements MPLS VPN BGP local convergence for Cisco VPN IPv6 provider edge devices (6VPE) and Cisco IPv6 provider edge devices (6PE) over MPLS. In 15.0(1)S, this feature was introduced. In Cisco IOS XE Release, 3.1S, this feature was implemented on Cisco ASR 1000 Series Routers. The following command was modified: **protection local-prefixes**.

CHAPTER 18

Multi-VRF Support

The Multi-VRF Support feature allows you to configure and maintain more than one instance of a routing and forwarding table within the same customer edge (CE) device.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for Multi-VRF Support

The network’s core and provider edge (PE) devices must be configured for Virtual Private Network (VPN) operation.

Restrictions for Multi-VRF Support

- You can configure the Multi-VRF Support feature only on Layer 3 interfaces.
- The Multi-VRF Support feature is not supported by Interior Gateway Routing Protocol (IGRP) nor Intermediate System to Intermediate System (IS-IS).

- Label distribution for a given VPN routing and forwarding (VRF) instance on a given device can be handled by either Border Gateway Protocol (BGP) or Label Distribution Protocol (LDP), but not by both protocols at the same time.
- Multicast cannot operate on a Layer 3 interface that is configured with the Multi-VRF Support feature.

Information About Multi-VRF Support

How the Multi-VRF Support Feature Works

The Multi-VRF Support feature enables a service provider to support two or more Virtual Private Networks (VPNs), where the IP addresses can overlap several VPNs. The Multi-VRF Support feature uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each virtual routing and forwarding (VRF) instance. Interfaces in a VRF can be either physical, such as FastEthernet ports, or logical, such as VLAN , but a Layer 3 interface cannot belong to more than one VRF at any one time. The Multi-VRF Support feature allows an operator to support two or more routing domains on a customer edge (CE) device, with each routing domain having its own set of interfaces and its own set of routing and forwarding tables. The Multi-VRF Support feature makes it possible to extend the label switched paths (LSPs) to the CE and into each routing domain that the CE supports.

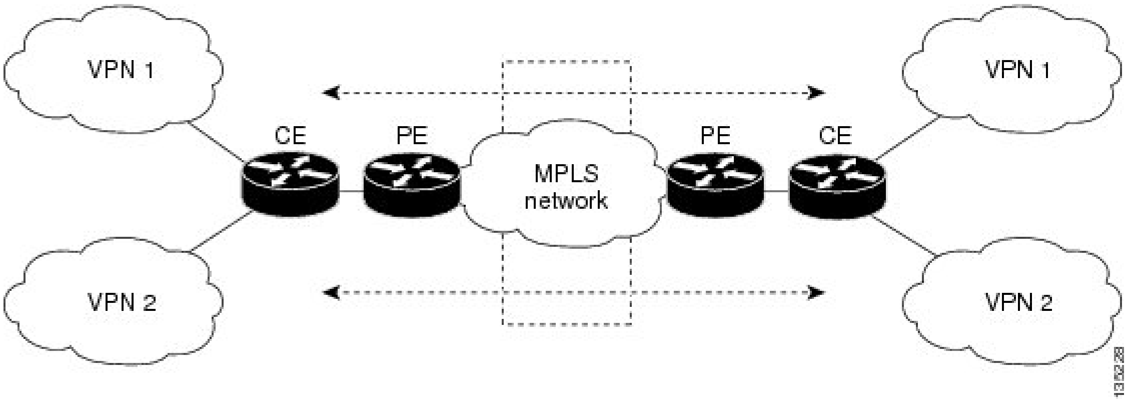
The Multi-VRF Support feature works as follows:

- Each CE device advertises its site's local routes to a provider edge (PE) device and learns the remote VPN routes from that provider edge (PE) device.
- PE devices exchange routing information with CE devices by using static routing or a routing protocol such as the Border Gateway Protocol (BGP), Routing Information Protocol version 1 (RIPv1), or RIPv2.
- PE devices exchange MPLS label information with CE devices through Label Distribution Protocol (LDP) or BGP.
- The PE device needs to maintain VPN routes only for those VPNs to which it is directly attached, eliminating the requirement that the PE maintain all of the service provider's VPN routes. Each PE device maintains a VRF for each of its directly connected sites. Two or more interfaces on a PE device can be associated with a single VRF if all the sites participate in the same VPN. Each VPN is mapped to a specified VRF. After learning local VPN routes from CE devices, the PE device exchanges VPN routing information with other PE devices through internal BGP (iBGP).

With the Multi-VRF Support feature, two or more customers can share one CE device, and only one physical link is used between the CE and the PE devices. The shared CE device maintains separate VRF tables for each customer and routes packets for each customer based on that customer's own routing table. The Multi-VRF Support feature extends limited PE device functionality to a CE device, giving it the ability, through the maintenance of separate VRF tables, to extend the privacy and security of a VPN to the branch office.

The figure below shows a configuration where each CE device acts as if it were two CE devices. Because the Multi-VRF Support feature is a Layer 3 feature, each interface associated with a VRF must be a Layer 3 interface.

Figure 20: Each CE Device Acting as Several Virtual CE Devices



How Packets Are Forwarded in a Network Using the Multi-VRF Support Feature

Following is the packet-forwarding process in an Multi-VRF customer edge (CE)-enabled network, as illustrated in the figure above:

- When the CE receives a packet from a Virtual Private Network (VPN), it looks up the routing table based on the input interface. When a route is found, the CE imposes the Multiprotocol Label Switching (MPLS) label that it received from the provider edge (PE) for that route and forwards the packet to the PE.
- When the ingress PE receives a packet from the CE, it swaps the incoming label with the corresponding label stack and sends the packet to the MPLS network.
- When an egress PE receives a packet from the network, it swaps the VPN label with the label that it had earlier received for the route from the CE, and it forwards the packet to the CE.
- When a CE receives a packet from an egress PE, it uses the incoming label on the packet to forward the packet to the correct VPN.

To configure Multi-VRF, you create a VRF table and then specify the Layer 3 interface associated with that VRF. Next, you configure the routing protocols within the VPN, and between the CE and the PE. The Border Gateway Protocol (BGP) is the preferred routing protocol for distributing VPN routing information across the provider’s backbone.

The Multi-VRF network has three major components:

- VPN route target communities: These are lists of all other members of a VPN community. You must configure VPN route targets for each VPN community member.
- Multiprotocol BGP peering of VPN community PE devices: This propagates VRF reachability information to all members of a VPN community. You must configure BGP peering in all PE devices within a VPN community.
- VPN forwarding: This transports all traffic between VPN community members across a VPN service-provider network.

Considerations When Configuring the Multi-VRF Support Feature

- A device with the Multi-VRF Support feature is shared by several customers, and each customer has its own routing table.
- Because each customer uses a different virtual routing and forwarding (VRF) table, the same IP addresses can be reused. Overlapping IP addresses are allowed in different Virtual Private Networks (VPNs).
- The Multi-VRF Support feature lets several customers share the same physical link between the provider edge (PE) and the customer edge (CE) devices. Trunk ports with several VLANs separate packets among the customers. Each customer has its own VLAN.
- For the PE device, there is no difference between using the Multi-VRF Support feature or using several CE devices.
- The Multi-VRF Support feature does not affect the packet-switching rate.

How to Configure Multi-VRF Support

Configuring VRFs

To configure virtual routing and forwarding (VRF) instances, complete the following procedure. Be sure to configure VRFs on both the provider edge (PE) and customer edge (CE) devices.

If a VRF has not been configured, the device has the following default configuration:

- No VRFs have been defined.
- No import maps, export maps, or route maps have been defined.
- No VRF maximum routes exist.
- Only the global routing table exists on the interface.

The following are the supported flavors of multicast over VRF on Cisco ASR 920 RSP2 module:

- Multicast with multi-VRF (MPLS VPN/MLDP)
- Multicast with GRE tunnel (MVPN GRE)
- Multicast with VRF-lite

Note Multi-VRF/MVPN GRE configured layer-3 interface cannot participate in more than one VRF at the same time.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip routing**

4. **ip vrf** *vrf-name*
5. **rd** *route-distinguisher*
6. **route-target** {**export** | **import** | **both**} *route-target-ext-community*
7. **import map** *route-map*
8. **exit**
9. **interface** *type slot*/*subslot*/*port*[**.***subinterface*]
10. **ip vrf forwarding** *vrf-name*
11. **end**
12. **show ip vrf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip routing** **Example:** `Device(config)# ip routing`	Enables IP routing.
Step 4	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf v1`	Names the VRF, and enters VRF configuration mode.
Step 5	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:1`	Creates a VRF table by specifying a route distinguisher. Enter either an autonomous system number and an arbitrary number (xxx:y), or an IP address and an arbitrary number (A.B.C.D:y).
Step 6	**route-target** {**export** | **import** | **both**} *route-target-ext-community* **Example:** `Device(config-vrf)# route-target export 100:1`	Creates a list of import, export, or import and export route target communities for the specified VRF. Enter either an autonomous system number and an arbitrary number (xxx:y), or an IP address and an arbitrary number (A.B.C.D:y). **Note** This command works only if BGP is running.

	Command or Action	Purpose

Step 7	**import map** *route-map*
	Example:
	`Device(config-vrf)# import map importmap1`
		(Optional) Associates a route map with the VRF.

Step 8	**exit** **Example:** `Device(config-vrf)# exit`	Returns to global configuration mode.
Step 9	**interface** *type slot***/***subslot***/***port*[**.***subinterface*] **Example:** `Device(config)# interface`	Specifies the Layer 3 interface to be associated with the VRF and enters interface configuration mode. The interface can be a routed port or an .
Step 10	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding v1`	Associates the VRF with the Layer 3 interface.

Step 11	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Step 12	**show ip vrf** **Example:** `Device# show ip vrf`	Displays the settings of the VRFs.

Configuring BGP as the Routing Protocol

Most routing protocols can be used between the customer edge (CE) and the provider edge (PE) devices. However, external BGP (eBGP) is recommended, because:

- BGP does not require more than one algorithm to communicate with many CE devices.
- BGP is designed to pass routing information between systems run by different administrations.
- BGP makes it easy to pass route attributes to the CE device.

When BGP is used as the routing protocol, it can also be used to handle the Multiprotocol Label Switching (MPLS) label exchange between the PE and CE devices. By contrast, if Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP), Routing Information Protocol (RIP), or static routing is used, the Label Distribution Protocol (LDP) must be used to signal labels.

To configure a BGP PE-to-CE routing session, perform the following steps on the CE and on the PE devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **network** *ip-address* **mask** *network-mask*
5. **redistribute ospf** *process-id* **match internal**
6. **network** *ip-address wildcard-mask* **area** *area-id*
7. **address-family ipv4 vrf** *vrf-name*
8. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
9. **neighbor** *address* **activate**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the BGP routing process with the autonomous system number passed to other BGP devices, and enters router configuration mode.
Step 4	**network** *ip-address* **mask** *network-mask* **Example:** `Device(config-router)# network 10.0.0.0 mask 255.255.255.0`	Specifies a network and mask to announce using BGP.
Step 5	**redistribute ospf** *process-id* **match internal** **Example:** `Device(config-router)# redistribute ospf 2 match internal`	Sets the device to redistribute OSPF internal routes.
Step 6	**network** *ip-address wildcard-mask* **area** *area-id* **Example:** `Device(config-router)# network 10.0.0.0 255.255.255.0 area 0`	Identifies the network address and mask on which OSPF is running, and the area ID of that network address.

	Command or Action	Purpose
Step 7	**address-family ipv4 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf v12`	Identifies the name of the virtual routing and forwarding (VRF) instance that will be associated with the next two commands, and enters VRF address-family mode.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router-af)# neighbor 10.0.0.3 remote-as 100`	Informs this device's BGP neighbor table of the neighbor's address (or peer group name) and the neighbor's autonomous system number.
Step 9	**neighbor** *address* **activate** **Example:** `Device(config-router-af)# neighbor 10.0.0.3 activate`	Activates the advertisement of the IPv4 address-family neighbors.

Configuring PE-to-CE MPLS Forwarding and Signaling with BGP

If the Border Gateway Protocol (BGP) is used for routing between the provider edge (PE) and the customer edge (CE) devices, configure BGP to signal the labels on the virtual routing and forwarding (VRF) interfaces of both the CE and the PE devices. You must enable signalling globally at the router-configuration level and for each interface:

- At the router-configuration level, to enable Multiprotocol Label Switching (MPLS) label signalling via BGP, use the **neighbor send-label** command).
- At the interface level, to enable MPLS forwarding on the interface used for the PE-to-CE external BGP (eBGP) session, use the **mpls bgp forwarding** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4 vrf** *vrf-name*
5. **neighbor** *address* **send-label**
6. **neighbor** *address* **activate**
7. **end**
8. **configure terminal**
9. **interface** *type slot/subslot/port*[**.***subinterface*]
10. **mpls bgp forwarding**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the BGP routing process with the autonomous system number passed to other BGP devices and enters router configuration mode.
Step 4	**address-family ipv4 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf v12`	Identifies the name of the VRF instance that will be associated with the next two commands and enters address family configuration mode.
Step 5	**neighbor** *address* **send-label** **Example:** `Device(config-router-af)# neighbor 10.0.0.3 send-label`	Enables the device to use BGP to distribute MPLS labels along with the IPv4 routes to the peer devices. If a BGP session is running when you issue this command, the command does not take effect until the BGP session is restarted.
Step 6	**neighbor** *address* **activate** **Example:** `Device(config-router-af)# neighbor 10.0.0.3 activate`	Activates the advertisement of the IPv4 address-family neighbors.

Step 7	**end**
	Example:
	`Device(config-router-af)# end`
		Returns to privileged EXEC mode.

Step 8	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 9	**interface** *type slot***/***subslot***/***port*[**.***subinterface*] **Example:**	Enters interface configuration mode for the interface to be used for the BGP session. The interface can be a routed port or an .

	Command or Action	Purpose
	`Device(config)# interface`	
Step 10	**mpls bgp forwarding** **Example:** `Device(config-if)# mpls bgp forwarding`	Enables MPLS forwarding on the interface.

Configuring a Routing Protocol Other than BGP

You can use the Routing Information Protocol (RIP), Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), or static routing. This configuration uses OSPF, but the process is the same for other protocols.

If you use OSPF as the routing protocol between the provider edge (PE) and the customer edge (CE) devices, issue the **capability vrf-lite** command in router configuration mode.

Note If RIP EIGRP, OSPF or static routing is used, the Label Distribution Protocol (LDP) must be used to signal labels.

The Multi-VRF Support feature is not supported by Interior Gateway Routing Protocol (IGRP) or Intermediate System-to-Intermediate System (IS-IS).

Multicast cannot be configured on the same Layer 3 interface as the Multi-VRF Support feature is configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id* [**vrf** *vpn-name*]
4. **log-adjacency-changes**
5. **redistribute bgp** *autonomous-system-number* **subnets**
6. **network** *ip-address subnet-mask* **area** *area-id*
7. **end**
8. **show ip ospf**

DETAILED STEPS

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

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**router ospf** *process-id* [**vrf** *vpn-name*] **Example:** `Device(config)# router ospf 100 vrf v1`	Enables OSPF routing, specifies a virtual routing and forwarding (VRF) table, and enters router configuration mode.
Step 4	**log-adjacency-changes** **Example:** `Device(config-router)# log-adjacency-changes`	(Optional) Logs changes in the adjacency state. This is the default state.
Step 5	**redistribute bgp** *autonomous-system-number* **subnets** **Example:** `Device(config-router)# redistribute bgp 800 subnets`	Sets the device to redistribute information from the Border Gateway Protocol (BGP) network to the OSPF network.
Step 6	**network** *ip-address subnet-mask* **area** *area-id* **Example:** `Device(config-router)# network 10.0.0.0 255.255.255.0 area 0`	Indicates the network address and mask on which OSPF runs, and the area ID of that network address.

Step 7	**end**
	Example:
	`Device(config-router)# end`
		Returns to privileged EXEC mode.

Step 8	**show ip ospf** **Example:** `Device# show ip ospf`	Displays information about the OSPF routing processes.

Configuring PE-to-CE MPLS Forwarding and Signaling with LDP

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* /*subslot*/*port*[**.***subinterface*]
4. **mpls ip**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* /*subslot*/*port*[.*subinterface*] **Example:** `Device(config)# interface`	Enters interface configuration mode for the interface associated with the VRF. The interface can be a routed port or an .
Step 4	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables MPLS forwarding of IPv4 packets along normally routed paths for this interface.

Configuration Examples for Multi-VRF Support

The figure below is an example of a Multi-VRF topology.

Example: Configuring Multi-VRF Support on the PE Device

The following example shows how to configure a VRF:

```
configure terminal
ip vrf v1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
 exit
ip vrf v2
 rd 100:2
 route-target export 100:2
 route-target import 100:2
 exit
```

The following example shows how to configure on PE device, PE-to-CE connections using BGP for both routing and label exchange:

The following example shows how to configure on PE device, PE-to-CE connections using OSPF for routing and LDP for label exchange:

Example: Configuring Multi-VRF Support on the CE Device

The following example shows how to configure VRFs:

```
configure terminal
 ip routing
 ip vrf v11
  rd 800:1
  route-target export 800:1
  route-target import 800:1
  exit
 ip vrf v12
  rd 800:2
  route-target export 800:2
  route-target import 800:2
  exit
```

The following example shows how to configure CE device VPN connections:

```
interface
 ip vrf forwarding v11
 ip address 10.0.0.8 255.255.255.0
 exit
interface
 ip vrf forwarding v12
 ip address 10.0.0.8 255.255.255.0
 exit
router ospf 1 vrf v11
 network 10.0.0.0 255.255.255.0 area 0
 network 10.0.0.0 255.255.255.0 area 0
 exit
router ospf 2 vrf v12
 network 10.0.0.0 255.255.255.0 area 0
 network 10.0.0.0 255.255.255.0 area 0
 exit
```

Note If BGP is used for routing between the PE and CE devices, the BGP-learned routes from the PE device can be redistributed into OSPF using the commands in the following example.

```
router ospf 1 vrf v11
 redistribute bgp 800 subnets
 exit
router ospf 2 vrf v12
 redistribute bgp 800 subnets
 exit
```

The following example shows how to configure on CE devices, PE-to-CE connections using BGP for both routing and label exchange:

The following example shows how to configure on CE devices, PE-to-CE connections using OSPF for both routing and LDP for label exchange:

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco Master Command List, All Releases
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference
OSPF with Multi-VRF	"OSPF Support for Multi-VRF in CE Routers" module in the OSPF Configuration Guide.

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Multi-VRF Support

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

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

Table 19: Feature Information for Multi-VRF Support

Feature Name	Releases	Feature Information
Multi-VRF Support	12.1(11)EA1 12.1(20)EW 12.2(4)T 12.2(8)YN 12.2(25)EWA	The Multi-VRF Support feature allows you to configure and maintain more than one instance of a routing and forwarding table within the same CE device.

CHAPTER 19

MPLS VPN per Customer Edge (CE) Label

The MPLS VPN per Customer Edge (CE) Label feature allows you to configure a single VPN label at the provider edge (PE) for every immediate next hop or set of next hops.

You can enable (or disable) the MPLS VPN per CE Label feature in global configuration mode.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for MPLS VPN per CE Label

- If your virtual routing and forwarding (VRF) domain has any of the following features enabled, disable them before you configure the MPLS VPN per CE Label feature:
    - External Border Gateway Protocol (EBGP) multipath feature
    - Internal Border Gateway Protocol (IBGP) multipath feature
    - Carrier Supporting Carrier (CSC) feature
- Before configuring Multiprotocol Label Switching (MPLS) Layer 3 VPNs, you must install MPLS, Label Distribution Protocol (LDP), and Cisco Express Forwarding in your network. All devices in the core,

including the provider edge (PE) devices, must be able to support Cisco Express Forwarding and MPLS forwarding.

Restrictions for MPLS VPN per CE Label

- Enabling the MPLS VPN per CE Label feature causes Border Gateway Protocol (BGP) reconvergence, which can result in data loss for traffic coming from the Multiprotocol Label Switching (MPLS) VPN core.

  Note You can minimize network disruption by enabling this feature during a scheduled MPLS maintenance window. Also, if possible, avoid enabling this feature on a live device.

- IPv6 Provider Edge devices (6PE) are not supported.
- Prefix-Independent Convergence (PIC) is not supported. Per CE Label with only multipath is supported. You cannot use this feature with:
  - Internal Border Gateway Protocol (IBGP) multipath feature
  - Carrier Supporting Carrier (CSC) feature
- When per CE label is configured, MPLS Forwarding Infrastructure (MFI) has to back up key and label information to a standby device. This will impact software downgrades.
- The BGP Best External feature provides the network with a backup external route to avoid loss of connectivity of the primary external route. This feature is not supported.
- Importing routes from protocols other than BGP on a PE device is not supported.
- Any network with a zero next hop is assigned one label per network, because the next hop cannot be reliably determined.
- Do not use per CE labels if there are multiple neighbors with the same address in a VRF domain.
- Only single hop EBGP is supported. Multihop EBGP is not supported.
- In high availability configurations, labels will be preserved after switchover from standby only if BGP Graceful Restart is configured before establishing BGP sessions.

Information About MPLS VPN per CE Label

MPLS VPN per CE Label Functionality

The provider edge (PE) devices store both local and remote routes and include a label entry for each route. For distributed platforms, the per-prefix labels consume memory. When there are many virtual routing and forwarding (VRF) domains and routes, the amount of memory that the per-prefix labels consume can become an issue. The purpose of using per CE label allocation is to avoid an additional lookup on the PE device's routing table and to conserve label space.

The MPLS VPN per CE Label feature allows the same label to be used for all the routes advertised from a unique customer edge (CE) peer device. The PE device allocates one label for every immediate next hop (in most cases, the next hop is a CE router). The label is directly mapped to the next hop so there is no VRF route lookup performed during data forwarding. However, the number of labels allocated is one for each CE rather than one for each prefix. As BGP is aware of all the next hops, it assigns a label for each next hop (not for each PE-CE interface).

How to Configure MPLS VPN per CE Label

Configuring the per CE Label Feature

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label mode** {**vrf** *vrf-name* | **all-vrfs**} **protocol** {**bgp-vpnv4** | **bgp-vpnv6** | **all-afs**} {**per-ce**}
4. **end**
5. **show ip vrf detail**
6. **show mpls forwarding-table**

DETAILED STEPS

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

Step 3	**mpls label mode** {**vrf** *vrf-name* | **all-vrfs**} **protocol** {**bgp-vpnv4** | **bgp-vpnv6** | **all-afs**} {**per-ce**}
	Example:
	`Device(config)# mpls label mode all-vrfs protocol bgp-vpnv6 per-ce`
		Configures the MPLS VPN per CE Label feature.

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 5	**show ip vrf detail** **Example:** `Device# show ip vrf detail`	Displays the VRF label mode.
Step 6	**show mpls forwarding-table** **Example:** `Device# show mpls forwarding-table`	Displays the contents of the Multiprotocol Label Switching (MPLS) Label Forwarding Information Base (LFIB).

Configuration Examples for MPLS VPN per CE Label

Examples: MPLS VPN per CE Label

```
Device> enable
Device# configure terminal
Device(config)# mpls label mode all-vrfs protocol bgp-vpnv6 per-ce
Device(config)# end
```

You can use **show** commands to view information about a per CE label configuration.

The following example shows how to display detailed information about the defined VPN routing and forwarding (VRF) instances and associated interfaces:

```
PE1# show ip vrf detail

VRF red (VRF Id = 1); default RD 1:1; default VPNID <not set="">
  New CLI format, supports multiple address-families
  Flags: 0x180C
  Interfaces:
    Et1/0                    Et2/0
VRF Table ID = 1
  Flags: 0x0
  Export VPN route-target communities
    RT:1:1
  Import VPN route-target communities
    RT:1:1
  No import route-map
  No global export route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-ce
```

The following example shows how to display the contents of the MPLS Label Forwarding Information Base (LFIB):

```
PE1# show mpls forwarding-table

Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
16         Pop Label  1.1.1.1/32       0             Et0/0      10.0.0.1
18         No Label   nh-id(1)         0             Et2/0      10.0.2.2
19         No Label   nh-id(2)         0             Et1/0      10.0.1.2
```

```
20         No Label   nh-id(3)         0             Et1/0      10.0.1.2
22         No Label   nh-id(5)         0             Et1/0      10.0.1.2
           No Label   nh-id(5)         0             Et2/0      10.0.2.2
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
MPLS commands	Cisco IOS Multiprotocol Label Switching Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2547	*BGP/MPLS*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

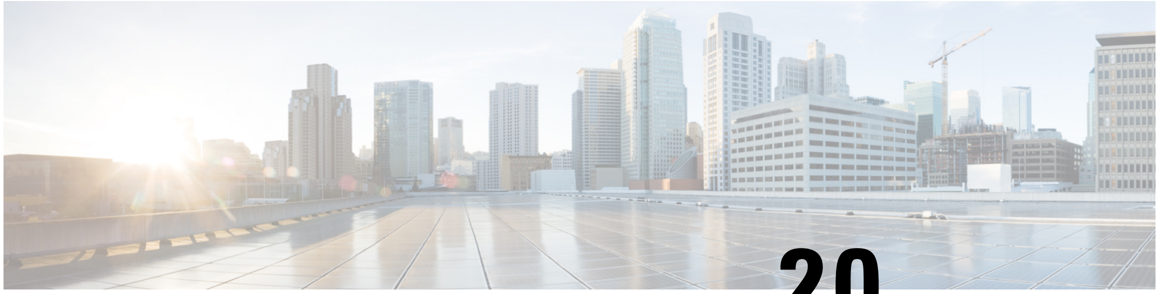
Feature Information for MPLS VPN per CE Label

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 20: Feature Information for MPLS VPN per CE Label

Feature Name	Releases	Feature Information
MPLS VPN per CE Label	15.4(1)T	The MPLS VPN per CE Label feature allows you to configure a single VPN label at the provider edge (PE) for every immediate next hop or set of next hops. The following commands were introduced or modified: **show mpls forwarding-table**, **mpls label mode**.



CHAPTER 20

IPv6 VRF Aware System Message Logging

The IPv6 VRF Aware System Message Logging feature enables a device to send system logging (syslog) messages to an IPv6-enabled syslog server connected through a VPN routing and forwarding (VRF) interface. You can use the logging information for network monitoring and troubleshooting. This feature extends this capability to network traffic connected through VRFs.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for IPv6 VRF Aware System Message Logging

You must configure a VPN routing and forwarding (VRF) instance on a routing device and associate the VRF with an interface before you can configure the IPv6 VRF Aware System Message Logging feature.

Restrictions for IPv6 VRF Aware System Message Logging

You cannot specify a source address for virtual routing and forwarding (VRF) system logging messages. The IPv6 VRF Aware System Message Logging feature uses the VRF interface address as the source address for all VRF aware system logging messages.

Information About IPv6 VRF Aware System Message Logging

Benefits of VRF Aware System Message Logging

A VPN routing and forwarding (VRF) instance is an extension of IP routing that provides multiple routing instances. A VRF provides a separate IP routing and forwarding table to each VPN. You must configure a VRF on a routing device before you configure the VRF Aware System Message Logging feature.

After you configure the VRF Aware System Message Logging feature on a routing device, the device can send system logging (syslog) messages to a syslog host through a VRF interface. Then you can use logging messages to monitor and troubleshoot network traffic connected through a VRF. If the VRF Aware System Message Logging feature is not configured on a routing device, the routing device sends syslog messages to the syslog host only through the global routing table.

You can receive system logging messages through a VRF interface on any device configured with a VRF, that is:

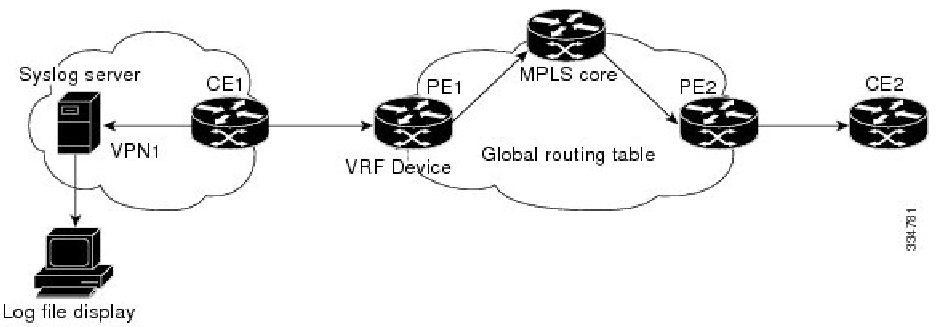
- On a provider edge (PE) device that is used with Multiprotocol Label Switching (MPLS) and multiprotocol Border Gateway Protocol (BGP) to provide a Layer 3 MPLS VPN network service.
- On a customer edge (CE) device that is configured for VRF-Lite, which is a VRF implementation without multiprotocol BGP.

VRF Aware System Message Logging on a Provider Edge Device in an MPLS VPN Network

You can configure the VRF Aware System Message Logging feature on a provider edge (PE) device in a Layer 3 Multiprotocol Label Switching (MPLS) VPN network. The PE device can then send system logging (syslog) messages through a VPN routing and forwarding (VRF) interface to a syslog server located in the VPN.

The figure below shows an MPLS VPN network and the VRF Aware System Message Logging feature configured on a PE device associated with VRF VPN1. The PE device sends log messages through a VRF interface to a syslog server located in VPN1. You can display the messages from the syslog server on a terminal.

Figure 21: MPLS VPN and VRF Aware System Message Logging Configured on a Provider Edge Device

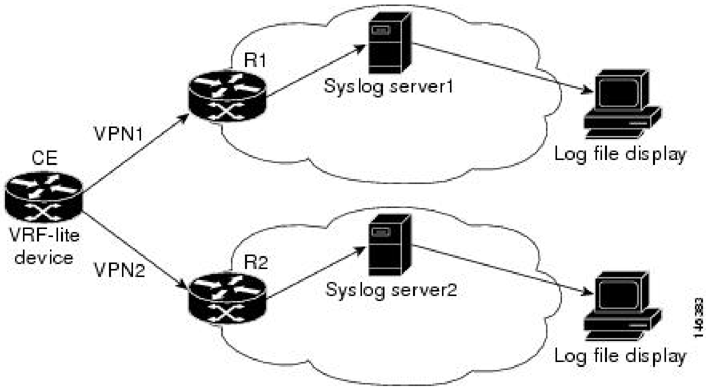


VRF Aware System Message Logging on a Customer Edge Device with VRF-Lite Configured

You can configure the VRF Aware System Message Logging feature on a customer edge (CE) device configured with the VRF-Lite feature. The CE device can then send system logging (syslog) messages through a VPN routing and forwarding (VRF) interface to syslog servers in multiple VPNs. The CE device can be either a router or a switch.

The figure below shows the VRF Aware System Message Logging feature configured on a VRF-Lite CE device. The CE device can send VRF syslog messages to syslog servers in the VPN1 network or the VPN2 network or to servers in both VPN1 and VPN2 networks. You can configure multiple VRFs on a VRF-Lite CE device, and the device can serve many customers.

Figure 22: VRF Aware System Message Logging Configured on a VRF-Lite Customer Edge Device



Message Levels for Logging Commands

The table below lists message levels for **logging** commands that you can use when you configure the VRF Aware System Message Logging feature. Information provided in the table below includes keyword level names and numbers, their description, and the associated syslog definitions. You can use either the level name or the level number with the **logging trap** *level* and **logging buffered** *severity-level* commands.

Table 21: Message Levels for logging Commands

Level Name	Level Number	Description	Syslog Definition
emergencies	**0**	System unusable	LOG_EMERG
alerts	**1**	Immediate action needed	LOG_ALERT
critical	**2**	Critical conditions	LOG_CRIT
errors	**3**	Error conditions	LOG_ERR
warnings	**4**	Warning conditions	LOG_WARNING
notifications	**5**	Normal but significant condition	LOG_NOTICE
informational	**6**	Informational messages only	LOG_INFO
debugging	**7**	Debugging messages	LOG_DEBUG

How to Configure IPv6 VRF Aware System Message Logging

Configuring VRF on a Routing Device

Configuring a VPN routing and forwarding (VRF) instance on a routing device helps provide customer connectivity to a VPN. The routing device can be a provider edge (PE) device connected to a Multiprotocol Label Switching (MPLS) VPN network or a customer edge (CE) device that is configured for VRF-Lite.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **address-family ipv6**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name*	Defines a VRF instance and enters VRF configuration mode.

	Command or Action	Purpose
	Example: `Device (config)# vrf definition vpn1`	- The *vrf-name* argument is a name assigned to the VRF.
Step 4	**address-family ipv6** **Example:** `Device(config-vrf)# address-family ipv6`	Enables IPv6 address-family for the defined VRF and enters address family configuration mode.

Step 5	**end**
	Example:
	`Device(config-vrf-af)# end`
		Exits address family configuration mode and returns to privileged EXEC mode.

Associating a VRF with an Interface

After configuring the VPN routing and forwarding (VRF) instance and associating it with an interface, you can configure the VRF Aware System Message Logging feature on the routing device.

Note You cannot configure a source address for VRF system logging messages. The VRF Aware System Message Logging feature uses the VRF interface address as the source address for all VRF-aware system logging messages.

Before you begin

A VRF must be associated with an interface before you can forward VPN traffic.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name*
5. **no ipv6 address**
6. **ipv6 address** *address.prefix*
7. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *type number* **Example:** `Device (config)# interface FastEthernet 0/0/0`	Configures an interface type and enters interface configuration mode. • The *type* argument is the type of interface to be configured. • The *number* argument is the port, connector, or interface card number. The numbers are assigned at the factory at the time of installation or when the port, connector, or interface card is added to a system. Use the **show interfaces** command in privileged EXEC mode to view the available interfaces.
Step 4	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding vpn1`	Associates a VRF with an interface or subinterface. • The *vrf-name* argument associates the interface with the specified VRF.
Step 5	**no ipv6 address** **Example:** `Device(config-if)# no ipv6 address`	Removes the existing IPv6 address set for an interface.
Step 6	**ipv6 address** *address.prefix* **Example:** `Device(config-if)# ipv6 address 2001:DB8::1/32`	Assigns an IPv6 address for the interface.

Step 7	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Configuring VRF as a Source Interface for Logging on a Routing Device

Before you begin

You must perform the following tasks before you perform this task:

- Configure a virtual routing and forwarding (VRF) instance on a routing device.
- Associate a VRF with an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **logging source-interface** *interface-type interface-number* **vrf** *vrf-name*
4. **logging host ipv6** *ipv6-address* **vrf** *vrf-name*

5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**logging source-interface** *interface-type interface-number* **vrf** *vrf-name* **Example:** `Device (config)# logging source-interface FastEthernet 0/0/0 vrf vpn1`	Configures the VRF interface as the source interface for logging.
Step 4	**logging host ipv6** *ipv6-address* **vrf** *vrf-name* **Example:** `Device(config)# logging host ipv6 2001:DB8:: vrf vpn1`	Configures and associates the IPv6-enabled logging host with the VRF.

Step 5	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Verifying IPv6 VRF Aware System Message Logging

SUMMARY STEPS

1. **enable**
2. **show running-config | include logging**
3. **show logging**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show running-config | include logging**

Displays the logging configuration for the device and the logging host for a virtual routing and forwarding (VRF) instance.

This example shows the configuration of a syslog server in VRF syslog with a server host address of 2001:DB8::1.

Example:

```
Device# show running-config | include logging

logging source-interface Ethernet0/1 vrf syslog
logging host ipv6 2001::DB8:1 vrf syslog
```

Step 3 **show logging**

Displays the state of syslog.

Example:

```
Device# show logging

Trap logging: level informational, 138 message lines logged
Logging to 2001:DB8::1 (v6) (udp port 514, audit disabled,
link up),
24 message lines logged,
0 message lines rate-limited,
0 message lines dropped-by-MD,
xml disabled, sequence number disabled
filtering disabled
Logging to 2001:DB8::1 (syslog) (udp port 514,
audit disabled,
link up),
4 message lines logged,
0 message lines rate-limited,
0 message lines dropped-by-MD,
xml disabled, sequence number disabled
filtering disabled
Logging Source-Interface: VRF Name:
GigabitEthernet0/0/0 syslog
```

Configuration Examples for IPv6 VRF Aware System Message Logging

Example: Configuring VRF on a Routing Device

```
Device> enable
Device# configure terminal
Device(config)# vrf definition syslog_v6
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# end
```

Example: Associating a VRF with an Interface

```
Device> enable
Device# configure terminal
Device(config)# interface FastEthernet 0/0/0
Device(config-if)# vrf forwarding vpn1
Device(config-if)# no ipv6 address
Device(config-if)# ipv6 address 2001:DB8::1/32
Device(config-if)# end
```

Example: Configuring VRF as a Source Interface for Logging on a Routing Device

```
Device> enable
Device# configure terminal
Device(config)# logging source-interface FastEthernet 0/0/0 vrf vpn1
Device(config)# logging host ipv6 address 2001:DB8::1 vrf vpn1
Device(config)# end
```

Additional References for IPv6 VRF Aware System Message Logging

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference
Concepts and tasks for configuring VRF-lite on a Catalyst 4500 switch	"Configuring VRF-lite" chapter in the *Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide*
Concepts and tasks for configuring VRF Lite on ML-Series Ethernet cards	"Configuring VRF-lite" chapter in the *Ethernet Card Software Feature and Configuration Guide* for the Cisco ONS 15454 SDH, ONS 15454, and ONS 15327

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/cisco/web/support/index.html

Feature Information for IPv6 VRF Aware System Message Logging

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 22: Feature Information for IPv6 VRF Aware System Message Logging

Feature Name	Releases	Feature Information
IPv6 VRF Aware System Message Logging	Cisco IOS 15.4(3)M	The IPv6 VRF Aware System Message Logging feature enables a device to send system logging (syslog) messages to an IPv6-enabled syslog server connected through a VPN routing and forwarding (VRF) interface. You can use the logging information for network monitoring and troubleshooting. This feature extends this capability to network traffic connected through VRFs. The following commands were modified: **logging source-interface** and **logging host**.