MPLS Layer 3 VPNs Configuration Guide, Cisco IOS Release 12.2SY

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

CONTENTS

Configuring MPLS Layer 3 VPNs

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 customer site, one or more customer edge (CE) routers attach to one or more provider edge (PE) routers. 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 feature information and caveats, see 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 at the end of this document.

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 Layer 3 VPNs

Before configuring MPLS Layer 3 VPNs, you should have MPLS, Label Distribution Protocol (LDP), and Cisco Express Forwarding installed in your network. All routers in the core, including the PE routers, must be able to support Cisco Express Forwarding and MPLS forwarding. See the Assessing the Needs of MPLS VPN Customers, page 9 for more information.

Cisco Express Forwarding must be enabled all routers in the core, including the PE routers. For information about how to determine if Cisco Express Forwarding is enabled, see Configuring Basic Cisco Express Forwarding--Improving Performance, Scalability, and Resiliency in Dynamic Network .

Restrictions for MPLS Layer 3 VPNs

When configuring static routes in an MPLS or MPLS 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 Cisco IOS releases that support the Tag Forwarding Information Base (TFIB), specifically Cisco IOS Releases 12.xT, 12.xM, and 12.0S. 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 Cisco IOS releases that support the MPLS Forwarding Infrastructure (MFI), specifically Cisco IOS Release 12.2(25)S and later. 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 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** command is 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 a 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 a 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 a MPLS VPN environment and enable load sharing with static nonrecursive routes and a specific outbound interfaces:

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 a 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 a 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 Router

The following **ip route vrf** command is supported when you configure static routes in a MPLS VPN environment, and the next hop is in the global table on the CE side. For example, the following command is supported when the destination-prefix is the CE router's loopback address, as in 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 a MPLS VPN environment, the next hop is in the global table on the CE side, and you enable load sharing with static non-recursive routes and a specific outbound interfaces:

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

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

Information About MPLS Layer 3 VPNs

MPLS VPN Definition

Before defining an MPLS VPN, you need to 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.

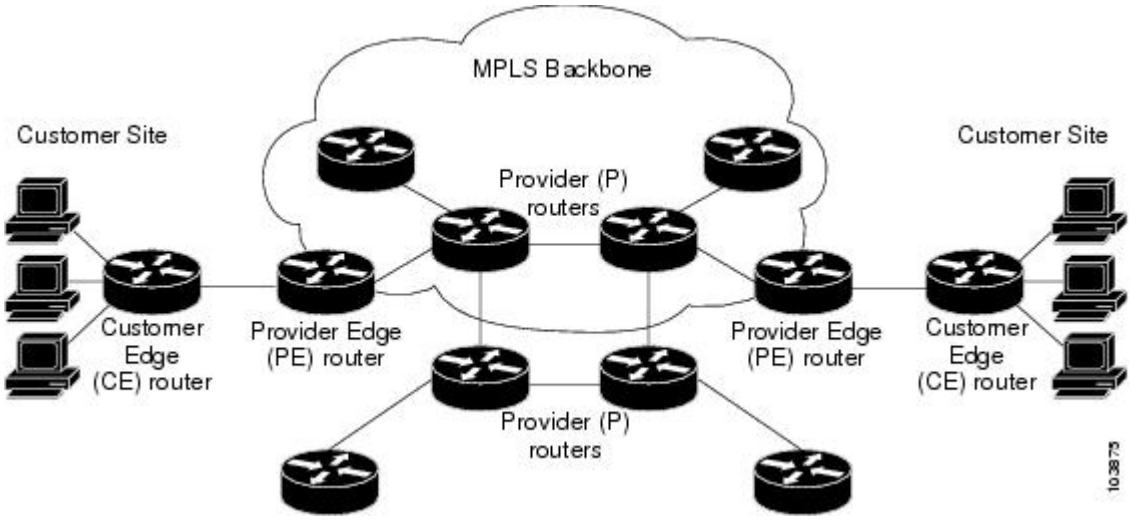
MPLS VPNs are easier to manage and expand than conventional VPNs. When a new site is added to an MPLS VPN, only the service provider’s edge router that provides services to the customer site needs to be updated.

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

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

The figure below shows a basic MPLS VPN.

Figure 1 Basic MPLS VPN Terminology



How an MPLS VPN Works

MPLS VPN functionality is enabled at the edge of an MPLS network. The PE router performs the following:

- Exchanges routing updates with the CE router
- Translates the CE routing information into VPNv4 routes
- Exchanges VPNv4 routes with other PE routers through the Multiprotocol Border Gateway Protocol (MP-BGP)

How Virtual Routing and Forwarding Tables Work in an MPLS VPN

Each 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 router. 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 also prevent packets that are outside a VPN from being forwarded to a router within the VPN.

How VPN Routing Information Is Distributed in an MPLS VPN

The distribution of VPN routing information is controlled through the use of VPN route target communities, implemented by BGP extended communities. VPN routing information is distributed as follows:

- When a VPN route that is learned from a CE router 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 VRF 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.

BGP Distribution of VPN Routing Information

A PE router can learn an IP prefix from the following sources:

- A CE router by static configuration
- A BGP session with the CE router
- A Routing Information Protocol (RIP) exchange with the CE router

The IP prefix is a member of the IPv4 address family. After the PE router 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 VRF on the PE router.

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

- Within 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 EIGRP PE-CE environment, when an EIGRP internal route is redistributed into BGP by one PE, 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 routers 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.

MPLS Forwarding

Based on routing information stored in the VRF IP routing table and VRF Cisco Express Forwarding table, packets are forwarded to their destination using MPLS.

A PE router binds a label to each customer prefix learned from a CE router and includes the label in the network reachability information for the prefix that it advertises to other PE routers. When a PE router forwards a packet received from a CE router across the provider network, it labels the packet with the label learned from the destination PE router. When the destination PE router receives the labeled packet, it pops the label and uses it to direct the packet to the correct CE router. 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 router.
- The second label indicates how that PE router should forward the packet to the CE router.

Major Components of MPLS VPNs

An MPLS-based VPN network 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 PE routers--MP-BGP propagates VRF reachability information to all members of a VPN community. MP-BGP peering needs to be configured in all PE routers 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 VPN

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 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 PE router as opposed to all other customer edge (CE) routers 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 routers and the further partitioning of VPN and IGP routes between PE routers and provider (P) routers in a core network.

- PE routers must maintain VPN routes for those VPNs who are members.
- P routers 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 router) 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.

Easy to Create

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 Quality of Service (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 router and no modifications are required to a customer's intranet.

How to Configure MPLS Layer 3 VPNs

Configuring the Core Network

Assessing the Needs of MPLS VPN Customers

Before you configure an 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 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 routers and ports 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.

	Command or Action	Purpose
Step 3	Determine if you need MPLS VPN High Availability support.	MPLS VPN Nonstop Forwarding and Graceful Restart are supported on select routers and Cisco software releases. Contact Cisco Support for the exact requirements and hardware support.
Step 4	Determine if you need BGP load sharing and redundant paths in the MPLS VPN core.	See *Load Sharing MPLS VPN Traffic* for configuration steps.

Configuring Routing Protocols in the Core

To configure a routing protocol, such as BGP, OSPF, IS-IS, EIGRP,and static, see the following documents:

- Configuring BGP
- Configuring OSPF
- Configuring IS-IS
- Configuring ERGRP
- Configuring static routes

Configuring MPLS in the Core

To enable MPLS on all routers in the core, you must configure a label distribution protocol. You can use either of the following as a label distribution protocol:

- MPLS Label Distribution Protocol (LDP). For configuration information, see the MPLS Label Distribution Protocol (LDP).
- MPLS Traffic Engineering Resource Reservation Protocol (RSVP). For configuration information, see MPLS Traffic Engineering and Enhancements.

Configuring Multiprotocol BGP on the PE Routers and Route Reflectors

Perform this task to configure multiprotocol BGP (MP-BGP) connectivity on the PE routers 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:** `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	**no bgp default ipv4-unicast** **Example:** `Router(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 MPLS routes only.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(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:** `Router(config-router)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.

	Command or Action	Purpose
Step 7	**address-family vpnv4** [**unicast**] **Example:** `Router(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:** `Router(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:** `Router(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 10	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

- Troubleshooting Tips, page 12

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 x.x.x.x events** command, where *x.x.x.x* is the IP address of the neighbor.

Connecting the MPLS VPN Customers

Defining VRFs on the PE Routers to Enable Customer Connectivity

To define VPN routing and forwarding (VRF) instances, perform this task.

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. **import map** *route-map*
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	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vpn1`	Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. • The *vrf-name* argument is the name assigned to a VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Router(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 an 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

	Command or Action	Purpose
Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf)# route-target import 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 (import and export) route-target extended communities.
Step 6	**import map** *route-map* **Example:** `Router(config-vrf)# import map vpn1-route-map`	(Optional) Configures an import route map for a VRF. • The *route-map* argument specifies the route map to be used as an import route map for the VRF.
Step 7	**exit** **Example:** `Router(config-vrf)# exit`	(Optional) Exits to global configuration mode.

Configuring VRF Interfaces on PE Routers for Each VPN Customer

To associate a VRF with an interface or subinterface on the PE routers, 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:** `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	**interface** *type number* **Example:** `Router(config)# interface Ethernet 5/0`	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:** `Router(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:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.

Configuring Routing Protocols Between the PE and CE Routers

Configure the PE router with the same routing protocol that the CE router uses. You can configure the following routing protocols:

- Configuring BGP as the Routing Protocol Between the PE and CE Routers, page 15
- Configuring RIPv2 as the Routing Protocol Between the PE and CE Routers, page 17
- Configuring Static Routes Between the PE and CE Routers, page 19
- Configuring OSPF as the Routing Protocol Between the PE and CE Routers, page 21
- Configuring EIGRP as the Routing Protocol Between the PE and CE Routers, page 23
- Configuring EIGRP Redistribution in the MPLS VPN, page 26

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

To configure PE-to-CE routing sessions using BGP, perform this task.

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:** `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	**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.

	Command or Action	Purpose
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor 10.0.0.1 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 8	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

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

To configure PE-to-CE routing sessions using RIPv2, perform this task.

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:** `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 rip** **Example:** `Router(config)# router rip`	Enables RIP.
Step 4	**version** {**1** | **2**} **Example:** `Router(config-router)# version 2`	Specifies a Routing Information Protocol (RIP) version used globally by the router.
Step 5	**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 specifies the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**network** *ip-address* **Example:** `Router(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:** `Router(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:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 9	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Configuring Static Routes Between the PE and CE Routers

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

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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip route vrf** *vrf-name* **Example:** `Router(config)# ip route vrf 200`	Defines static route parameters for every PE-to-CE session.
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 specifies 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:** `Router(config-router-af)# redistribute static`	Redistributes routes from one routing domain into another routing domain. • To redistribute VRF static routes into the VRF BGP table, use the **redistribute static** command. See the command for information about other arguments and keywords.

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

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

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

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:** `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 ospf** *process-id* [**vrf** *vpn-name*] **Example:** `Router(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 VPN. Create a separate OSPF process for each VRF that will receive VPN routes.
Step 4	**network** *ip-address wildcard-mask* **area** *area-id* **Example:** `Router(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 as 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:** `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.

	Command or Action	Purpose
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:** `Router(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 IBGP routes are distributed into the VRF.
Step 7	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 8	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

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

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

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

BGP must be configured in the network core.

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:** `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 10`	Enters router configuration mode, and creates a BGP routing process.
Step 4	**no synchronization** **Example:** `Router(config-router)# no synchronization`	Configures BGP to send advertisements without waiting to synchronize with the IGP.

	Command or Action	Purpose
Step 5	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Router(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 router that is connected to the CE router at the other CE site.
Step 6	**neighbor** *ip-address update-source* **loopback** *interface-number* **Example:** `Router(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 affects of interface or link flapping.
Step 7	**address-family vpnv4** **Example:** `Router(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:** `Router(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 routers.
Step 9	**neighbor** *ip-address* **send-community extended** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 send-community extended`	Configures the local router 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:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode and enters router configuration mode.

	Command or Action	Purpose
Step 11	**address-family ipv4 vrf** *vrf-name* **Example:** `Router(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 routers.
Step 12	**redistribute eigrp** *as-number* [**metric** *metric-value*] [**route-map** *map-name*] **Example:** `Router(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:** `Router(config-router-af)# no synchronization`	Configures BGP to send advertisements without waiting to synchronize with the IGP.
Step 14	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode and enters router configuration mode.
Step 15	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.

Configuring EIGRP Redistribution in the MPLS VPN

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

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 router. The metric can be configured in the redistribute statement using the redistribute (IP) command or 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 router.

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:** `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 eigrp** *as-number* **Example:** `Router(config)# router eigrp 1`	Enters router configuration mode and creates an EIGRP routing process. • The EIGRP routing process for the PE router is created in this step.
Step 4	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Router(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.

	Command or Action	Purpose
Step 5	**network** *ip-address wildcard-mask* **Example:** `Router(config-router-af)# network 172.16.0.0 0.0.255.255`	Specifies the network for the VRF. • 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:** `Router(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 is 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:** `Router(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:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode and enters router configuration mode.
Step 9	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.

Verifying the VPN Configuration

A route distinguisher must be configured for the VRF, and 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
Use this command to display 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 VPN Sites

To verify that the local and remote CE routers can communicate across the MPLS core, perform the following tasks:

- Verifying IP Connectivity from CE Router to CE Router Across the MPLS Core, page 29
- Verifying that the Local and Remote CE Routers Are in the Routing Table, page 30

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

Perform this task to verify IP connectivity from CE router to CE router across the MPLS VPN.

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**
Use this command to enable privileged EXEC mode.

Step 2 **ping** [*protocol*] {*host-name* | *system-address*}
Use this command to diagnoses basic network connectivity on AppleTalk, CLNS, IP, Novell, Apollo, VINES, DECnet, or XNS networks. Use the **ping** command to verify the connectivity from one CE router to another.

Step 3 **trace** [*protocol*] [*destination*]
Use this command to discover the routes that packets take when traveling to their destination. Use the **trace** command to verify the path that a packet goes through before reaching the final destination. The **trace** command can help isolate a trouble spot if two routers cannot communicate.

Step 4 **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | *protocol* [*process-id*]] | [**list** [*access-list-name* |*access-list-number*]
Use this command to display 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 Routers Are in the Routing Table

Perform this task to check that the local and remote CE routers are in the routing table of the PE routers.

SUMMARY STEPS

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

DETAILED STEPS

Step 1 **enable**
Use this command to enable privileged EXEC mode.

Step 2 **show ip route vrf** *vrf-name* [*prefix*]
Use this command to display the IP routing table associated with a VRF. Check that the loopback addresses of the local and remote CE routers are in the routing table of the PE routers.

Step 3 **show ip cef vrf** *vrf-name* [*ip-prefix*]
Use this command to display the Cisco Express Forwarding forwarding table associated with a VRF. Check that the prefix of the remote CE router is in the Cisco Express Forwarding table.

Step 4 **exit**

Configuration Examples for MPLS VPNs

Configuring an MPLS VPN Using BGP Example

This example shows an MPLS VPN that is configured using BGP.

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 Ethernet0/0 ip vrf forwarding vpn1 ip address 34.0.0.2 255.0.0.0 no cdp enable ! interface Ethernet 1/1 ip address 30.0.0.1 255.0.0.0 mpls label protocol ldp mpls ip ! router ospf 100 network 10.0.0. 0.0.0.0 area 100 network 30.0.0.0 0.255.255.255 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 34.0.0.1 remote-as 200 neighbor 34.0.0.1 activate neighbor 34.0.0.1 as-override neighbor 34.0.0.1 advertisement-interval 5 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 Ethernet0/0 ip address 34.0.0.1 255.0.0.0 no cdp enable ! router bgp 200 bgp log-neighbor-changes neighbor 34.0.0.2 remote-as 100 ! address-family ipv4 redistribute connected neighbor 34.0.0.2 activate neighbor 34.0.0.2 advertisement-interval 5 no auto-summary no synchronization exit-address-family

Configuring an MPLS VPN Using RIP Example

This example shows an MPLS VPN that is configured using RIP.

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 Ethernet0/0 ip vrf forwarding vpn1 ip address 34.0.0.2 255.0.0.0 no cdp enable interface Ethernet 1/1 ip address 30.0.0.1 255.0.0.0 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 34.0.0.0 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	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 Ethernet0/0 ip address 34.0.0.1 255.0.0.0 no cdp enable router rip version 2 timers basic 30 60 60 120 redistribute connected network 10.0.0.0 network 34.0.0.0 no auto-summary

Configuring an MPLS VPN Using Static Routes Example

This example shows an MPLS VPN that is configured 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 Ethernet0/0 ip vrf forwarding vpn1 ip address 34.0.0.2 255.0.0.0 no cdp enable ! interface Ethernet 1/1 ip address 30.0.0.1 255.0.0.0 mpls label protocol ldp mpls ip ! router ospf 100 network 10.0.0. 0.0.0.0 area 100 network 30.0.0.0 0.255.255.255 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 34.0.0.1 ip route vrf vpn1 34.0.0.0 255.0.0.0 34.0.0.1	ip cef ! interface Loopback0 ip address 10.0.0.9 255.255.255.255 ! interface Ethernet0/0 ip address 34.0.0.1 255.0.0.0 no cdp enable ! ip route 10.0.0.9 255.255.255.255 34.0.0.2 3 ip route 31.0.0.0 255.0.0.0 34.0.0.2 3

Configuring an MPLS VPN Using OSPF Example

This example shows an MPLS VPN that is configured 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 Ethernet0/0 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 Ethernet0/0 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

Configuring an MPLS VPN Using EIGRP Example

This example shows an MPLS VPN that is configured using EIGRP.

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 Ethernet0/0 ip vrf forwarding vpn1 ip address 34.0.0.2 255.0.0.0 no cdp enable interface Ethernet 1/1 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	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 Ethernet0/0 ip address 34.0.0.1 255.0.0.0 no cdp enable ! router eigrp 1000 network 34.0.0.0 auto-summary

Additional References

Related Documents

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

Feature Information for MPLS Layer 3 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 1 Feature Information for MPLS Layer 3 VPNs

Feature Name	Releases	Feature Configuration Information
MPLS Virtual Private Networks	12.0(5)T 12.0(21)ST 12.0(22)S 12.0(23)S 12.2(13)T 12.2(14)S 12.0(26)S	This feature allows a set of sites that to be interconnected by means of an MPLS provider core network. At each customer site, one or more customer edge (CE) routers attach to one or more provider edge (PE) routers.
MPLS VPN Support for EIGRP Between Provider Edge and Customer Edge	12.0(22)S 12.2(15)T 12.2(18)S 12.0(27)S	This feature allows you to connect customers running EIGRP to an MPLS VPN.

Assigning an ID Number to a VPN

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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 at the end of this document.

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.

Information About VPN ID

Introduction to VPN ID

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. 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 MP-BGP VPNv4 routing updates.

Multiple VPNs can be configured in a router. 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 router. Alternately, you can use a VPN ID to identify a particular VPN in the router. 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 routers 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 router. The VPN name is not affected by the VPN ID configuration. These are two independent mechanisms to identify VPNs.

Components of the VPN ID

Each 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 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 VPN IDs

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

- Dynamic Host Configuration Protocol, page 40
- Remote Authentication Dial-In User Service, page 40

Dynamic Host Configuration Protocol

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

1. A VPN DHCP client requests a connection to a provider edge (PE) router from a VRF interface.
2. The PE router determines the VPN ID associated with that interface.
3. The PE router 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 router, 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, 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 a VPN ID

Specifying a VPN ID

Use this procedure to specify a VPN ID.

Restrictions

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 MP-BGP VPNv4 routing updates.

Each VRF configured on a PE router can have a VPN ID configured. Configure all the PE routers that belong to the same 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vrf1`	Creates a VRF routing table and a CEF forwarding table and enters VRF configuration mode. • *vrf-name* --Name assigned to a VRF.
Step 4	**vpn id** *oui*:*vpn-index* : **Example:** `Router(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 VPN ID Configuration

To verify the VPN ID configuration, perform the following steps.

SUMMARY STEPS

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

DETAILED STEPS

Step 1 **show ip vrf**

Use this command to display information about the VRF tables on the PE router. This example displays three VRF tables called vpn1, vpn2, and vpn5.

Example:

```
Router# show ip vrf
```

```
  Name                             Default RD          Interfaces
  vpn1                             100:1               Ethernet1/1
                                                       Ethernet1/4
  vpn2                             <not set>
  vpn5                             500:1               Loopback2
```

Step 2 **show ip vrf id**

Use this command to ensure that the PE router contains the 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:

```
Router# show ip vrf id
VPN Id          Name                             RD
2:3             vpn2                             <not set>
A1:3F6C         vpn1                             100:1
```

Step 3 **show ip vrf detail**

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

Example:

```
Router# show ip vrf detail
VRF vpn1; default RD 100:1; default VPNID A1:3F6C
  Interfaces:
    Ethernet1/1             Ethernet1/4
  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
Basic MPLS VPNs	Configuring MPLS Layer 3 VPNs

Related Topic	Document Title
MPLS VPN Carrier Supporting Carrier	• MPLS VPN Carrier Supporting Carrier Using LDP and an IGP • MPLS VPN Carrier Supporting Carrier with BGP
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
IEEE Std 802-1990	IEEE Local and Metropolitan Area Networks: Overview and Architecture

MIBs

MIB	MIBs Link
None	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 2685	Virtual Private Networks Identifier

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 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 Assigning an ID Number to a VPN

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

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

Table 2 Feature Information for Assigning an ID Number to a VPN

Feature Name	Releases	Feature Configuration Information
VPN ID	12.0(17)ST 12.2(4)B 12.2(8)T 12.2(14)S	This feature lets you you identify VPNs by a VPN identification number, as described in RFC 2685.

Multi-VRF Selection Using Policy-Based Routing (PBR)

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.

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.

This feature and the Directing MPLS VPN Traffic Using a Source IP Address feature can be configured together on the same interface.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 at the end of this document.

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 router must support policy-based routing (PBR) in order for you to configure this feature. For platforms that do not support PBR, use the Directing MPLS VPN Traffic Using a Source IP Address feature.
- A VRF must be defined before you configure this feature. An error message is displayed on the console if no VRF exists.

Restrictions for Multi-VRF Selection Using Policy-Based Routing

- 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 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 (PBR) 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 (PBR) feature supports VRF-lite; that is, only IP routing protocols run on the router. Multiprotocol Label Switching (MPLS) and VPN cannot be configured.

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 VRF Selection Based on Source IP Address feature. The PBR implementation of the VRF selection feature allows you to policy route 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 Ciscosoftware:

- 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 VRF.

Policy-Based Routing set Commands

Policy-routing Packets for VRF Instances

To enable policy-routing packets for VRF instances, you can use route map commands with the following **set** commands. They are listed in the order in which the router 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 packets that pass a match criteria of a route map for policy routing when the next hop must be under a specified VRF.
- **set ip global next-hop**--Indicates where to forward packets that pass a match criterion of a route map for policy routing and for which the Cisco IOS software uses 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 inherit-VRF and inter-VRF routing. With inherit-VRF routing, packets arriving at a VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, packets arriving at a VRF interface are routed via any other outgoing VRF interface.
- **set ip default global**--Provides 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 global next-hop**--Routes packets through the global routing table, where the next hop lookup will be in the global routing table.
- **set ip default next-hop**--Indicates where to output packets that pass a match criterion of a route map for policy routing and for which the Cisco IOS software has no explicit route to a destination.

Change of Normal Routing and Forwarding Behavior

When you configure PBR, you can use the following four **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 VRF. 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, 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 next-hop**--Indicates where to output packets that pass a match criterion of a route map for policy routing and for which the Cisco IOS software has no explicit route to a destination.
- **set ip next-hop**--Indicates where to output packets that pass a match criterion of a route map for policy routing. If a packet is received on a VRF interface and is transmitted from another interface within the same VPN, the VRF context of the incoming packet will be 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 VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, packets arriving at a VRF interface are routed via any other outgoing VRF interface.

VRF-to-global routing causes packets that enter any VRF interface to be routed via 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 (PBR) 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 router 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 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 router to look up the next hop in the VRF table. If a packet arrives on an interface that belongs to a VRF and the packet needs to be routed via a different VRF, you can use the **set ip vrf next-hop** command.
- **set ip default vrf**—Provides inherit-VRF and inter-VRF routing. With inherit-VRF routing, packets arriving at a VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, packets arriving at a VRF interface are routed via 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 packets through the global routing table in an IP-to-IP 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 PBR

Define the match criteria for multi-VRF selection using PBR so that you can selectively route the packets instead of using their default routing and forwarding.

The match criteria for multi-VRF selection using PBR 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 PBR with a Standard Access List

The tasks in the following sections assume that the VRF 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:** `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	**access-list** *access-list-number* {**deny** | **permit**} [**source** *source-wildcard*] [**log**] **Example:** `Router(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 PBR with a Named Extended Access List

To configure Multi-VRF Selection using PBR with a named extended access list, complete the following steps.

The tasks in the following sections assume that the VRF 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:** `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	**ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] **Example:** `Router(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:** `Router(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 vrf** command configuration determines the VRF through which the outbound VPN packets will be policy routed.

You must define the VRF 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. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. Do one of the following:
   - **set ip vrf** *vrf-name* **next-hop** *ip-address* [*...ip-address*]
   - or
   - **set ip next-hop recursive vrf** *ip-address* [*...ip-address*]
   - or
   - **set ip global next-hop** *ip-address* [*...ip-address*]
5. Do one of the following:
   - **match ip address** {*acl-number* [*acl-name* | *acl-number*]}
   - or
   - **match length** *minimum-lengthmaximum-length*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-tag* [**permit**	**deny**] [*sequence-number*] **Example:** `Router(config)# route-map map1 permit 10`	Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing. • Enters route-map configuration mode.

Command or Action	Purpose

Step 4 Do one of the following:
• **set ip vrf** *vrf-name* **next-hop** *ip-address* [*...ip-address*]
• or
• **set ip next-hop recursive vrf** *ip-address* [*...ip-address*]
• or
• **set ip global next-hop** *ip-address* [*...ip-address*]
Example:
`Router(config-route-map)# set ip vrf myvrf next-hop 10.0.0.0`
Example:
`Router(config-route-map)# set ip next-hop recursive vrf 10.0.0.0`
Example:
`Router(config-route-map)# set ip global next-hop 10.0.0.0`
	Indicates where to forward packets that pass a match criterion of a route map for policy routing when the next hop must be under a specified VRF.
	or
	Indicates which destination or next hop will be used for packets that pass the match criterion configured in the route map.
	or
	Indicates where 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.

Step 5 Do one of the following:
• **match ip address** {*acl-number* [*acl-name* | *acl-number*]}
• or
• **match length** *minimum-lengthmaximum-length*
Example:
`Router(config-route-map)# match ip address 1 or`
Example:
`Router(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 length.

Step 6 **end** **Example:** `Router(config-route-map)# end`	Exits route-map configuration mode and returns to privileged EXEC mode.

Configuring Multi-VRF Selection Using PBR 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 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface FastEthernet 0/1/0`	Configures an interface and enters interface configuration mode.
Step 4	**ip policy route-map** *map-tag* **Example:** `Router(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.

	Command or Action	Purpose
Step 5	**ip vrf receive** *vrf-name* **Example:** `Router(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:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying the Configuration of Multi-VRF Selection Using PBR

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*]

To verify the configuration of match criteria for PBR multi-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:

Example:

```
Router# 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*]

Use this command to verify **match** and **set** commands within the route map:

Example:

```
Router# show route-map
```

To verify the 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.

Example:

```
Router# 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
Router# show route-map map2
route-map map2, permit, sequence 10
Match clauses:
Set clauses:
 vrf myvrf
Policy routing matches: 0 packets, 0 bytes
Router# 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:

```
Router(config)# route-map test

Router(config-route-map)# set ip vrf myvrf next-hop
Router(config-route-map)# set ip vrf myvrf next-hop 192.168.3.2
Router(config-route-map)# match ip address 255 101
Router(config-route-map)# end
Router# 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:

```
Router(config)# route-map test
Router(config-route-map)# match ip address 255 101
Router(config-route-map)# set ip global next-hop 192.168.4.2
Router(config-route-map)# end
Router# 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**

To verify the PBR multi-VRF selection policy, use the **show ip policy** command:

Example:

```
Router# show ip policy
```

The following **show ip policy** command output displays the interface and associated route map that is configured for policy routing:

Example:

```
Router# show ip policy
Interface                Route map
FastEthernet0/1/0        PBR-VRF-Selection
```

Configuration Examples for Multi-VRF Selection Using Policy-Based Routing

Defining the Match Criteria for Multi-VRF Selection Using PBR Example

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 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 vrf forwarding VRF4
 ip policy route-map PBR-VRF-Selection
 ip vrf receive VRF1
 ip vrf receive VRF2
 ip vrf receive VRF3
```

Configuring Multi-VRF Selection in a Route Map Example

The following example shows a **set ip vrf next-hop** command that applies policy-based routing to the VRF interface named myvrf and specifies that the IP address of the next hop is 10.0.0.2:

```
Router(config)# route-map map1 permit
Router(config)# set vrf myvrf
Router(config-route-map)# set ip vrf myvrf next-hop 10.0.0.2
Router(config-route-map)# match ip address 101
Router(config-route-map)# end
```

The following example shows a **set ip global** command that specifies that the router should use the next hop address 10.0.0.1 in the global routing table:

```
Router(config-route-map)# set ip global next-hop 10.0.0.1
```

Additional References

Related Documents

Related Topic	Document Title
MPLS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Multiprotocol Label Switching Command Reference*
IP access list commands	*Cisco IOS Security Command Reference*

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

Technical Assistance

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

Feature Information for 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 3 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	The Multi-VRF Selection Using Policy-Based Routing (PBR) feature allows a specified interface on a provider edge (PE) router to route packets to VPNs based on packet length or match criteria defined in an IP access list. This feature and the *Directing MPLS VPN Traffic Using a Source IP Address* feature can be configured together on the same interface. In 12.2(33)SRB1, this feature was introduced. In 12.2(33)SXH1, support was added. The following commands were modified: **set ip global next-hop** and **set ip vrf next-hop**. In 12.4(24)T, this feature was integrated.

Glossary

CE router—customer edge router. A router that is part of a customer network and that interfaces to a provider edge (PE) router.

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 router—provider edge router. A router that is part of a service provider's network and that is connected to a CE router. 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.

VRF Aware System Message Logging

The VRF Aware System Message Logging (Syslog) feature allows a router to send system logging (syslog) messages to a syslog server host connected through a Virtual Private Network (VPN) routing and forwarding (VRF) interface.

You can use 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 feature information and caveats, see 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 at the end of this document.

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 VRF Aware System Message Logging

You must configure a VRF on a routing device and associate the VRF with an interface (see Associating a VRF with an Interface, page 70) before you can configure the VRF Aware System Message Logging feature.

Restrictions for VRF Aware System Message Logging

You cannot specify 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.

Information About VRF Aware System Message Logging

VRF Aware System Message Logging Benefit

A VPN routing and 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 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. Without the VRF Aware System Message Logging feature on a routing device, you do not have this benefit; the routing device can send syslog messages to the syslog host only through the global routing table.

You can receive system logging messages through a VRF interface on any router where you can configure a VRF, that is:

- On a provider edge (PE) router 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 (switch or router) that is configured for VRF-Lite, which is a VRF implementation without multiprotocol BGP.

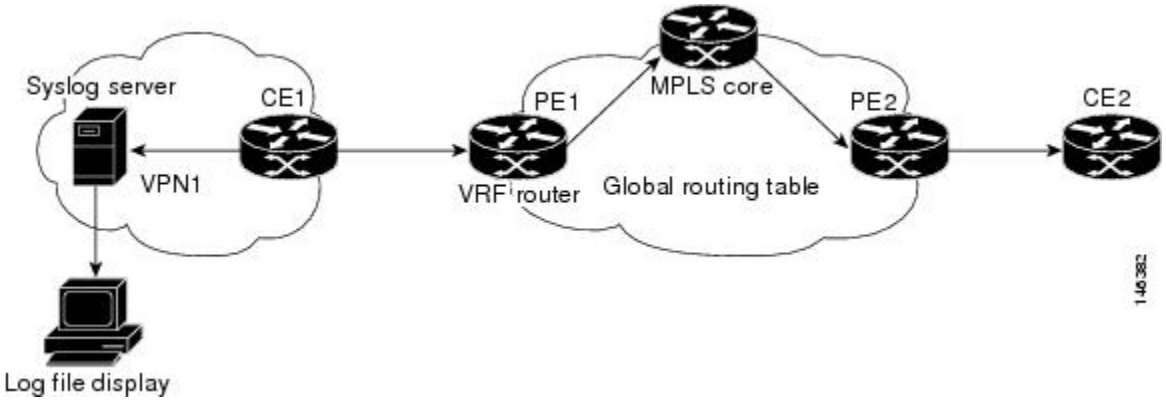
VRF Aware System Message Logging on a Provider Edge Router in an MPLS VPN Network

You can configure the VRF Aware System Message Logging feature on a PE router in a Layer 3 MPLS VPN network. The PE router can then send syslog messages through a 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 router associated with VRF VPN1. The PE router 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 2 MPLS VPN and VRF Aware System Message Logging Configured on a Customer Edge Router

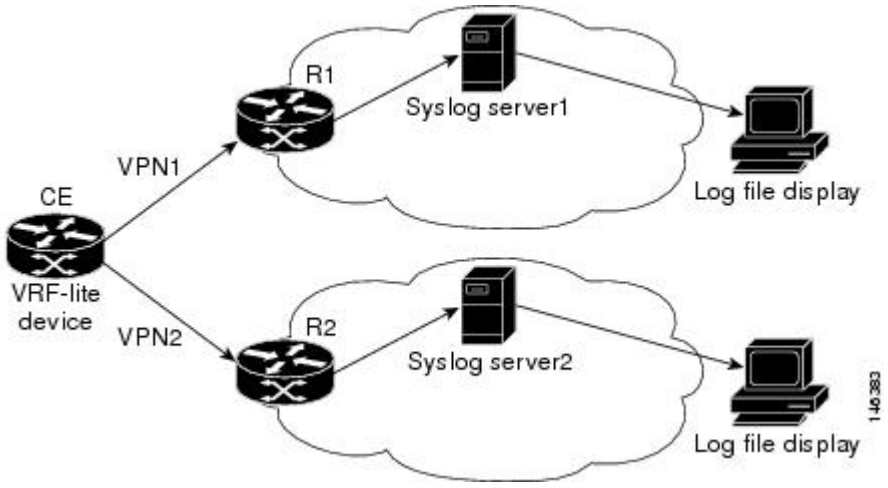


VRF Aware System Message Logging on a Customer Edge Device with VRF-Lite Configured

You can configure the VRF Aware System Message Logging feature on a CE device where you have configured the VRF-Lite feature. The CE device can then send syslog messages through a 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 VPN1 or VPN2 or to servers in both VPN1 and VPN2. You can configure multiple VRFs on a VRF-Lite CE device, and the device can serve many customers.

Figure 3 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 by the table below includes keyword level names and numbers, their description, and the associated syslog definitions. You can use either the level keyword name or number with the **logging trap** *level* and **logging buffered** *severity-level* commands.

Table 4 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 and Verify VRF Aware System Message Logging

Configuring a VRF on a Routing Device

Configuring a VRF on a routing device helps provides customer connectivity to a VPN. The routing device can be a PE router connected to an MPLS VPN network or a CE (switch or router) that is configured for VRF-Lite.

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. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vpn1`	Defines a VRF and enters VRF configuration mode. • The *vrf-name*argument is a name assigned to the VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 100:1`	Creates routing and forwarding tables for a VRF. • The *route-distinguisher* argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. • The route distinquisher (RD) is either an autonomous system number (ASN)-relative RD, in which case it is composed of an autonomous system number and an arbitrary number, or it is an IP-address-relative RD, in which case it is composed of an IP address and an arbitrary number. • You can enter an RD in either of these formats: ◦ 16-bit autonomous system number: your 32-bit number For example, 101:3. ◦ 32-bit IP address: your 16-bit number For example, 10.0.0.1:1.

	Command or Action	Purpose
Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(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 (import and export) route-target extended communities. The route target specifies a target VPN extended community. Like a route distinguisher, an extended community is composed of either an autonomous system number and an arbitrary number or an IP address and an arbitrary number. You can enter the numbers in either of these formats: • 16-bit autonomous system 1 32-bit number For example, 101:3. • 32-bit IP address: your 16-bit number For example, 10.0.0.2.15: 1.
Step 6	**end** **Example:** `Router(config-vrf)# end`	Exits to privileged EXEC mode.

Associating a VRF with an Interface

Perform this task to associate a VRF instance with an interface. A VRF must be associated with an interface before you can forward VPN traffic.

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.

After configuring the VRF and associating it with an interface, you can configure the VRF Aware System Message Logging feature on the routing device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip vrf forwarding** *vrf-name*
5. **end**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface FastEthernet 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. On Cisco 4700 series routers, it specifies the network interface module (NIM) or network processor module (NPM) 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, and can be displayed with the **show interfaces** command.
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vpn1`	Associates a VRF with an interface or subinterface. • The *vrf-name* argument associates the interface with the specified VRF.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.
Step 6	**copy running-config startup-config** **Example:** `Router# copy running-config startup-config`	(Optional) Saves configuration changes to NVRAM.

Configuring VRF Aware System Message Logging on a Routing Device

Configure the VRF Aware System Message Logging feature on a routing device so that logging messages can be used to monitor and troubleshoot network traffic connected through VRF instances.

You must perform the following tasks before you perform this task:

- Configure a VRF on a routing device.
- Associate a VRF with an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **logging host** {*ip-address* | *hostname*} [**vrf** *vrf-name*]
4. **logging trap** *level*
5. **logging facility** *facility-type*
6. **logging buffered** [*buffer-size* | *severity-level*]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**logging host** {*ip-address* | *hostname*} [**vrf** *vrf-name*] **Example:** `Router(config)# logging host 10.0.150.63 vrf vpn1`	Specifies a host to receive syslog messages. • The *ip-address* argument is the IP address of the syslog server host. • The *hostname* argument is the name of the IP or IPv6 host that receives the syslog messages. • The **vrf** *vrf-name* keyword argument pair specifies a VRF that connects to the syslog server host.
Step 4	**logging trap** *level* **Example:** `Router(config)# logging trap debugging`	Limits messages logged to the syslog servers based on severity. • The *level* argument limits the logging of messages to the syslog servers to a specified level. You can enter the level number or level name. See the "Message Levels for Logging Commands" section for a description of acceptable keywords.

	Command or Action	Purpose
Step 5	**logging facility** *facility-type* **Example:** `Router(config)# logging facility local6`	(Optional) Configures the syslog facility in which error messages are sent. • The *facility-type* argument names the syslog facility type keyword. For locally defined messages, the range of acceptable keywords is local0 to local7. The default is **local7**.
Step 6	**logging buffered** [*buffer-size* | *severity-level*] **Example:** `Router(config)# logging buffered debugging`	(Optional) Limits messages logged to an internal buffer on the router based on severity. • The*buffer-size* argument is the size of the buffer from 4096 to 4,294,967,295 bytes. The default size varies by platform. • The *severity-level* argument limits the logging of messages to the buffer to a specified level. You can enter the level name or level number. See the "Message Levels for Logging Commands" section for a list of the acceptable level name or level number keywords. The default logging level varies by platform, but is generally 7, meaning that messages at all levels (0–7) are logged to the buffer.
Step 7	**end** **Example:** `Router(config)# end`	(Optional) Exits to privileged EXEC mode.

Verifying VRF Aware System Message Logging Operation

SUMMARY STEPS

1. **enable**
2. **show running-config | include logging**
3. **show ip vrf interfaces**
4. **show running-config** [**interface** *type number*]
5. **ping vrf** *vrf-name target-ip-address*
6. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. You can also enter this command in user EXEC mode. Enter your password if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 **show running-config | include logging**

Use this command to display the logging configuration for the router and the logging host for a VRF. For example:

Example:

```
Router# show running-config | include logging
logging queue-limit 100
logging buffered 100000 debugging
mpls ldp logging neighbor-changes
logging trap debugging
logging facility local6
logging host vrf vpn1 10.0.0.3
Router#
```

This example shows the configuration of a syslog server in VRF vpn1 with a server host address of 10.0.0.3.

Step 3 **show ip vrf interfaces**

Use this command to display the interfaces associated with the VRF that links to a syslog server host. The following example displays a list of VRF interfaces and their associated IP addresses that are configured on the router:

Example:

```
Router# show ip vrf interfaces
Interface              IP-Address      VRF                              Protocol
FastEthernet0/0/0      10.0.0.0        vpn1                              up
Loopback1              10.0.0.6        vpn1                              up
```

Step 4 **show running-config** [**interface** *type number*]

Use this command to display interface specific configuration information for an interface associated with a VRF. For example:

Example:

```
Router# show running-config interface FastEthernet 0/0/0
Building configuration...
Router#
.
.
.
!
Current configuration : 116 bytes
!
interface FastEthernet0/0/0
 ip vrf forwarding vpn1
 ip address 10.0.0.98 255.0.0.0
 duplex half
 no cdp enable
end
```

This example displays configuration information for Fast Ethernet interface 0/0/0 in VRF vpn1.

Step 5 **ping vrf** *vrf-name target-ip-address*

Use this command to verify that you can reach the syslog server host, the *target-ip-address*, through the specified VRF. For example:

Example:

```
Router# ping vrf vpn1 10.3.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.0.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

In this example, the syslog server has an IP address of 10.3.0.1 and the VRF is named vpn1. The server is reached successfully four of five times.

Step 6 **exit**

Use this command to exit privileged EXEC mode. For example:

Example:

```
Router# exit
Router>
```

Configuration Examples for VRF Aware System Message Logging

Example Configuring a VRF on a Routing Device

```
enable
configure terminal
!
ip vrf vpn1
 rd 100:1
 route-target both 100:1
 end
```

Example Associating a VRF with an Interface

```
enable
configure terminal
!
interface FastEthernet 0/0/0
 ip vrf forwarding vpn1
 end
```

Example Configuring VRF Aware System Message Logging on a Routing Device

The following example shows how to configure the VRF Aware System Message Logging feature on a routing device. The IP address of the syslog server host is 10.0.1.3 and the VRF is vpn1.

```
enable
configure terminal
!
 logging host 10.0.1.3 vrf vpn1
 logging trap debugging
 logging facility local6
 logging buffered 10000
 logging buffered debugging
 end
```

The following example shows how to turn off logging to the syslog server:

```
enable
configure terminal
!
 no logging 10.0.1.3
 end
```

Additional References

Related Documents

Related Topic	Document Title
Concepts and tasks for configuring MPLS VPNs	Configuring MPLS Layer 3 VPNs
Basic tasks for troubleshooting your system and the network	Troubleshooting and Fault Management
Description of commands associated with MPLS and MPLS applications	*Cisco IOS Multiprotocol Label Switching Command Reference*
Concepts and tasks for configuring VRF-lite on a Catalyst 4500 switch	" Configuring VRF-lite" chapter, 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, Ethernet Card Software Feature and Configuration Guide for the Cisco ONS 15454 SDH, ONS 15454, and ONS 15327

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

Technical Assistance

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

Feature Information for 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 5 Feature Information for VRF Aware System Message Logging

Feature Name	Releases	Feature Information
VRF Aware System Message Logging (Syslog)	12.4(4)T 12.2(33)SRA 12.2(31)SB2 12.4(13) 12.2(33)SXH	The VRF Aware System Message Logging feature allows a router to send syslog messages to a syslog server host connected through a VPN VRF interface. In 12.4(4)T, this feature was introduced. In 12.2(33)SRA, this feature was integrated. In 12.2(31)SB2, support was added for the Cisco 10000 series routers. In 12.4(13), this feature was integrated. In 12.2(33)SXH, this feature was integrated. The following command was modified by this feature: **logging host**.

Glossary

CE router --customer edge router. A router on the border between a VPN provider and a VPN customer that belongs to the customer.

LSR --label switching router. A device that forwards MPLS packets based on the value of a fixed-length label encapsulated in each packet.

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 VPN --Multiprotocol Label Switching Virtual Private Network. An IP network infrastructure delivering private network services over a public infrastructure using a Layer 3 backbone. Using MPLS VPNs in a Cisco network provides the capability to deploy and administer scalable Layer 3 VPN backbone services including applications, data hosting network commerce, and telephony services to business customers.

PE router --provider edge router. A router on the border between a VPN provider and a VPN customer that belongs to the provider.

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. 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. *Seealso* MPLS VPN.

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 PE router.

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) routers 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 feature information and caveats, see 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 at the end of this document.

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 Virtual Private Networks (MPLS 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 router 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

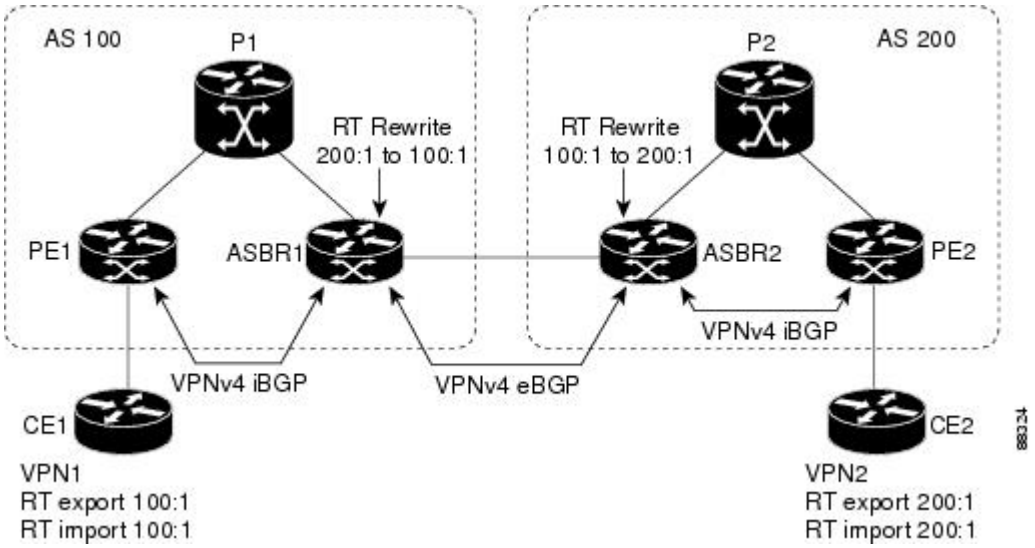
Routing policies for a peer include all configurations that may impact inbound or outbound routing table updates. The MPLS VPN Route Target Rewrite feature can influence routing table updates by allowing the replacement of route targets on inbound and outbound BGP updates. Route targets are carried as 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, 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 PE routers and RR routers.

The figure below shows an example of route target replacement on ASBRs in an 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.

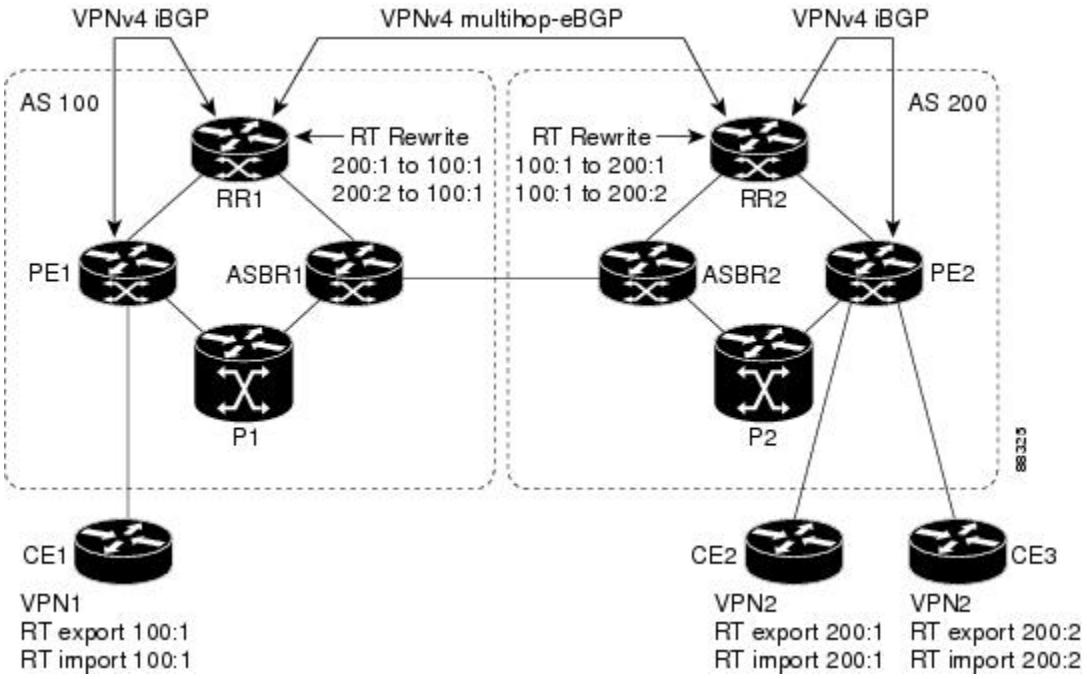
Figure 4 Route Target Replacement on ASBRs in an MPLS VPN Interautonomous System Topology



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:

- EBGP is configured on the route reflectors.
- EBGP and IBGP IPv4 label exchange is configured between all BGP routers.
- Peer groups are configured on the routers 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 5 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 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 an RT replacement policy for your internetwork.

If you configure a PE to rewrite RT *x* to RT *y* and the PE has a VRF 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Command or Action	Purpose
Step 3 **ip extcommunity-list** {*standard-list-number* | *expanded-list-number*} {**permit** | **deny**} [*regular-expression*] [**rt** | **soo** *extended-community-value*] **Example:** `Router(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 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.

Command or Action	Purpose
Step 4 **route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(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 may 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*}
 	
Example:
`Router(config-route-map)# match extcommunity 1`
 	
Example:
`Router(config-route-map)# match extcommunity 101`
	Matches BGP extended community list attributes.
	• 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:** `Router(config-route-map)# set extcomm-list 1 delete`	Removes a route target from an extended community attribute of an inbound or outbound BGP VPNv4 update. • The *extended-community-list-number* argument specifies the extended community list number.

	Command or Action	Purpose
Step 7	**set extcommunity** {**rt** *extended-community-value* [**additive**] | **soo** *extended-community-value*} **Example:** `Router(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:** `Router(config-route-map)# end`	(Optional) Exits to privileged EXEC mode.
Step 9	**show route-map** *map-name* **Example:** `Router# show route-map extmap`	(Optional) Use this command to verify 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, page 87
- Refreshing BGP Session to Apply Route Target Replacement Policy, page 89
- Troubleshooting Tips, page 90

Associating Route Maps with Specific BGP Neighbors

Perform this task to associate 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and places the router in router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 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. • 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 5	**address-family vpnv4** [**unicast**] **Example:** `Router(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 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 172.16.0.2 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **send-community** [**both** | **extended** | **standard**] **Example:** `Router(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:** `Router(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.
Step 9	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Refreshing BGP Session to Apply Route Target Replacement Policy

Perform this task to refresh the BGP session to apply the RT replacement policy.

After you have defined two routers to be BGP neighbors, the routers 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 RT replacement policy and applying it to the target routers 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:** `Router> 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:** `Router# 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 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 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:** `Router# disable`	(Optional) Exits to user EXEC mode.

Troubleshooting Tips

To determine whether a BGP router supports the route refresh capability, use the **show ip bgp neighbors** command. If a router 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 router 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 router sends or receives a large number of BGP updates.

Verifying the Route Target Replacement Policy

Perform this task to verify the operation of your RT replacement policy.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4 all** *network-address*
3. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 **show ip bgp vpnv4 all** *network-address*

Use this command to verify that all VPNv4 prefixes with a specified RT extended community attribute are replaced with the proper RT extended community attribute at the ASBRs or route reflectors and to verify that the PE routers 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:

```
Router# 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:

```
Router# 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:

```
Router# 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:

```
Router# 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**

Use this command to exit to user EXEC mode:

Example:

```
Router# exit
Router>
```

Troubleshooting Your Route Target Replacement Policy

Perform this task to troubleshoot your RT 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**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 **debug ip bgp updates**

Use the following command to verify that BGP updates are occurring on the ASBR. The ASBR in this example has the IP address 172.16.16.16.

Example:

```
Router# 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 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*

Use this command to verify that RT extended community attributes are replaced correctly. For example:

Example:

```
Router# 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 VPN address information from the BGP table and verifies that RT extended community attributes are replaced correctly.

Step 4 **exit**

Use this command to exit to user EXEC mode:

Example:

```
Router# exit
Router>
```

Configuration Examples for MPLS VPN--Route Target Rewrite

Configuring Route Target Replacement Policies Examples

This example shows the RT replacement configuration of an ASBR (ASBR1) that exchanges 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 router 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.

Applying Route Target Replacement Policies Examples

This section contains the following examples:

Associating Route Maps with Specific BGP Neighbor Example

This example shows the association of route map extmap with a BGP neighbor. The BGP inbound route map is configured to replace 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
```

Refreshing the BGP Session to Apply the Route Target Replacement Policy Example

The following example shows the **clear ip bgp** command used to initiate a dynamic reconfiguration in the BGP peer 172.16.0.2. This command requires that the peer supports the route refresh capability.

```
Router# clear ip bgp 172.16.0.2 vpnv4 unicast in
```

Additional References

Related Documents

Related Topic	Document Title
MPLS, MPLS VPN, and MPLS VPN interautonomous systems configuration tasks	*MPLS Layer 3 Inter-AS and CSC Configuration Guide*
Commands to configure MPLS and MPLS VPNs	*Cisco IOS Multiprotocol Label Switching Command Reference*
BGP configuration tasks	*IP Routing Protocols Configuration Guide*
Commands to configure and monitor BGP	*Cisco IOS IP Routing Protocols Command Reference*

Standards

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

MIBs

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

RFCs

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

Technical Assistance

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

Feature Information for MPLS VPN--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 6 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) routers 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 12.0(26)S, this feature was introduced for the Cisco 7200, 7500, and 12000 series routers. In 12.2(25)S, this feature was integrated into a Cisco IOS 12.2S release to support the Cisco 7500 series router. In 12.2(33)SRA, this feature was integrated into a Cisco IOS 12.2SRA release. In 12.2(33)SXH, this feature was integrated into a Cisco IOS 12.2SXH release. In 12.4(20)T, this feature was integrated into a Cisco IOS 12.4T release. 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 router that connects and exchanges information between two or more autonomous systems.

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

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

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

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

IGP --Interior Gateway Protocol. Internet protocol used to exchange routing information within an autonomous system. Examples of common Internet IGPs include 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 routers 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 router that performs label imposition and disposition.

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

MPLS --Multiprotocol Label Switching. A switching method that forwards IP traffic using a label. This label instructs the routers 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 router --provider router. The core router in the service provider network that connects to provider edge (PE) routers. In a packet-switched star topology, a router that is part of the backbone and that serves as the single pipe through which all traffic from peripheral networks must pass on its way to other peripheral networks.

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

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 router 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) router.

MPLS VPN--Show Running VRF

The MPLS VPN--Show Running VRF feature provides a Cisco IOS command-line interface (CLI) option to display a subset of the running configuration on a router 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 router.

On heavily loaded routers, 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 router 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 router.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 at the end of this document.

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 IOS image that supports VRFs installed on the router
- At least one VRF configured on the router
- Cisco Express Forwarding for MPLS VPN routing and forwarding

Restrictions for MPLS VPN--Show Running VRF

Any element of the running configuration of the router that is not linked directly to a VRF 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 (for more information, see the Display of Configuration Not Directly Linked to a VRF, page 103).

Information About MPLS VPN--Show Running VRF

Configuration Elements Displayed for the MPLS VPN--Show Running VRF Feature

You can display the running configuration associated with a specific VRF or all VRFs on the router 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:

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

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

- 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 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 VRF 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.

How to Configure MPLS VPN--Show Running VRF

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

Configuration Examples for MPLS VPN--Show Running VRF

Additional References

Related Documents

Related Topic	Document Title
MPLS command descriptions	*Cisco IOS Multiprotocol Label Switching Command Reference*

Standards

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

MIBs

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

RFCs

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

Technical Assistance

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

Feature Information for MPLS VPN--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	The MPLS VPN--Show Running VRF feature provides a CLI option to display a subset of the running configuration on a router that is linked to a VRF. You can display the configuration of a specific VRF or of all VRFs configured on a router. 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 router. 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.
		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 routers 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 router configuration that enables IP traffic to use tunneling to travel securely over a public TCP/IP network.

MPLS VPN--VRF CLI 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.

The MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs feature introduces Cisco IOS command-line interface (CLI) 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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 at the end of this document.

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) VPN VRF 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 routers in the core, including the provider edge (PE) routers.

Restrictions for MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs

- Once you have converted to a multiprotocol VRF, 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

VPNs for IPv6 use the same VRF concepts that IPv4 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 Cisco IOS Release 12.2(33)SRB, 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.

Cisco IOS Release 12.2(33)SRB 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 Cisco IOS Release 12.2(33)SRB, you could create a single-protocol IPv4-only VRF. 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 in Cisco IOS Release 12.2(33)SRB, 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 router. 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:

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

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

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

If you try to edit an IPv4-only VRF with the multiprotocol VRF commands, you would 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) *name* 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 Configurations Characteristics

In a multiprotocol VRF, 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 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 Cisco IOS Release 12.2(33)SRB. This means that you might have single-protocol and multiprotocol CLI on the same router, 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

This feature provides Cisco IOS CLI commands that allow you to configure a multiprotocol VRF (IPv4 and IPv6 VPNs in the same VRF) and to migrate a single-protocol VRF configuration (IPv4-only VRF) 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.

- Configuring a VRF for IPv4 and IPv6 MPLS VPNs, page 112
- Associating a Multiprotocol VRF with an Interface, page 114
- Verifying the MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs Configuration, page 116
- Migrating from a Single-Protocol IPv4-Only VRF to a Multiprotocol VRF Configuration, page 119

Configuring a VRF for IPv4 and IPv6 MPLS VPNs

Perform the following task to configure a VRF for IPv4 and IPv6 MPLS 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:** `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	**vrf definition** *vrf-name* **Example:** `Router(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:** `Router(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:** `Router(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:** `Router(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.

	Command or Action	Purpose
Step 7	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits from VRF address family configuration mode.
Step 8	**address-family** {**ipv4** | **ipv6**} **Example:** `Router(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:** `Router(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:** `Router(config-vrf-af)# end`	Exits to privileged EXEC mode.

Associating a Multiprotocol VRF with an Interface

Perform the following task to associate a multiprotocol VRF 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-addressmask* [**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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(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:** `Router(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-addressmask* [**secondary**] **Example:** `Router(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.

	Command or Action	Purpose
Step 6	**ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*} **Example:** `Router(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:** `Router(config-if) end`	Exits 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 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**

Use this command to enable privileged EXEC mode. Enter your password, if prompted. For example:

Example:

```
Router> enable
Router#
```

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

Use this command to verify 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:

```
Router# 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:

```
Router# 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**

Use this command to verify 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:

```
Router# 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:

```
Router# 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:

```
Router# 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*]

Use this command to display 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 would display the following:

Example:

```
Router# 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**

Use this command to exit to user EXEC mode. For example:

Example:

```
Router# exit
Router>
```

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 VRF 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**vrf upgrade-cli multi-af-mode** {**common-policies** | **non-common-policies**} [**vrf** *vrf-name*] **Example:** `Router(config)# vrf upgrade-cli multi-af-mode common-policies vrf vpn4`	Upgrades a VRF instance or all VRFs configured on the router 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:** `Router(config)# exit`	Exits to privileged EXEC mode.
Step 5	**show running-config vrf** [*vrf-name*] **Example:** `Router# show running-config vrf vpn4`	Displays the subset of the running configuration of a router that is linked to a specific VRF instance or to all VRFs configured on the router. • The *vrf-name* argument is the name of the VRF of which you want to display the configuration. **Note** The Cisco IOS 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 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 VRF configuration for IPv4 and IPv6 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 VRF configuration for IPv4 and IPv6 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 VRF 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 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
```

Example Configuring a VRF for IPv4 and IPv6 VPNs

The following example shows how to configure a VRF for IPv4 and IPv6 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 VRF 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
```

Example 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 VRF to a multiprotocol VRF configuration.

This example shows a single-protocol IPv4-only VRF before the Cisco IOS VRF CLI for IPv4 and IPv6 is entered on the router:

```
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:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
!
Router(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
Router(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
MPLS	MPLS Product Literature
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Commands for configuring MPLS and MPLS VPNs	*Cisco IOS Multiprotocol Label Switching Command Reference*

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 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 8 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)SRB 12.2(33)SXI	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 Cisco IOS command-line interface (CLI) commands that allow you to enable an IPv4 and IPv6 VPN in the same Multiprotocol Label Switching (MPLS) VRF instance and to simplify the migration from a single-protocol VRF configuration to a multiprotocol VRF configuration. In 12.2(33)SRB, this feature was introduced on the Cisco 7600 router. In 12.2(33)SXI, this feature was integrated into a Cisco IOS 12.2SXI release.
		The following commands were introduced or modified: **show vrf**, **vrf definition**, **vrf forwarding**, **vrf upgrade-cli**.

Glossary

6PE --IPv6 provider edge router or a Multiprotocol Label Switching (MPLS) label switch router (LSR) edge router using IPv6.

6VPE --IPv6 Virtual Private Network (VPN) provider edge router.

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 router. A service provider router 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 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.

PE --provider edge router. A router that is part of a service provider’s network and that is connected to a customer edge (CE) router. The PE router 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 router 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 router.

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) router to maintain independent routing states for each customer.

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 feature information and caveats, see 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 at the end of this document.

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 VRF Selection Using Policy-Based Routing

The router must support PBR to configure this feature. For platforms that do not support PBR, use the "VRF Selection Based on Source IP Address" feature introduced in Cisco IOS Release 12.0(22)S.

A VRF must be defined prior to the configuration of this feature. An error message is displayed on the console if no VRF exists.

This document assumes that multiprotocol BGP (mBGP), Multiprotocol Label Switching (MPLS), and Cisco Express Forwarding are enabled in your network.

Restrictions for VRF Selection Using Policy-Based Routing

The VRF Selection Using Policy-Based Routing feature is supported only in service provider (-p-) images.

The VRF Selection Using Policy-Based Routing feature can coexist with the VRF Selection Based on Source IP address feature on the same router, but these features cannot be configured together on the same interface. This is designed behavior to prevent 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 PBR and cannot be configured for a source IP address that is a match criterion for this feature.

The VRF Selection Using Policy-Based Routing feature cannot be configured with IP prefix lists.

Information About VRF Selection Using Policy-Based Routing

Introduction to VRF Selection Using Policy-Based Routing

The VRF Selection Using Policy-Based Routing feature is an extension of the VRF Selection Based on Source IP Address feature. The PBR implementation of the VRF selection feature allows you to policy route 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 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 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 VRF after the successful match occurs in the route map.

How to Configure VRF Selection Using Policy-Based Routing

Defining the Match Criteria for PBR VRF Selection Based on Packet Length

The match criteria for PBR 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.

Prerequisites

Before you perform this task, make sure that the VRF and associated IP address are already defined.

Configuring PBR VRF Selection with a Standard Access List

Use the following commands to create a standard access list and define the PBR 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:** `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	**access-list** *access-list-number* {**deny** | **permit**} *source-addr* [*source-wildcard*] [**log**] **Example:** `Router(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 PBR VRF Selection with a Named Access List

Use the following commands to define the PBR VRF route selection match criteria in a named access list in order to permit or deny the transmission of 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Command or Action	Purpose
Step 3 **ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] **Example:** `Router(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:** `Router(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 PBR VRF Selection in a Route Map

Use the following commands to configure the VRF through which the outbound 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.

- The VRF 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:** `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	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(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:
	`Router(config-route-map)# match ip address 1`
	
	Example:
	`Router(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:** `Router(config-route-map)# set vrf map1`	Defines which VRF to route VPN packets that are successfully matched in the same route map sequence for PBR VRF selection. • The example policy routes matched packets out to the VRF named map1.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.

Configuring PBR on the Interface

Use the following commands to filter incoming 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 VRF Selection Using Policy-Based Routing feature can coexist with the VRF Selection Based on Source IP address feature on the same router, but the two features cannot be configured together on the same interface. This is designed behavior to prevent 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:** `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	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface FastEthernet 0/1`	Configures an interface and enters interface configuration mode.
Step 4	**ip policy route-map** *map-tag* **Example:** `Router(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:** `Router(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:** `Router(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 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface FastEthernet 0/1`	Configures an interface and enters interface configuration mode.
Step 4	**ip policy route-map** *map-tag* **Example:** `Router(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:** `Router(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.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode, and enters privileged EXEC mode.

Verifying the Configuration of the VRF Selection Using Policy-Based Routing

To verify the configuration of the VRF Selection Using Policy-Based Routing feature, perform each of the following steps in this section in the order specified.

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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip access-list** [*access-list-number*	*access-list-name*] **Example:** `Router# show ip access-list`	Displays the contents of all current IP access lists. • 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:** `Router# 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.

	Command or Action	Purpose
Step 4	**show ip policy** **Example:** `Router# 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 VRF Selection Using Policy-Based Routing

Example Defining PBR VRF Selection in 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 PBR-VRF-Selection route map to the 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 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
```

Example Verifying VRF Selection Using Policy-Based Routing

The following verification examples show defined match criteria and route-map policy configuration.

Verifying Match Criteria

To verify the configuration of match criteria for 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:

```
Router# 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
```

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.

```
Router# 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
```

Verifying PBR VRF Selection Policy

The following **show ip policy** command output displays the interface and associated route map that is configured for policy routing:

```
Router# show ip policy
Interface               Route map
FastEthernet0/1/0       PBR-VRF-Selection
```

Additional References

Related Documents

Related Topic	Document Title
VRF selection based on the source IP address	Directing MPLS VPN Traffic Using a Source IP Address

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

Technical Assistance

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

Feature Information for 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 9 Feature Information for 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	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 12.3(7)T, this feature was introduced. This feature was integrated into Cisco IOS Release 12.2(25)S. This feature was integrated into Cisco IOS Release 12.2(33)SRB. This feature was integrated into Cisco IOS Release 12.2(33)SXI. The following commands were introduced or modified: **ip vrf receive**, **set vrf**.

Glossary

PBR --policy-based routing.

VPN --Virtual Private Network.

VRF --virtual routing and forwarding.