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

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.

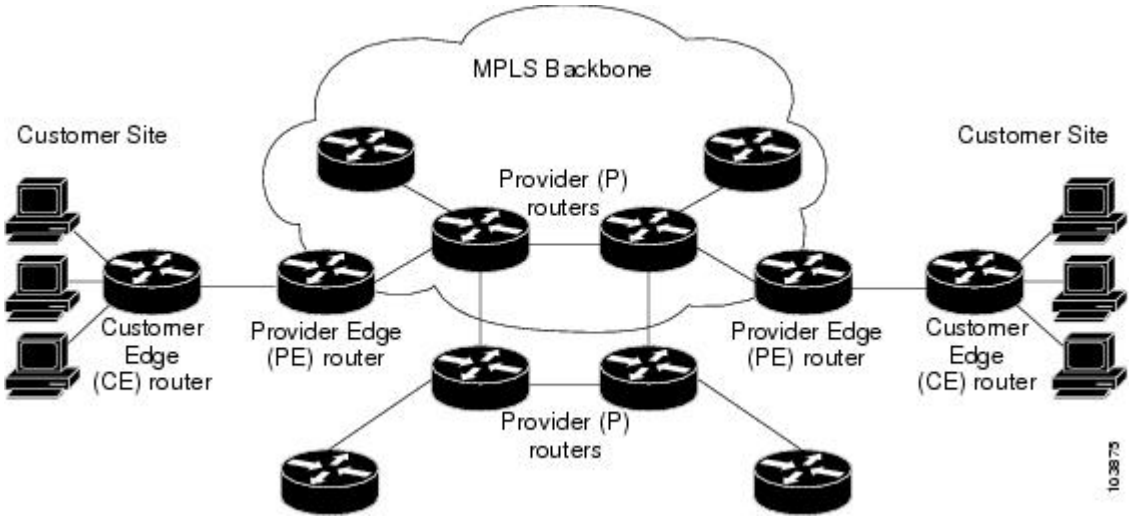
MPLS VPNs are easier to manage and expand than conventional VPNs. When a new site is added to an MPLS VPN, only the service provider’s edge router that provides services to the customer site needs to be updated.

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.

MPLS VPN Half-Duplex VRF

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

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest 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 Configuring MPLS VPN Half-Duplex VRF

You must have a working MPLS core network.

Restrictions for MPLS VPN Half-Duplex VRF

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

- Multicast

- MPLS VPN Carrier Supporting Carrier
- MPLS VPN Interautonomous Systems

Information About Configuring MPLS VPN Half-Duplex VRF

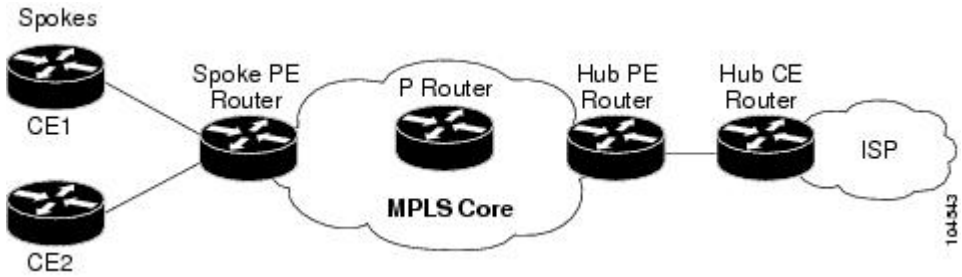
MPLS VPN Half-Duplex VRF Overview

The MPLS VPN Half-Duplex VRF feature provides:

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

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

Figure 2 Hub-and-Spoke Topology



Upstream and Downstream VRFs

The MPLS VPN Half-Duplex VRF feature uses two unidirectional VRFs to forward IP traffic between the spokes and the hub PE router:

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

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

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

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

Reverse Path Forwarding Check

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

Unicast RPF is not on by default. You need to enable it, as described in Configuring Unicast Reverse Path Forwarding .

How to Configure MPLS VPN Half-Duplex VRF

Configuring the Upstream and Downstream VRFs on the Spoke PE Router

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `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.

	Command or Action	Purpose
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. **Note** The MPLS VPN Half Duplex VRF feature supports only the IPv4 address family.
Step 6	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `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.
Step 7	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode.
Step 8	**end** **Example:** `Router(config-vrf)# end`	Exits to privileged EXEC mode.

Associating a VRF with an Interface

Perform the following task to associate a VRF with an interface, which activates the VRF.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `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* [**downstream** *vrf-name2* **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. • The **downstream** *vrf-name2* keyword and argument combination is the name of the downstream VRF into which peer and per-user routes are installed.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `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.
Step 6	**end** **Example:** `Router(config-if) end`	Exits to privileged EXEC mode.

Configuring the Downstream VRF for an AAA Server

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

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

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

ip:vrf-id=U downstream D

Verifying MPLS VPN Half-Duplex VRF Configuration

To verify the Downstream VRF for an AAA Server configuration, perform the following steps.

SUMMARY STEPS

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

DETAILED STEPS

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

Use this command to display information about all of the VRFs configured on the router, including the downstream VRF for each associated interface or VAI:

Example:

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

show vrf detail *vrf-name*

Use this command to display detailed information about the VRF you specify, including all interfaces, subinterfaces, and VAIs associated with the VRF.

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

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

Example:

```
Router# show vrf detail vrf1
VRF D; default RD 2:0; default VPNID <not set>
  Interfaces:
         Loopback2           Virtual-Access3 [D]  Virtual-Access4 [D]
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:2:0
  Import VPN route-target communities
    RT:2:1
```

```
  No import route-map
  No export route-map
VRF U; default RD 2:1; default VPNID <not set>
  Interfaces:
    Virtual-Access3          Virtual-Access4
  Connected addresses are not in global routing table
  No Export VPN route-target communities
  Import VPN route-target communities
    RT:2:1
  No import route-map
  No export route-map
```

The following example shows the VRF detail in a standard VPN situation:

Example:

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

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

Use this command to display the IP routing table for the VRF you specify, and information about the per-user routes installed in the downstream VRF.

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

Example:

```
Router# show ip route vrf D

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

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

Example:

```
Router# show ip route vrf Down

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

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

Example:

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

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

Example:

```
Router# show ip route vrf Up
Routing Table: Up
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is 10.13.13.13 to network 0.0.0.0
```

```
    10.2.0.0/32 is subnetted, 1 subnets
C          10.2.0.1 is directly connected, Pos3/0/3
     10.3.0.0/32 is subnetted, 1 subnets
B       10.3.16.16 [200/0] via 10.13.13.13, 1w3d
B     10.6.0.0/8 [200/0] via 10.13.13.13, 1w3d
    10.0.0.0/32 is subnetted, 1 subnets
C          10.0.0.1 is directly connected, Pos3/0/1
B*   0.0.0.0/0 [200/0] via 10.13.13.13, 1w3d
```

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

Use this command to display information about the interfaceyou specify, including information about the associated upstream and downstream VRFs.

The following example shows how to display information about the subinterface named POS3/0/1:

Example:

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

Configuration Examples for MPLS VPN Half-Duplex VRF

Example Configuring the Upstream and Downstream VRFs on the Spoke PE Router

The following example configures an upstream VRF named Up:

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

The following example configures a downstream VRF named Down:

```
Router> enable
Router# configure terminal
Router(config)# vrf definition Down
```

```
Router(config-vrf)# rd 1:8
Router(config-vrf)# address-family ipv4
Router(config-vrf-af)# route-target import 1:8
Router(config-vrf-af)# exit-address-family
```

Example Associating a VRF with an Interface

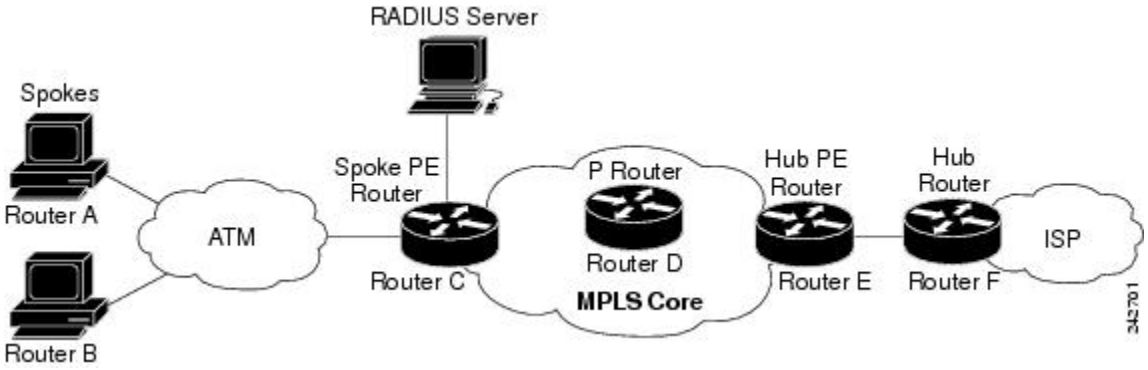
The following example associates the VRF named Up with POS 3/0/1 subinterface and specifies the downstream VRF named Down:

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

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

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

Figure 3 Sample Topology




```
vrf definition D
 rd 1:8
 address-family ipv4
 route-target export 1:100
 exit-address-family
!
vrf definition U
 rd 1:0
 address-family ipv4
 route-target import 1:0
 exit-address-family
!
ip cef
vpdn enable
!
vpdn-group U
 accept-dialin
  protocol pppoe
  virtual-template 1
!
interface Loopback 2
 vrf forwarding U
```

```
 ip address 10.0.0.8 255.255.255.255
!
interface ATM 2/0
 description Mze ATM3/1/2
 no ip address
 no atm ilmi-keepalive
 pvc 0/16 ilmi
!
 pvc 3/100
  protocol pppoe
!
pvc 3/101
  protocol pppoe
!
```

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

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

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

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

```
aaa new-model
!
aaa group server radius R
 server 10.0.20.26 auth-port 1812 acct-port 1813
!
aaa authentication ppp default group radius
aaa authorization network default group radius
!
vrf defintion D
 description Downstream VRF - to spokes
 rd 1:8
 address-family ipv4
 route-target export 1:100
 exit-address-family
!
vrf definition U
 description Upstream VRF - to hub
 rd 1:0
 address-family ipv4
 route-target import 1:0
 exit-address-family
!
ip cef
vpdn enable
!
vpdn-group U
 accept-dialin
  protocol pppoe
  virtual-template 1
!
interface Loopback2
 vrf forwarding U
 ip address 10.0.0.8 255.255.255.255
!
interface ATM2/0
```

```
  pvc 3/100
  protocol pppoe
 !
pvc 3/101
  protocol pppoe
 !
interface virtual-template 1
 no ip address
 ppp authentication chap
!
router bgp 1
 no synchronization
 neighbor 172.16.0.34 remote-as 1
 neighbor 172.16.0.34 update-source Loopback0
 no auto-summary
 !
address-family vpnv4
  neighbor 172.16.0.34 activate
  neighbor 172.16.0.34 send-community extended
  auto-summary
  exit-address-family
 !
address-family ipv4 vrf U
  no auto-summary
  no synchronization
  exit-address-family
!
address-family ipv4 vrf D
  redistribute static
  no auto-summary
  no synchronization
  exit-address-family
!
ip local pool U-pool 10.8.1.1 2.8.1.100
ip route vrf D 10.0.0.0 255.0.0.0 Null0
!
radius-server host 10.0.20.26 auth-port 1812 acct-port 1813
radius-server key cisco
```

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

The following example shows how to use OSPF to dynamically advertise the routes on the spoke sites.

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

Creating the VRFs

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

Enabling MPLS

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

Configuring BGP Toward Core

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

Configuring BGP Toward Edge

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

Spoke PE's Core-Facing Interfaces and Processes

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

Spoke PE's Edge-Facing Interfaces and Processes

```
interface Loopback 100
vrf forwarding Down
 ip address 10.22.22.22 255.255.255.255
!
interface POS 3/0/1
vrf forwarding Up downstream Down
 ip address 10.0.0.1 255.0.0.0
!
interface POS 3/0/3
vrf forwarding Up downstream Down
 ip address 10.2.0.1 255.0.0.0
!
router ospf 1000 vrf Down
 router-id 10.22.22.22
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 nsf enforce global
 redistribute connected subnets
```

```
redistribute bgp 100 metric-type 1 subnets
network 10.22.22.22 0.0.0.0 area 300
network 10.0.0.0 0.255.255.255 area 300
network 10.2.0.0 0.255.255.255 area 300
default-information originate
```

Additional References

Related Documents

Related Topic	Document Title
MPLS VPNs	Configuring MPLS Layer 3 VPNs
MPLS commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
Configuring IPv4 and IPv6 VRFs	MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs
Unicast Reverse Path Forwarding	Configuring Unicast Reverse Path Forwarding

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 VPN Half-Duplex VRF

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

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

Table 2 Feature Information for MPLS VPN Half-Duplex VRF

Feature Name	Releases	Feature Information
MPLS VPN - Half Duplex VRF (HDVRF) Support with Static Routing	12.3(6) 12.3(11)T 12.2(28)SB	This feature ensures that VPN clients that connect to the same PE router at the edge of the MPLS VPN use the hub site to communicate. In 12.3(6), this feature was introduced. In 12.4(20)T, this feature was integrated. In 12.2(28)SB, this feature was integrated

Feature Name	Releases	Feature Information
MPLS VPN Half-Duplex VRF	12.2(28)SB2 12.4(20)T 12.2(33)SRC	In 12.2(28)SB2, support for dynamic routing protocols was added. For the Cisco 10000 series routers, see the "Half-Duplex VRF" section of the "Configuring Multiprotocol Label Switching" chapter in the Cisco 10000 Series Router Broadband Aggregation, Leased-Line, and MPLS Configuration Guide at the following URL: http://www.cisco.com/univercd/cc/td/doc/product/aggr/10000/swconfig/cfggdes/bba/dffsrv.htm#wp1065648 In 12.4(20)T, this feature, with support for dynamic routing protocols, was integrated. In Cisco IOS Release 12.2(33)SRC this feature, with support for dynamic routing protocols, was integrated into the SR train. The following commands were introduced or modified: **show ip interface**, **show vrf**

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 59).

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 3 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 68
- Associating a Multiprotocol VRF with an Interface, page 70
- Verifying the MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs Configuration, page 72
- Migrating from a Single-Protocol IPv4-Only VRF to a Multiprotocol VRF Configuration, page 75

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 4 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--BGP Local Convergence

This document provides information about reducing the downtime of a provider edge (PE) to customer edge (CE) link failure. It describes how to reroute PE-egress traffic onto a backup path to the CE before BGP has reconverged. The MPLS VPN--BGP Local Convergence feature is also referred to as "local protection."

This document explains how to use PE-CE local convergence. For information on using BGP PIC Edge for BGP local convergence support, see BGP PIC Edge for IP and MPLS-VPN.

Note The MPLS VPN--BGP Local Convergence feature affects only traffic exiting the Virtual Private Network (VPN). Therefore, it cannot fully protect traffic end-to-end by itself.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest 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--BGP Local Convergence

- Before MPLS VPN --BGP Local Convergence link protection can be enabled, the customer site must be connected to the provider site by more than one path.

- Both the main forwarding path and the redundant backup path must have been installed within Border Gateway Protocol (BGP), and BGP must support lossless switchover between operational paths.
- Any of the supported routing protocols can be used between the PE and CE as long as the path is redistributed into BGP. The supported protocols for IPv4 are External BGP (eBGP), Routing Information Protocol (RIP), Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), and static routing.The supported protocols for IPv6 are External BGP (eBGP) and static routing.
- All PE routers that are serving as backup to the link must have assigned a unique Route Distinguisher to each VRF table involved with the link to ensure that the route reflectors advertise all available paths.
- Although not required, it is recommended that the backup PE (shown as “PE2” in the figure below) runs the same Cisco IOS release that is running on the PE (“PE1”) whose link with the CE will be protected; that is, Cisco IOS Release 12.2(33) SRC, 12.2(33)SB, Cisco IOS 15.0(1)M, Cisco IOS 15.0(1)S, or a more recent version of those products.

Restrictions for MPLS VPN--BGP Local Convergence

- The MPLS VPN--BGP Local Convergence feature affects only traffic exiting the VPN. Therefore, it cannot fully protect traffic end-to-end by itself.
- This link protection cannot be initiated *during* a high availability (HA) stateful switchover (SSO). But links already configured with this protection *before* the switchover begins will remain protected after the switchover.
- If you perform an in-service software downgrade from an image that does include this link protection to an image that does not support this feature, active protection will be halted when BGP routes are refreshed.
- Any next-hop core tunneling technology that is supported by BGP is also supported for protection, including Multiprotocol Label Switching (MPLS), IP/Layer 2 Tunneling Protocol version 3 (L2TPv3), and IP/generic routing encapsulation (GRE). Enabling a Carrier Supporting Carrier (CsC) protocol between the PE and CE is also supported. Interautonomous system option A (back-to-back virtual routing and forwarding (VRF)) is supported because it is essentially the same as performing the PE-CE link protection in both autonomous systems. However, interautonomous system options B and C protection are not supported.
- The MPLS VPN--BGP Local Convergence feature for IPv4 supports the eBGP, RIP, EIGRP, OSPF, and static routing protocols only.
- The MPLS VPN--BGP Local Convergence feature for IPv6 supports the eBGP and static routing protocols only.

Information About MPLS VPN--BGP Local Convergence

- How Link Failures Are Handled with BGP, page 87
- How Links Are Handled with the MPLS VPN--BGP Local Convergence Feature, page 87
- How Link Failures Are Detected, page 88

How Link Failures Are Handled with BGP

Within a Layer 3 VPN network, the failure of a PE-CE link can cause a loss of connectivity (LoC) to a customer site, which is detrimental to time-sensitive applications. Several factors contribute to the duration of such an outage:

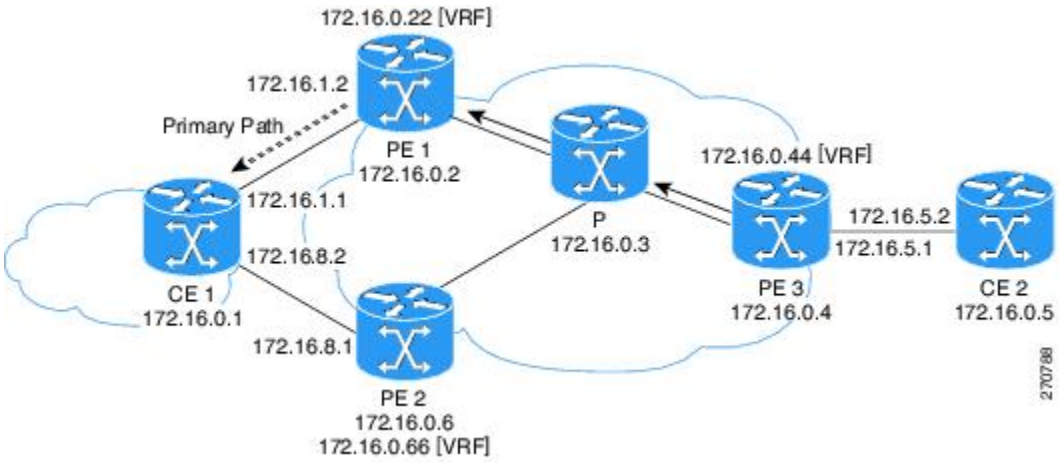
- The time to detect the failure
- The programming of the forwarding
- The convergence of BGP (in large networks, the restored traffic arrival time at its destination varies according to the prefix)

When BGP detects a PE-CE link failure, it removes all of the BGP paths through the failing link. BGP runs the best-path algorithm on the affected prefixes and selects alternate paths for each prefix. These new paths (which typically include a remote PE) are installed into forwarding. The local labels are removed and BGP withdrawals are sent to all BGP neighbors. As each BGP neighbor receives the withdrawal messages (typically indirectly using routereflectors), the best-path algorithm is called and the prefixes are switched to an alternate path. Only then is connectivity restored.

How Links Are Handled with the MPLS VPN--BGP Local Convergence Feature

The MPLS VPN--BGP Local Convergence feature requires that the prefixes to be protected on a PE-CE link have at least one backup path that does not include that link. (See the figure below.) The customer site must have backup paths to the provider site.

Figure 4 Network Configured with Primary and Backup Paths

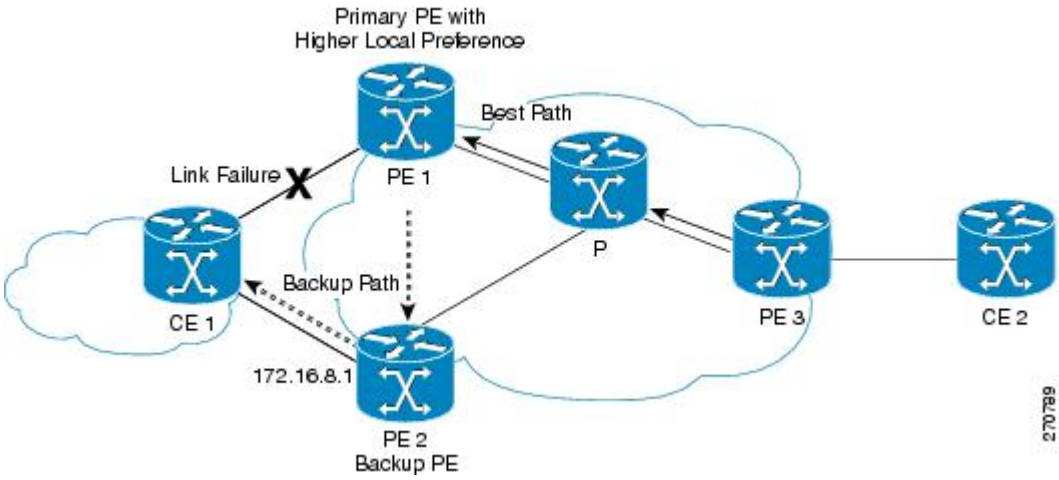


The MPLS VPN--BGP Local Convergence feature reduces LoC time by sending the broken link's traffic over a backup path (as shown in the figure below) instead of waiting for total network convergence. The local label is maintained for 5 minutes while prefixes switch from the failing local path to the backup path. Because the label is not freed as had been the usual practice, forwarding continues to take place.

The best-path algorithm selects the backup path. Thus, the local label has been applied in place of the failed BGP best-path label (which is sometimes called "label swapping"). Traffic is restored locally while the

network propagation of the BGP withdrawal messages takes place. Eventually, the egress PE router converges and bypasses the local repair.

Figure 5 Network Using the Backup Path After a PE-CE Link Failure on the Primary Path



Note After the 5-minute label preservation, the local labels are freed. Any BGP prefix that is remote and is not part of a CsC network does not have a local label and is removed. The delay in local label deletion does not modify normal BGP addition and deletion of BGP paths. Rather, BGP reprograms the new backup bestpath into forwarding as usual.

How Link Failures Are Detected

Local protection relies on BGP being notified of the interface failure. Detection can occur using either the interface drivers or the routing tables. If an interface or route goes down, the corresponding path in the routing table is removed and BGP will be notified using the routing application programming interfaces (APIs).

However, when the routing table cannot detect the failure (as when a Layer 2 switch goes down), BGP determines that a neighbor is down through use of its hold-down timer. However, that determination can be extremely slow because of the 3-minute default for BGP session timeout.

You can reduce the detection delay by either reducing the BGP session timeout interval (as described in the Configuring Internal BGP Features document) or by enabling the Bidirectional Forwarding Detection (BFD) protocol within eBGP between the PE and CE. For complete instructions to enable BFD, see the Bidirectional Forwarding Detection document.

How to Enable MPLS VPN--BGP Local Convergence

Note To configure a VPN routing and forwarding (VRF) instance for IPv4 and IPv6 VPNs or to upgrade your existing single-protocol IPv4-only VRF to a multiprotocol VRF configuration, see MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs.

Configuring MPLS VPN--BGP Local Convergence with IPv4

Ensure that the CE is already connected to the PE by a minimum of two paths.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **protection local-prefixes**
6. **do show ip vrf detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `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`	Enters VRF configuration mode. • If no VRF routing table and Cisco Express Forwarding table had been previously created for this named VRF, then this command also creates them, giving both tables the specified value for the *vrf-name* argument (in this example, the name is vpn1).

	Command or Action	Purpose
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 100:3`	(Optional) Establishes the route distinguisher for the named VRF. • If no route distinguisher had been previously established for the named VRF, then you must enter this command. • The route distinguisher value can be either an: ◦ Autonomous system number followed by a colon and an arbitrary number (for example, 100:3) or • ◦ IP address followed by a colon and an arbitrary number (for example, 192.168.122.15:1)
Step 5	**protection local-prefixes** **Example:** `Router(config-vrf)# protection local-prefixes`	Allows a preconfigured backup path to carry traffic if the PE-CE link breaks by preserving the local prefixes while BGP reconverges.
Step 6	**do show ip vrf detail** **Example:** `Router(config-vrf)# do show ip vrf detail`	(Optional) Verifies that the MPLS VPN--BGP Local Convergence feature has been configured.

Configuring MPLS VPN--BGP Local Convergence with IPv6

Ensure that the CE is already connected to the PE by a minimum of two paths.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **address-family** [**ipv4** | **ipv6**]
6. **protection local-prefixes**
7. **do show ip vrf detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **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 vrf2`	Enters VRF configuration mode. • If no VRF routing table and Cisco Express Forwarding table had been previously created for this named VRF, then this command also creates them, giving both tables the specified value for the *vrf-name* argument (in this example, the name is vrf2).
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 100:3`	(Optional) Establishes the route distinguisher for the named VRF. • If no route distinguisher had been previously established for the named VRF, then you must enter this command. • The route distinguisher value can be either an: ◦ Autonomous system number followed by a colon and an arbitrary number (for example, 100:3) or • ◦ IP address followed by a colon and an arbitrary number (for example, 192.168.122.15:1)
Step 5	**address-family** [**ipv4** | **ipv6**] **Example:** `Router(config-vrf)# address-family ipv6`	Enters VRF address family configuration mode and specifies the IPv4 or IPv6 protocol.
Step 6	**protection local-prefixes** **Example:** `Router(config-vrf-af)# protection local-prefixes`	Allows a preconfigured backup path to carry traffic if the PE-CE link breaks by preserving the local prefixes while BGP reconverges.

Command or Action	Purpose
Step 7 **do show ip vrf detail** **Example:** `Router(config-vrf-af)# do show ip vrf detail`	(Optional) Verifies that the MPLS VPN to BGP Local Convergence feature has been configured.

- Examples, page 92

Examples

To verify that local link protection has been enabled, enter the **show ip vrf detail** command. If the protection is enabled, the status message “Local prefix protection enabled” will be shown in the display:

```
Router# show ip vrf detail

VRF vpn1 (VRF Id = 1); default RD 100:1; default VPNID <not set>
Interfaces:
    AT1/0/1.1
VRF Table ID = 1
  Export VPN route-target communities
    RT:100:1
  Import VPN route-target communities
    RT:100:1                RT:100:2
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
     Local prefix protection enabled
```

Troubleshooting Tips

- Ensure that a minimum of two paths are present for the protected prefix in BGP in steady state condition on the PE. The path using the protected PE should be the BGP best-path before failover occurs. To display the configuration, enter the **show ip bgp vpnv4 vrf** *vpn ip-prefix* command.
- Ensure that local protection has been enabled in the protected PE by entering the **show ip vrf detail** command, as shown in the Examples, page 92.
- When route reflectors exist in the topology, ensure that each VRF has a unique route distinguisher.

Configuration Examples for MPLS VPN--BGP Local Convergence

- Example MPLS VPN--BGP Local Convergence, page 93
- Example MPLS VPN--BGP Local Convergence for 6VPE 6PE, page 95

Example MPLS VPN--BGP Local Convergence

The following examples show how MPLS VPN--BGP local convergence can prevent traffic loss after a link failure. You can display a detailed view of local link protection before, during, and after BGP convergence by using the **show bgp vpnv4** and **show mpls forwarding-table vrf** commands as shown in the following three-stage example.

Note The **show bgp vpnv4 unicast** command is equivalent to the **show ip bgp vpnv4** command.

Example 1: Before the Link Failure

Both a primary path and a backup path have been configured:

```
Router# show bgp vpnv4 unicast all 172.16.0.1
BGP routing table entry for 100:1:172.16.0.1/32, version 2
Paths: (2 available, best #2, table v1)
Flag: 0x820
  Advertised to update-groups:
     1
  100, imported path from 100:2:172.16.0.1/32
    172.16.0.6 (metric 21) from 172.16.0.7 (172.16.0.7)
      Origin incomplete, metric 0, localpref 100, valid, internal
      Extended Community: RT:100:0
      Originator: 172.16.0.6, Cluster list: 172.16.0.7
      mpls labels in/out 16/17
  100
    172.16.1.1 from 172.16.1.1 (172.16.0.1)
      Origin incomplete, metric 0, localpref 100, valid, external, best
      Extended Community: RT:100:0
      mpls labels in/out 16/nolabel
BGP routing table entry for 100:2:172.16.0.1/32, version 9
Paths: (1 available, best #1, no table)
Flag: 0x820
  Not advertised to any peer
  100
    172.16.0.6 (metric 21) from 172.16.0.7 (172.16.0.7)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:0
      Originator: 172.16.0.6, Cluster list: 172.16.0.7
      mpls labels in/out nolabel/17
```

Label information for both paths can be displayed:

```
Router# show bgp vpnv4 unicast all labels
Network          Next Hop      In label/Out label
Route Distinguisher: 100:1 (v1)
   172.16.0.1/32    172.16.0.6      16/17
                    172.16.1.1      16/nolabel
   172.16.0.5/32    172.16.0.4      nolabel/23
   172.16.0.22/32   0.0.0.0         17/nolabel(v1)
   172.16.0.44/32   172.16.0.4      nolabel/24
   172.16.0.66/32   172.16.0.6      nolabel/21
   172.16.1.0/24    172.16.1.1      18/nolabel
                    0.0.0.0         18/nolabel(v1)
   172.16.5.0/24    172.16.0.4      nolabel/25
   172.16.8.0/24    172.16.0.6      19/23
                    172.16.1.1      19/nolabel
Route Distinguisher: 100:2
   172.16.0.1/32    172.16.0.6      nolabel/17
   172.16.0.66/32   172.16.0.6      nolabel/21
   172.16.8.0/24    172.16.0.6      nolabel/23
```

The PE1 (see the first figure above) forwarding table contains BGP best-path information:

```
Router# show mpls forwarding-table vrf v1 172.16.0.1 detail
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
16         No Label   172.16.0.1/32[V] 570           Et0/0      172.16.1.1
        MAC/Encaps=14/14, MRU=1504, Label Stack{}
        AABBCC000B00AABBCC000C000800
        VPN route: v1
        No output feature configured
```

Example 2: After the Link Failure and Before BGP Convergence

After the link failure on only one path, the backup path remains available (see the second figure above):

```
Router# show bgp vpnv4 unicast all 172.16.0.1

BGP routing table entry for 100:1:172.16.0.1/32, version 19
Paths: (1 available, best #1, table v1)
  Not advertised to any peer
  100, imported path from 100:2:172.16.0.1/32
    172.16.0.6 (metric 21) from 172.16.0.7 (172.16.0.7)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:0
      Originator: 172.16.0.6, Cluster list: 172.16.0.7
      mpls labels in/out 16/17
BGP routing table entry for 100:2:172.16.0.1/32, version 9
Paths: (1 available, best #1, no table)
  Not advertised to any peer
  100
    172.16.0.6 (metric 21) from 172.16.0.7 (172.16.0.7)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:0
      Originator: 172.16.0.6, Cluster list: 172.16.0.7
      mpls labels in/out nolabel/17
```

The label information for the backup path label can be displayed:

```
Router# show bgp vpnv4 unicast all labels

Network          Next Hop      In label/Out label
Route Distinguisher: 100:1 (v1)
   172.16.0.1/32    172.16.0.6      16/17
   172.16.0.5/32    172.16.0.4      nolabel/23
   172.16.0.22/32   0.0.0.0         17/nolabel(v1)
   172.16.0.44/32   172.16.0.4      nolabel/24
   172.16.0.66/32   172.16.0.6      nolabel/21
   172.16.1.0/24    172.16.0.6      nolabel/22
   172.16.5.0/24    172.16.0.4      nolabel/25
   172.16.8.0/24    172.16.0.6      19/23
Route Distinguisher: 100:2
   172.16.0.1/32    172.16.0.6      nolabel/17
   172.16.0.66/32   172.16.0.6      nolabel/21
   172.16.1.0/24    172.16.0.6      nolabel/22
   172.16.8.0/24    172.16.0.6      nolabel/23
```

The PE 1 (see the first figure above) forwarding table contains new label and next-hop information to direct traffic onto the backup path:

```
Router# show mpls forwarding-table vrf v1 172.16.0.1 detail

Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
16         17         172.16.0.1/32[V] 0             Et1/0      172.16.3.2
        MAC/Encaps=14/22, MRU=1496, Label Stack{21 17}
        AABBCC000D00AABBCC000C018847 0001500000011000
        VPN route: v1
        No output feature configured
```

Example 3: After Local Label Expiration and BGP Reconvergence

Because the local label preservation window has expired, the replacement local label is now gone from the PE 1 forwarding table information:

```
Router# show mpls forwarding-table vrf v1 172.16.0.1 detail
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
None       17         172.16.0.1/32[V] 0             Et1/0      172.16.3.2
       MAC/Encaps=14/22, MRU=1496, Label Stack{21 17}
       AABBCC000D00AABBCC000C018847 0001500000011000
       VPN route: v1
       No output feature configured
```

The new BGP information reverts to the configuration shown in the first figure above:

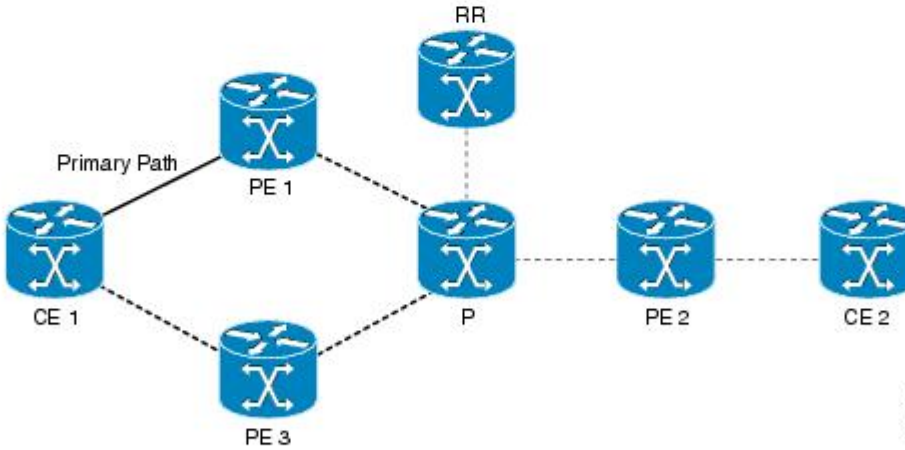
```
Router# show bgp vpnv4 unicast all 172.16.0.1
BGP routing table entry for 100:1:172.16.0.1/32, version 23
Paths: (1 available, best #1, table v1)
  Not advertised to any peer
  100, imported path from 100:2:172.16.0.1/32
    172.16.0.6 (metric 21) from 172.16.0.7 (172.16.0.7)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:0
      Originator: 172.16.0.6, Cluster list: 172.16.0.7
      mpls labels in/out nolabel/17
BGP routing table entry for 100:2:172.16.0.1/32, version 9
Paths: (1 available, best #1, no table)
  Not advertised to any peer
  100
    172.16.0.6 (metric 21) from 172.16.0.7 (172.16.0.7)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:0
      Originator: 172.16.0.6, Cluster list: 172.16.0.7
      mpls labels in/out nolabel/17
Router# show bgp vpnv4 unicast all labels

Network          Next Hop      In label/Out label
Route Distinguisher: 100:1 (v1)
   172.16.0.1/32    172.16.0.6      nolabel/17
   172.16.0.5/32    172.16.0.4      nolabel/23
   172.16.0.22/32   0.0.0.0         17/nolabel(v1)
   172.16.0.44/32   172.16.0.4      nolabel/24
   172.16.0.66/32   172.16.0.6      nolabel/21
   172.16.1.0/24    172.16.0.6      nolabel/22
   172.16.5.0/24    172.16.0.4      nolabel/25
   172.16.8.0/24    172.16.0.6      nolabel/23
Route Distinguisher: 100:2
   172.16.0.1/32    172.16.0.6      nolabel/17
   172.16.0.66/32   172.16.0.6      nolabel/21
   172.16.1.0/24    172.16.0.6      nolabel/22
   172.16.8.0/24    172.16.0.6      nolabel/23
```

Example MPLS VPN--BGP Local Convergence for 6VPE 6PE

You can display a detailed view of local link protection before, during, and after BGP local convergence for Cisco IOS VPN IPv6 provider edge routers (6VPE) and Cisco IOS IPv6 provider edge routers (6PE) over MPLS by using the **show bgp vpnv6** and **show mpls forwarding-table vrf** commands as shown in the following three-stage example.

The figure below shows an MPLS VPN with BGP local convergence configured. The PE to CE routing protocol is eBGP, and the PE to route reflector (RR) sessions are BGP VPNv6. The protected prefix is the CE 1 loopback (2001:0DB8::/128). The primary path is from PE 1 to CE 1. The secondary path is from PE 1, through P and PE3, to CE 1.


RR
Primary Path
PE 1
CE 1
P
PE 2
CE 2
PE 3

Example 1: Before the Link Failure

Both a primary path and a backup path have been configured for the prefix 2001:0DB8::/128. The inlabel/outlabel settings for the two paths are 28/28 and 28/nolabel.

```
Router# show bgp vpnv6 unicast all 2001:0DB8::/128
BGP routing table entry for [1:1]2001:0DB8::/128, version 5
Paths: (2 available, best #2, table v1)
 Advertised to update-groups:
     2
  100, imported path from [2:2]2001:0DB8::/128
    ::FFFF:10.6.6.6 (metric 21) from 10.7.7.7 (10.7.7.7)
      Origin incomplete, metric 0, localpref 100, valid, internal
      Extended Community: RT:1:1
      Originator: 10.6.6.6, Cluster list: 10.7.7.7
      mpls labels in/out 28/28
  100
    2001:0DB8:0:ABCD::1 (FE80::A8BB:CCFF:FE00:B00) from 2001:0DB8:0:ABCD::1 (10.1.1.1)
      Origin incomplete, metric 0, localpref 100, valid, external, best

      Extended Community: RT:1:1
      mpls labels in/out 28/nolabel
BGP routing table entry for [2:2]2001:0DB8::/128, version 11
Paths: (1 available, best #1, no table)
  Not advertised to any peer
  100
    ::FFFF:10.6.6.6 (metric 21) from 10.7.7.7 (10.7.7.7)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:1:1
      Originator: 10.6.6.6, Cluster list: 10.7.7.7
      mpls labels in/out nolabel/28
```

The PE 1 forwarding table contains new label and next-hop information to direct traffic onto the backup path:

```
Router#
show mpls forwarding-table vrf v1 2001:0DB8::/128 detail
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
28         No Label   2001:0DB8::/128[V]     804           Et0/0
FE80::A8BB:CCFF:FE00:B00
        MAC/Encaps=14/14, MRU=1504, Label Stack{}
        AABBCC000B00AABBCC000C0086DD
        VPN route: v1
        No output feature configured
```

Example 2: After the Link Failure

After the link failure, the backup path is still available, the original path is removed from BGP, and the backup path is activated:

```
Router# show mpls forwarding-table vrf v1 2001:0DB8::/128 detail
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
28         28         2001:0DB8::/128[V]     0             Et1/0      10.3.0.2
        MAC/Encaps=14/22, MRU=1496, Label Stack{23 28}
        AABBCC000D00AABBCC000C018847 000170000001C000
        VPN route: v1
        No output feature configured
```

After a configured length of time, the local label expires. The output from the **show mpls forwarding-table**command also verifies that the local label has expired:

```
Router# show mpls forwarding-table vrf v1 2001:0DB8::/128 detail
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
None       28         2001:0DB8::/128[V]     0             Et1/0      10.3.0.2
        MAC/Encaps=14/22, MRU=1496, Label Stack{23 28}
```

```
        AABBCC000D00AABBCC000C018847 000170000001C000
        VPN route: v1
        No output feature configured
Example 3: After the Link Is Restored
```

When the link is restored the original path is added to BGP and the traffic switches back to this path:

```
Router# show bgp vpnv6 unicast all 2001:0DB8::/128
BGP routing table entry for [1:1]2001:0DB8::/128, version 28
Paths: (2 available, best #1, table v1)
  Advertised to update-groups:
     2
  100
    2001:0DB8:0:ABCD::1 (FE80::A8BB:CCFF:FE00:B00) from 2001:0DB8:0:ABCD::1 (10.1.1.1)
      Origin incomplete, metric 0, localpref 100, valid, external, best
      Extended Community: RT:1:1
      mpls labels in/out 16/nolabel
  100, imported path from [2:2]2001:0DB8::/128
    ::FFFF:10.6.6.6 (metric 21) from 10.7.7.7 (10.7.7.7)
      Origin incomplete, metric 0, localpref 100, valid, internal
      Extended Community: RT:1:1
      Originator: 10.6.6.6, Cluster list: 10.7.7.7
      mpls labels in/out 16/28
BGP routing table entry for [2:2]2001:0DB8::/128, version 11
Paths: (1 available, best #1, no table)
  Not advertised to any peer
  100
    ::FFFF:10.6.6.6 (metric 21) from 10.7.7.7 (10.7.7.7)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:1:1
      Originator: 10.6.6.6, Cluster list: 10.7.7.7
      mpls labels in/out nolabel/28
Router# show mpls for vrf v1 2001:0DB8::/128 detail
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
16         No Label   2001:0DB8::/128[V] 0         Et0/0      FE80::A8BB:CCFF:FE00:B00
        MAC/Encaps=14/14, MRU=1504, Label Stack{}
        AABBCC000B00AABBCC000C0086DD
        VPN route: v1
        No output feature configured
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP configuration	Configuring a Basic BGP Network
Protocol for quickly detecting failed forwarding paths	Bidirectional Forwarding Detection
Configuration of BGP PIC Edge for IP and MPLS-VPN	BGP PIC Edge for IP and MPLS-VPN
Configuration of internal BGP features	Configuring Internal BGP Features
Configuration of VRF under the specific cases of IPv4 and IPv6 situations	MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs

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

Feature Information for MPLS VPN--BGP Local Convergence

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

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

Table 5 Feature Information for MPLS VPN--BGP Local Convergence

Feature Name	Releases	Feature Information
MPLS VPN--BGP Local Convergence	12.2(33)SRC 12.2(33)SB 15.0(1)M	This feature reduces the downtime of a PE-CE link failure by rerouting PE-egress traffic onto a backup path to the CE before BGP has reconverged. In 12.2(33)SRC, this feature was introduced on the Cisco 7200 and the Cisco 7600. In 12.2(33)SB, this feature became available on the Cisco 7300 series and the Cisco 10000 series routers. This feature was integrated into Cisco IOS Release 15.0(1)M. The following command was introduced: **protection local-prefixes**.
MPLS VPN--BGP Local Convergence for 6VPE/6PE	15.0(1)S	This feature implements MPLS VPN--BGP local convergence for Cisco IOS 6VPE and Cisco IOS 6PE over MPLS. The following command was modified: **protection local-prefixes**.

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

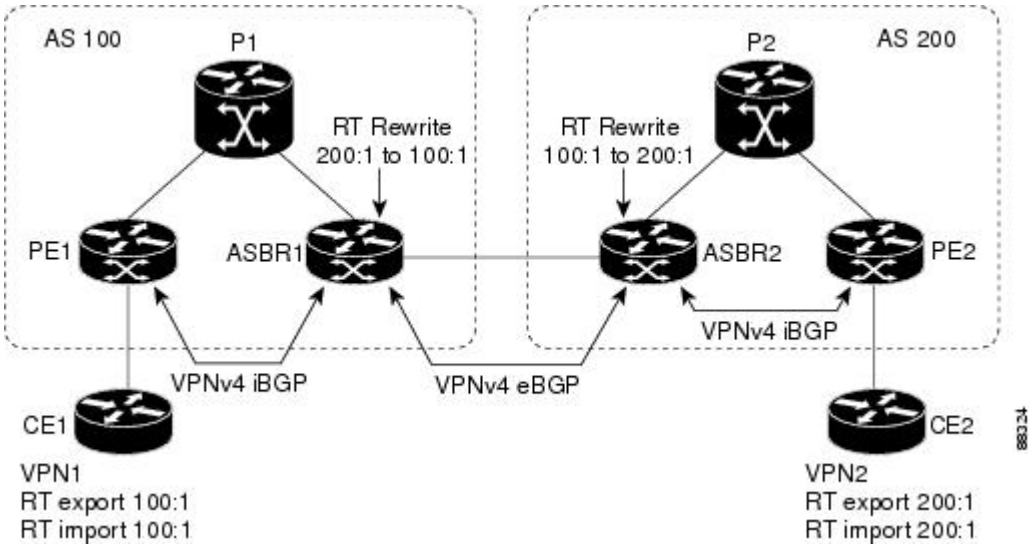
Routing policies for a peer include all configurations that may impact inbound or outbound routing table updates. The MPLS VPN Route Target Rewrite feature can influence routing table updates by allowing the replacement of route targets on inbound and outbound BGP updates. Route targets are carried as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

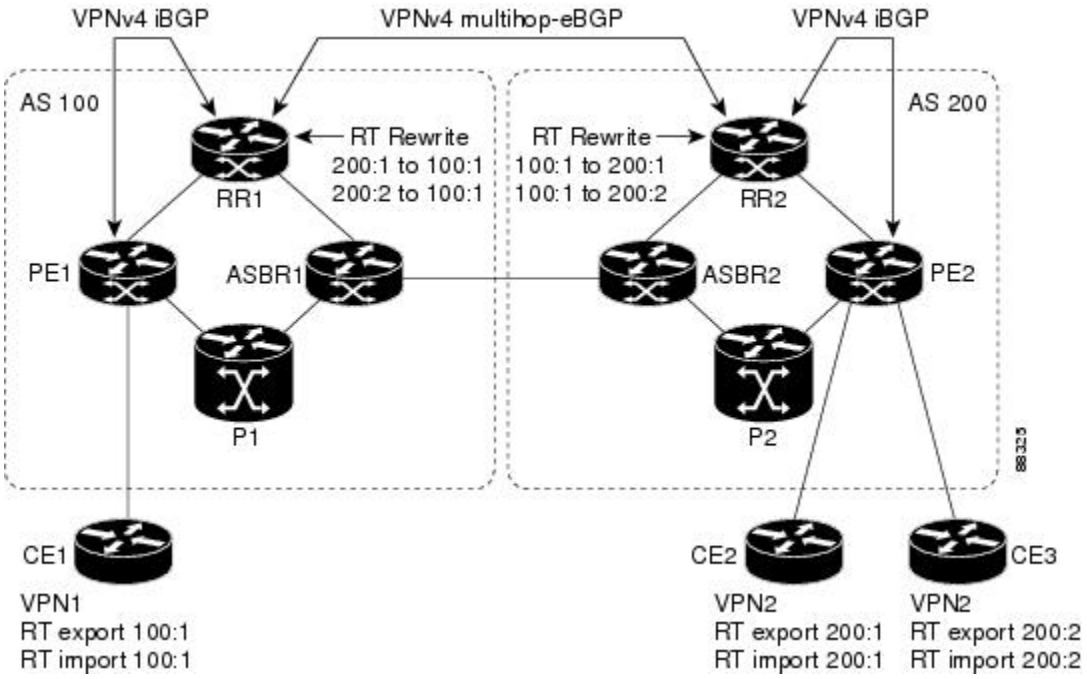
Figure 6 Route Target Replacement on ASBRs in an MPLS VPN Interautonomous System Topology



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 7 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 107
- Refreshing BGP Session to Apply Route Target Replacement Policy, page 109
- Troubleshooting Tips, page 110

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, page 115
- Refreshing the BGP Session to Apply the Route Target Replacement Policy Example, page 116

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 - Per VRF Label

The MPLS VPN - Per VRF Label feature (hereafter, in this document, referred to as the Per VRF Label feature or the Per VRF feature) allows you to configure a single Virtual Private Network (VPN) label for all local routes in the entire VPN routing and forwarding (VRF) domain on Cisco 6500 routers. This MPLS VPN - Per VRF Label feature incorporates a single (per VRF) VPN label that for all local routes in the VRF table.

You can enable (or disable) the MPLS VPN - Per VRF Label feature in global configuration mode. This feature is available for the Cisco 6500 router only.

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 the Per VRF Label Feature

- If your VRF domain has the external/internal Border Gateway Protocol (EIBGP) multipath feature or the Carrier Supporting Carrier (CSC) feature enabled, disable those features before you configure the Per VRF Label feature.
- Before configuring Multiprotocol Label Switching (MPLS) Layer 3 VPNs, you must have MPLS, Label Distribution Protocol (LDP), and Cisco Express Forwarding (CEF) installed in your network.

All routers in the core, including the Provider Edge (PE) routers, must be able to support CEF and MPLS forwarding.

Restrictions for the Per VRF Label Feature

- Enabling the Per VRF Label feature causes BGP reconvergence, which can result in data loss for traffic coming from the MPLS VPN core.

Note You can minimize network disruption by enabling this feature during a scheduled MPLS maintenance window. Also, if possible, avoid enabling this feature on a live router.

- There is no performance degradation when you configure up to 511 VRFs; however, when you add more than 511 VRFs, your network might experience some minor performance degradation (similar to the normal degradation experienced by any of the directly connected VRF prefixes present in the router).
- Per-prefix MPLS counters for VPN prefixes are lost when you enable the Per VRF Label feature.
- You cannot use this feature with CSC and EIBGP multipath features.

Information About the Per VRF Label Feature

MPLS VPN - Per VRF Label Functionality

The PE stores both local and remote routes and includes a label entry for each route. For distributed platforms, the per-prefix labels consume memory. When there are many VRFs and routes, the amount of memory that the per-prefix labels consume can become an issue.

This new Per VRF Label feature allows the advertisement of a single VPN label for local routes throughout the entire VRF. The router uses a new VPN label for the VRF decoding and IP-based lookup to learn where to forward packets for the PE or customer edge (CE) interfaces.

The following conditions apply when you configure the Per VRF Label feature:

- The VRF uses one label for all local routes.
- When you *enable* the Per VRF Label feature, any existing Per VRF Aggregate label is used. If no Per VRF Aggregate label is present, the software creates a new Per VRF label.
- When you *enable* the Per VRF Label feature, the CE router's learned local routes will experience some data loss.

The CE does not lose data when you disable the Per VRF Label feature because when you disable the feature, the configuration reverts to the default labeling configuration of the Cisco 6500 platform, which uses the Per VRF Aggregate label from the local nonCE-sourced routes.

- When you *disable* the Per VRF Label feature, the configuration reverts to the default configuration of the Cisco 6500 routers.
- A Per VRF label forwarding entry is deleted only if the VRF or the BGP configuration is removed.

Summarization of Label Allocation Modes

The table below defines the label allocations used with various route types.

Table 7 Label Allocation Modes

Route Types	Label Mode: Cisco 6500 Default	Label Mode: Per VRF Label Feature
Local to the PE (connected, static route to NULL0, BGP aggregates), redistributed to BGP	Per VRF Aggregate label	Per VRF label
Locally learned from CE (through EBGP or other PE or CE protocols)	Per Prefix label	Per VRF label

How to Configure the Per VRF Label Feature

Configuring the Per VRF Label Feature

To configure the Per VRF Label feature, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label mode** {**vrf** *vrf-name* | **all-vrfs**} **protocol bgp-vpnv4** {**per-prefix** | **per-vrf**}
4. **end**
5. **show ip vrf detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `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	**mpls label mode** {**vrf** *vrf-name* | **all-vrfs**} **protocol bgp-vpnv4** {**per-prefix** | **per-vrf**} **Example:** `Router(config)# mpls label mode all-vrfs protocol bgp-vpnv4 per-vrf`	Configures the Per VRF Label feature.
Step 4	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 5	**show ip vrf detail** **Example:** `Router# show ip vrf detail`	Displays the VRF label mode.

- Examples, page 124

Examples

The following command example shows how to verify the Per VRF Label configuration:

In this example output, the **bold** text indicates the label modes:

```
Router# show ip vrf detail
VRF vpn1; default RD 1:1; default VPNID <not set>
VRF Table ID = 1
  Interfaces:
    Ethernet0/0             Serial5/0               Loopback1
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:1:1
  Import VPN route-target communities
    RT:1:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-vrf (Label 19)
VRF vpn2; default RD 2:1; default VPNID <not set>
VRF Table ID = 2
  Interfaces:
    Ethernet2/0             Loopback2
  Connected addresses are not in global routing table
  Export VPN route-target communities
```

```
    RT:2:1
  Import VPN route-target communities
    RT:2:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-vrf (Label 20)
VRF vpn3; default RD 3:1; default VPNID <not set>
VRF Table ID = 3
  Interfaces:
    Ethernet3/0             Loopback3
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:3:1
  Import VPN route-target communities
    RT:3:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-vrf (Label 23)
Router# show ip bgp vpnv4 all labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 1:1 (vpn1)
   127.0.0.1/32     192.168.1.1     IPv4 VRF Aggr:19/nolabel
   127.0.0.5/32     127.0.0.4       nolabel/19
   192.168.1.0/24   192.168.1.1     IPv4 VRF Aggr:19/nolabel
                    0.0.0.0         IPv4 VRF Aggr:19/aggregate(vpn1)
   192.168.4.0/24   127.0.0.4       nolabel/20
   172.16.0.0/16    0.0.0.0         IPv4 VRF Aggr:19/aggregate(vpn1)
   172.16.128.0/32  192.168.1.1     IPv4 VRF Aggr:19/nolabel
Route Distinguisher: 2:1 (vpn2)
   127.0.2.2/32     0.0.0.0         IPv4 VRF Aggr:20/aggregate(vpn2)
   127.0.0.6/32     192.168.5.1     IPv4 VRF Aggr:20/nolabel
   192.168.5.0/24   0.0.0.0         IPv4 VRF Aggr:20/aggregate(vpn2)
   172.17.128.0/32  192.168.5.1     IPv4 VRF Aggr:20/nolabel
Route Distinguisher: 3:1 (vpn3)
   127.0.3.2/32     0.0.0.0         IPv4 VRF Aggr:23/aggregate(vpn3)
   127.0.0.8/32     192.168.7.1     IPv4 VRF Aggr:23/nolabel
   192.168.7.0/24   0.0.0.0         IPv4 VRF Aggr:23/aggregate(vpn3)
   172.16.128.0/32  192.168.7.1     IPv4 VRF Aggr:23/nolabel
Router# show mpls forwarding-table

Local  Outgoing    Prefix              Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
16     Pop tag     192.168.3.0/24    0          Et1/0      192.168.2.3
17     Pop tag     127.0.0.3/32      0          Et1/0      192.168.2.3
18     17          127.0.0.4/32      0          Et1/0      192.168.2.3
19     Pop Label   IPv4 VRF[V]       0          aggregate/vpn1
20     Pop Label   IPv4 VRF[V]       0          aggregate/vpn2
23     Pop Label   IPv4 VRF[V]       0          aggregate/vpn3
PE1#
```

Configuration Examples for the Per VRF Label feature

No Label Mode for Cisco 6500 Router Default Example

The following example shows the default label mode configuration (no label mode) for the Cisco 6500 router.

In this example output, the **bold** text indicates the label modes:

```
Router# show ip vrf detail
```

```
VRF vpn1; default RD 1:1; default VPNID <not set>
VRF Table ID = 1
  Interfaces:
    Ethernet0/0              Serial5/0               Loopback1
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:1:1
  Import VPN route-target communities
    RT:1:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-prefix
    per-vrf-aggr for connected and BGP aggregates (Label 19)
VRF vpn2; default RD 2:1; default VPNID <not set>
VRF Table ID = 2
  Interfaces:
    Ethernet2/0              Loopback2
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:2:1
  Import VPN route-target communities
    RT:2:1
  No import route-map
  No export route-map
CSC is not configured.

VRF label allocation mode: per-prefix

    per-vrf-aggr for connected and BGP aggregates (Label 20)
VRF vpn3; default RD 3:1; default VPNID <not set>
VRF Table ID = 3
  Interfaces:
    Ethernet3/0              Loopback3
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:3:1
  Import VPN route-target communities
    RT:3:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-prefix
    per-vrf-aggr for connected and BGP aggregates (Label 23)
Router# show ip bgp vpnv4 all labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 1:1 (vpn1)
   127.0.0.1/32     192.168.1.1     27/nolabel
   127.0.0.5/32     127.0.0.4       nolabel/19
   192.168.1.0/24   192.168.1.1     IPv4 VRF Aggr:19/nolabel
                    0.0.0.0         IPv4 VRF Aggr:19/aggregate(vpn1)
   192.168.4.0/24   127.0.0.4       nolabel/20
   172.16.0.0/16    0.0.0.0         IPv4 VRF Aggr:19/aggregate(vpn1)
   172.16.128.0/32  192.168.1.1     28/nolabel
Route Distinguisher: 2:1 (vpn2)
   127.0.2.2/32     0.0.0.0         IPv4 VRF Aggr:20/aggregate(vpn2)
   127.0.0.6/32     192.168.5.1     21/nolabel
   192.168.5.0/24   0.0.0.0         IPv4 VRF Aggr:20/aggregate(vpn2)
   172.17.128.0/32  192.168.5.1     22/nolabel
Route Distinguisher: 3:1 (vpn3)
   127.0.3.2/32     0.0.0.0         IPv4 VRF Aggr:23/aggregate(vpn3)
   127.0.0.8/32     192.168.7.1     24/nolabel
   192.168.7.0/24   0.0.0.0         IPv4 VRF Aggr:23/aggregate(vpn3)
   172.16.128.0/32  192.168.7.1     25/nolabel
Router# show mpls forwarding-table
Local  Outgoing    Prefix             Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id       switched   interface
16     Pop tag     192.168.3.0/24    0          Et1/0      192.168.2.3
17     Pop tag     127.0.0.3/32      0          Et1/0      192.168.2.3
18     17          127.0.0.4/32      0          Et1/0      192.168.2.3
19     Pop Label   IPv4 VRF[V]       0          aggregate/vpn1
20     Pop Label   IPv4 VRF[V]       0          aggregate/vpn2
21     Untagged    127.0.0.6/32[V]   0          Et2/0      192.168.5.1
```

```
22     Untagged    172.17.128.0/32[V]0          Et2/0      192.168.5.1
23     Pop Label   IPv4 VRF[V]       0          aggregate/vpn3
24     Untagged    127.0.0.8/32[V]   0          Et3/0      192.168.7.1
25     Untagged    172.16.128.0/32[V]0          Et3/0      192.168.7.1
27     Untagged    127.0.0.1/32[V]   0          Et0/0      192.168.1.1
28     Untagged    172.16.128.0/32[V]0          Et0/0      192.168.1.1
```

Mixed Mode with Global Per-Prefix Example

For this example, the following commands set VPN 1 for per-vrf label mode, VPN 2 for per-prefix label mode, and all remaining VPNs for per-prefix (globally).

In this example output, the **bold** text indicates the label modes:

```
Router# mpls label mode vrf vpn1 protocol bgp-vpnv4 per-vrf
Router# mpls label mode vrf vpn2 protocol bgp-vpnv4 per-prefix
```

Use the following show commands to display the label mode settings:

```
Router# show ip vrf detail
VRF vpn1; default RD 1:1; default VPNID <not set>
VRF Table ID = 1
  Interfaces:
    Ethernet0/0             Serial5/0               Loopback1
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:1:1
  Import VPN route-target communities
    RT:1:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-vrf (Label 26)
VRF vpn2; default RD 2:1; default VPNID <not set>
VRF Table ID = 2
  Interfaces:
    Ethernet2/0             Loopback2
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:2:1
  Import VPN route-target communities
    RT:2:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-prefix
    per-vrf-aggr for connected and BGP aggregates (Label 27)
VRF vpn3; default RD 3:1; default VPNID <not set>
VRF Table ID = 3
  Interfaces:
    Ethernet3/0             Loopback3
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:3:1
  Import VPN route-target communities
    RT:3:1
  No import route-map
  No export route-map
CSC is not configured.

VRF label allocation mode: per-prefix
    per-vrf-aggr for connected and BGP aggregates (Label 28)
Router# show ip bgp vpnv4 all label
   Network          Next Hop      In label/Out label
Route Distinguisher: 1:1 (vpn1)
   127.0.0.1/32     192.168.1.1     IPv4 VRF Aggr:26/nolabel
   127.0.0.5/32     127.0.0.4       nolabel/19
   192.168.1.0/24   0.0.0.0         IPv4 VRF Aggr:26/aggregate(vpn1)
                    192.168.1.1     IPv4 VRF Aggr:26/nolabel
   192.168.4.0/24   127.0.0.4       nolabel/20
```

```
   172.16.0.0/16    0.0.0.0         IPv4 VRF Aggr:26/aggregate(vpn1)
   172.16.128.0/32  192.168.1.1     IPv4 VRF Aggr:26/nolabel
Route Distinguisher: 2:1 (vpn2)
   127.0.2.2/32     0.0.0.0         IPv4 VRF Aggr:27/aggregate(vpn2)
   127.0.0.6/32     192.168.5.1     20/nolabel
   192.168.5.0/24   0.0.0.0         IPv4 VRF Aggr:27/aggregate(vpn2)
   172.17.128.0/32  192.168.5.1     21/nolabel
Route Distinguisher: 3:1 (vpn3)
   127.0.3.2/32     0.0.0.0         IPv4 VRF Aggr:28/aggregate(vpn3)
   127.0.0.8/32     192.168.7.1     22/nolabel
   192.168.7.0/24   0.0.0.0         IPv4 VRF Aggr:28/aggregate(vpn3)
   172.16.128.0/32  192.168.7.1     23/nolabel
Router# show mpls forwarding-table

Local  Outgoing    Prefix              Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
16     Pop tag     192.168.3.0/24    0          Et1/0      192.168.2.3
17     Pop tag     127.0.0.3/32      0          Et1/0      192.168.2.3
18     17          127.0.0.4/32      0          Et1/0      192.168.2.3
20     Untagged    127.0.0.6/32[V]   0          Et2/0      192.168.5.1
21     Untagged    172.17.128.0/32[V]0          Et2/0      192.168.5.1
22     Untagged    127.0.0.8/32[V]   0          Et3/0      192.168.7.1
23     Untagged    172.16.128.0/32[V]0          Et3/0      192.168.7.1
26     Pop Label   IPv4 VRF[V]       0          aggregate/vpn1
27     Pop Label   IPv4 VRF[V]       0          aggregate/vpn1
28     Pop Label   IPv4 VRF[V]       0          aggregate/vpn1
```

Mixed Mode with Global Per-VRF Example

For this example, the following commands set VPN 1 for per-vrf label mode, VPN 2 for per-prefix label mode, and all remaining VPNs for per-vrf (globally).

In this example output, the **bold** text indicates the label modes:

```
Router# mpls label mode vrf vpn1 protocol bgp-vpnv4 per-vrf
Router# mpls label mode vrf vpn2 protocol bgp-vpnv4 per-prefix
Router# mpls label mode all-vrfs protocol bgp-vpnv4 per-vrf
Router# show ip vrf detail
VRF vpn1; default RD 1:1; default VPNID <not set>
VRF Table ID = 1
  Interfaces:
    Ethernet0/0             Serial5/0               Loopback1
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:1:1
  Import VPN route-target communities
    RT:1:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-vrf (Label 26)
VRF vpn2; default RD 2:1; default VPNID <not set>
VRF Table ID = 2
  Interfaces:
    Ethernet2/0             Loopback2
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:2:1
  Import VPN route-target communities
    RT:2:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-prefix
    per-vrf-aggr for connected and BGP aggregates (Label 27)
VRF vpn3; default RD 3:1; default VPNID <not set>
VRF Table ID = 3
  Interfaces:
    Ethernet3/0             Loopback3
  Connected addresses are not in global routing table
  Export VPN route-target communities
```

```
    RT:3:1
  Import VPN route-target communities
    RT:3:1
  No import route-map
  No export route-map
CSC is not configured.
  VRF label allocation mode: per-vrf (Label 28)
Router# show ip bgp vpnv4 all label

   Network          Next Hop      In label/Out label
Route Distinguisher: 1:1 (vpn1)
   127.0.0.1/32     192.168.1.1     IPv4 VRF Aggr:26/nolabel
   127.0.0.5/32     127.0.0.4       nolabel/19
   192.168.1.0/24   0.0.0.0         IPv4 VRF Aggr:26/aggregate(vpn1)
                    192.168.1.1     IPv4 VRF Aggr:26/nolabel
   192.168.4.0/24   127.0.0.4       nolabel/20
   172.16.0.0/16    0.0.0.0         IPv4 VRF Aggr:26/aggregate(vpn1)
   172.16.128.0/32  192.168.1.1     IPv4 VRF Aggr:26/nolabel
Route Distinguisher: 2:1 (vpn2)
   127.0.2.2/32     0.0.0.0         IPv4 VRF Aggr:27/aggregate(vpn2)
   127.0.0.6/32     192.168.5.1     20/nolabel
   192.168.5.0/24   0.0.0.0         IPv4 VRF Aggr:27/aggregate(vpn2)
   172.17.128.0/32  192.168.5.1     21/nolabel
Route Distinguisher: 3:1 (vpn3)
   127.0.3.2/32     0.0.0.0         IPv4 VRF Aggr:28/aggregate(vpn3)
   127.0.0.8/32     192.168.7.1     IPv4 VRF Aggr:28/nolabel
   192.168.7.0/24   0.0.0.0         IPv4 VRF Aggr:28/aggregate(vpn3)
   172.16.128.0/32  192.168.7.1     IPv4 VRF Aggr:28/nolabel
Router# show mpls forwarding-table

Local  Outgoing    Prefix              Bytes tag  Outgoing      Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
16     Pop tag     192.168.3.0/24    0          Et1/0         192.168.2.3
17     Pop tag     127.0.0.3/32      0          Et1/0         192.168.2.3
18     17          127.0.0.4/32      0          Et1/0         192.168.2.3
20     Untagged    127.0.0.6/32[V]   0          Et2/0         192.168.5.1
21     Untagged    172.17.128.0/32[V]0          Et2/0         192.168.5.1
26     Pop Label   IPv4 VRF[V]       0          aggregate/vpn1
27     Pop Label   IPv4 VRF[V]       0
aggregate/vpn2
28     Pop Label   IPv4 VRF[V]       0          aggregate/vpn3
```

Additional References

Related Documents

Related Topic	Document Title
MPLS VPNs	*MPLS Layer 3 VPNs Configuration Guide*

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 Isoftware 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*

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

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Multiprotocol Label Switching Command Reference* at http://www.cisco.com/en/US/docs/ios/mpls/command/reference/mp_book.html . For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases* , at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html .

- **debug ip bgp vpnv4 unicast**
- **mpls label mode**

Feature Information for MPLS VPN - Per VRF Label

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software

release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 8 Feature Information for MPLS VPN - Per VRF Label

Feature Name	Releases	Feature Information
MPLS VPN - Per VRF Label	12.2(33)SRD	This feature allows a user to configure a single VPN label for all local routes in the entire VPN routing and forwarding (VRF) domain on Cisco 6500 routers. The feature incorporates a single (per VRF) VPN label for all local routes in the VRF table. You can enable (or disable) the MPLS VPN - Per VRF Label feature in global configuration mode using a new, hidden, command. This feature is available for the Cisco 6500 router only In 12.2(33)SRD, this feature was integrated.

MPLS VPN 6VPE per VRF Label

The MPLS VPN 6VPE per VRF Label feature allows you to configure a single Virtual Private Network (VPN) label for all local routes in the entire IPv6 VPN routing and forwarding (VRF) domain on Cisco 7600 routers. This MPLS VPN 6VPE per VRF Label feature incorporates a single (per VRF) VPN label for *all* local IPv6 routes in the VRF table.

You can enable (or disable) the MPLS VPN 6VPE per VRF Label feature in global configuration mode. This feature is available for the Cisco 7600 router only.

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 the MPLS VPN 6VPE per VRF Label feature

- If your VRF domain has the external/internal Border Gateway Protocol (EIBGP) multipath feature or the Carrier Supporting Carrier (CSC) feature enabled, disable those features before you configure the MPLS VPN 6VPE per VRF Label feature.
- Before configuring Multiprotocol Label Switching (MPLS) Layer 3 VPNs, you must have MPLS, Label Distribution Protocol (LDP), and Cisco Express Forwarding installed in your network. All routers in the core, including the provider edge (PE) routers, must be able to support Cisco Express Forwarding and MPLS forwarding.

- Before configuring a 6VPE per VRF label, be sure that the IPv6 address family is configured on that VRF.

Restrictions for the MPLS VPN 6VPE per VRF Label feature

- Enabling the MPLS VPN 6VPE per VRF Label feature causes BGP reconvergence, which can result in data loss for traffic coming from the MPLS VPN core.

Note You can minimize network disruption by enabling this feature during a scheduled MPLS maintenance window. Also, if possible, avoid enabling this feature on a live router.

- Per-prefix MPLS counters for VPN prefixes are lost when you enable the MPLS VPN 6VPE per VRF Label feature.
- You cannot use this feature with CSC and EIBGP multipath features.

Information About the MPLS VPN 6VPE per VRF Label feature

- MPLS VPN 6VPE per VRF Label Functionality, page 134

MPLS VPN 6VPE per VRF Label Functionality

The PE router stores both local and remote routes and includes a label entry for each route. For distributed platforms, the multiplicity of per-prefix labels consume memory. When there are many VRFs and routes, the amount of memory that the per-prefix labels consume can cause performance degradation on some platform devices. To avoid this issue, the MPLS VPN 6VPE per VRF Label feature allows the advertisement of a single VPN label for local routes throughout the entire VRF. The router uses a new VPN label for the VRF decoding and IP-based lookup to learn where to forward packets for the PE or customer edge (CE) interfaces.

The following conditions apply when you configure the MPLS VPN 6VPE per VRF Label feature:

- The VRF uses one label for all local routes.
- When you enable the MPLS VPN 6VPE per VRF Label feature, any existing per VRF aggregate label is used. If no per VRF aggregate label is present, the software creates a new 6VPE per VRF label.
- When you enable the MPLS VPN 6VPE per VRF Label feature, the CE router's learned local routes will experience some data loss.

The CE does not lose data when you disable the MPLS VPN 6VPE per VRF Label feature because the configuration reverts to the default labeling configuration of the Cisco 7600 platform, which uses the Per VRF Aggregate label from the local nonCE-sourced routes.

- When you disable the MPLS VPN 6VPE per VRF Label feature, the configuration reverts to the default configuration of the Cisco 7600 routers.
- A 6VPE Per VRF Label forwarding entry is deleted only if the VRF, the IPv6 VRF address family, or the BGP configuration is removed.

See the Implementing IPv6 VPN over MPLS (6VPE) configuration guide for detailed information about IPv6 VPN services and 6VPE.

Summarization of Label Allocation Modes

The table below defines the label allocations used with various route types.

Table 9 Label Allocation Modes

Route Types	Label Mode: Cisco 7600 Default	Label Mode: MPLS VPN 6VPE per VRF Label feature
Local to the PE (connected, static route to NULL0, BGP aggregates), redistributed to BGP	Per VRF Aggregate label	6VPE Per VRF Label
Locally learned from CE (through external BGP or other PE or CE protocols)	Per Prefix label	6VPE Per VRF Label

How to Configure the MPLS VPN 6VPE per VRF Label Feature

Configuring the MPLS VPN 6VPE per VRF Label Feature

To configure a single (per VRF) VPN label for all local IPv6 routes in the VRF table, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label mode** {**vrf** *vrf-name* | **all-vrfs**} **protocol** {**bgp-vpnv6** | **all-afs**} {**per-prefix** | **per-vrf**}
4. **end**
5. **show vrf detail** *vrf-name*

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	**mpls label mode** {**vrf** *vrf-name* | **all-vrfs**} **protocol** {**bgp-vpnv6** | **all-afs**} {**per-prefix** | **per-vrf**} **Example:** `Router(config)# mpls label mode all-vrfs protocol bgp-vpnv6 per-vrf`	Configures a single (per VRF) VPN label for all local IPv6 routes in the VRF table.
Step 4	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 5	**show vrf detail** *vrf-name* **Example:** `Router# show vrf detail vpn1`	Displays the VRF label mode for the specified VRF.

- Examples, page 136
- Troubleshooting Tips, page 137

Examples

The following example shows how to verify the 6VPE per VRF label configuration.

In this example output, the **bold** text indicates the 6VPE per VRF label mode for VPN1.

```
Router# show vrf detail vpn1
VRF vpn1 (VRF Id = 1); default RD 1:1; default VPNID <not set>
  Interfaces:
    GE4/1                   Lo1
Address family ipv4 (Table ID = 1 (0x1)):
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:1:1
  Import VPN route-target communities
    RT:1:1                  RT:2:2
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
    vrf-conn-aggr for connected and BGP aggregates (Label 17)
Address family ipv6 (Table ID = 503316481 (0x1E000001)):
  Connected addresses are not in global routing table
  Export VPN route-target communities
```

```
    RT:1:1
  Import VPN route-target communities
    RT:1:1
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-vrf (Label 18)

Router# show bgp vpnv6 unicast vrf vpn1 label
   Network          Next Hop      In label/Out label
Route Distinguisher: 1:1 (vpn1)
   2001:DB8:1:2::/96
                    2001:DB8:1:2::1 IPv6 VRF Aggr:18/nolabel
                    ::              IPv6 VRF Aggr:18/nolabel(vpn1)
   2001:DB8:4:5::/96
                    ::FFFF:127.0.0.4
                                    nolabel/17
   2001:DB8:2::1/128
                    ::              IPv6 VRF Aggr:18/nolabel(vpn1)
   2001:DB8:4::1/128
                    ::FFFF:127.0.0.4
                                    nolabel/18
   2001:DB8:CE2::1/128
                    ::FFFF:127.0.0.4
                                    nolabel/19
   2001:DB8:CE1::1/128
                    2001:DB8:1:2::1 IPv6 VRF Aggr:18/nolabel

Router# show mpls forwarding
Local  Outgoing      Prefix           Bytes Label   Outgoing   Next Hop
Label  Label or VC   or Tunnel Id     Switched      interface
16     Pop Label     127.0.0.4/32     0             AT3/0/0.1  point2point
17     Pop Label     IPv4 VRF[V]      0             aggregate/vpn1
18     Pop Label     IPv6 VRF[V]      0             aggregate/vpn1
```

Troubleshooting Tips

The **debug ip bgp vpnv6 unicast** command can help troubleshoot the 6VPE per VRF label configuration.

Configuration Examples for MPLS VPN 6VPE per VRF Label

6VPE No Label Mode for Cisco 7600 Router Default Example

The following example shows the 6VPE default label mode configuration (no label mode) for the Cisco 7600 router.

In this example output, the **bold** text indicates the default label mode for VPN1.

```
Router# show vrf detail vpn1
VRF vpn1 (VRF Id = 1); default RD 1:1; default VPNID <not set>
  Interfaces:
    GE4/1                   Lo1
Address family ipv4 (Table ID = 1 (0x1)):
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:1:1
  Import VPN route-target communities
    RT:1:1                  RT:2:2
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
```

```
    vrf-conn-aggr for connected and BGP aggregates (Label 17)
Address family ipv6 (Table ID = 503316481 (0x1E000001)):
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:1:1
  Import VPN route-target communities
    RT:1:1
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
    vrf-conn-aggr for connected and BGP aggregates (Label 18)

Router# show bgp vpnv6 unicast vrf vpn1 label
   Network          Next Hop      In label/Out label
Route Distinguisher: 1:1 (vpn1)
   2001:DB8:1:2::/96
                    2001:DB8:1:2::1 IPv6 VRF Aggr:18/nolabel
                    ::              IPv6 VRF Aggr:18/nolabel(vpn1)
   2001:DB8:4:5::/96
                    ::FFFF:127.0.0.4
                                    nolabel/17
   2001:DB8:2::1/128
                    ::              IPv6 VRF Aggr:18/nolabel(vpn1)
   2001:DB8:4::1/128
                    ::FFFF:127.0.0.4
                                    nolabel/18
   2001:DB8:CE2::1/128
                    ::FFFF:127.0.0.4
                                    nolabel/19
   2001:DB8:CE1::1/128
                    2001:DB8:1:2::1 19/nolabel
Router# show mpls forwarding
Local  Outgoing      Prefix            Bytes Label   Outgoing   Next Hop
Label  Label or VC   or Tunnel Id      Switched      interface
16     Pop Label     127.0.0.4/32      0             AT3/0/0.1  point2point
17     Pop Label     IPv4 VRF[V]       0             aggregate/vpn1
18     Pop Label     IPv6 VRF[V]       0             aggregate/vpn1
19     No Label      2001:DB8:CE1::1/128[V]
                                       0             GE4/1      FE80::20C:CFFF:FEAD:A00A
```

Additional References

Related Documents

Related Topic	Document Title
MPLS VPNs	• Configuring MPLS Virtual Private Networks • Implementing IPv6 VPN over MPLS (6VPE)

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*

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 6VPE per VRF Label

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

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

Table 10 Feature Information for MPLS VPN 6VPE per VRF Label

Feature Name	Releases	Feature Information
MPLS VPN 6VPE per VRF Label	12.2(33)SRD	This feature allows a user to configure a single VPN label for all local routes in the entire IPv6 VPN routing and forwarding (VRF) domain on Cisco 7600 routers. The feature incorporates a single (per VRF) VPN label for all local IPv6 routes in the VRF table. You can enable (or disable) the MPLS VPN 6VPE per VRF Label feature in global configuration mode. This feature is available for the Cisco 7600 only. In Release 12.2(33)SRD, this feature was introduced on the Cisco 7600 router. The following commands were introduced: **debug ip bgp vpnv6 unicast** and **mpls lavel mode** (6VPE).

MPLS Multi-VRF (VRF-Lite)

The MPLS Multi-VRF feature allows you to configure and maintain more than one instance of a routing and forwarding table within the same customer edge (CE) 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 Multi-VRF

The network's core and provider edge routers must be configured for MPLS Virtual Private Network (VPN) operation.

Restrictions for MPLS Multi-VRF

You can configure the MPLS Multi-VRF feature only on Layer 3 interfaces.

The MPLS Multi-VRF feature is not supported by Interior Gateway Routing Protocol (IGRP) nor IS-IS.

Label distribution for a given VPN routing and forwarding (VRF) instance on a given router can be handled by either Border Gateway Protocol (BGP) or Label Distribution Protocol (LDP), but not by both protocols at the same time.

Multicast cannot operate on a Layer 3 interface that is configured with the MPLS Multi-VRF feature.

Multicast cannot be configured at the same time on the same layer 3 interface as the MPLS Multi-VRF feature.

Information About MPLS Multi-VRF

How the MPLS Multi-VRF Feature Works

The MPLS Multi-VRF feature enables a service provider to support two or more VPNs, where the IP addresses can overlap several VPNs. The MPLS Multi-VRF feature uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF. Interfaces in a VRF can be either physical, such as FastEthernet ports, or logical, such as VLAN Switched Virtual Interfaces (SVIs), but a Layer 3 interface cannot belong to more than one VRF at any one time. The Multi-VRF feature allows an operator to support two or more routing domains on a CE router, with each routing domain having its own set of interfaces and its own set of routing and forwarding tables. The MPLS Multi-VRF feature makes it possible to extend the Label Switched Paths (LSPs) to the CE and into each routing domain that the CE supports.

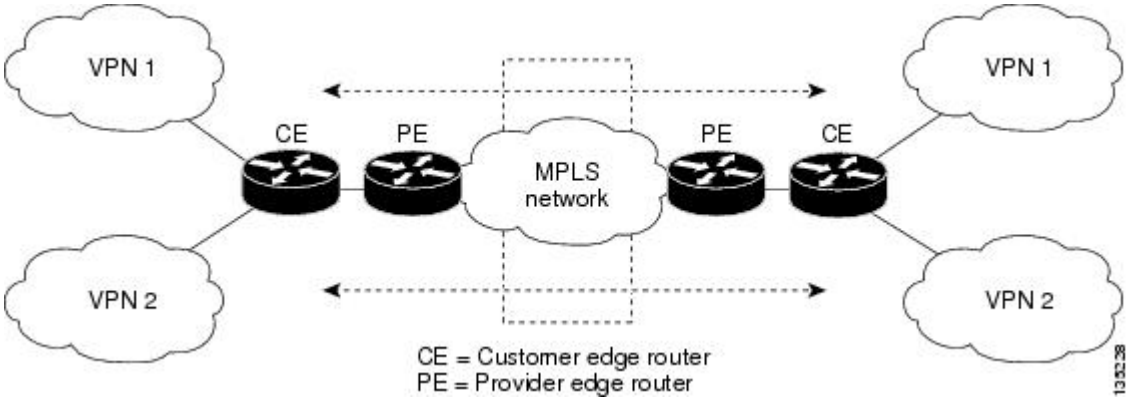
The MPLS Multi-VRF feature works as follows:

- Each CE router advertises its site's local routes to a provider edge (PE) router and learns the remote VPN routes from that PE router.
- PE routers exchange routing information with CE routers by using static routing or a routing protocol such as BGP, RIPv1, or RIPv2.
- PE routers exchange MPLS label information with CE routers through LDP or BGP.
- The PE router needs to maintain VPN routes only for those VPNs to which it is directly attached, eliminating the requirement that the PE maintain all of the service provider's VPN routes. Each PE router maintains a VRF for each of its directly connected sites. Two or more interfaces on a PE router can be associated with a single VRF if all the sites participate in the same VPN. Each VPN is mapped to a specified VRF. After learning local VPN routes from CE routers, the PE router exchanges VPN routing information with other PE routers through internal BGP (iBPG).

With the MPLS Multi-VRF feature, two or more customers can share one CE router, and only one physical link is used between the CE and the PE routers. The shared CE router maintains separate VRF tables for each customer and routes packets for each customer based on that customer's own routing table. The MPLS Multi-VRF feature extends limited PE router functionality to a CE router, giving it the ability, through the maintenance of separate VRF tables, to extend the privacy and security of a VPN to the branch office.

The figure below shows a configuration where each CE router acts as if it were two CE routers. Because the MPLS Multi-VRF feature is a Layer 3 feature, each interface associated with a VRF must be a Layer 3 interface.

Figure 8 Each CE Router Acting as Several Virtual CE Routers



How Packets Are Forwarded in a Network Using the MPLS Multi-VRF Feature

Following is the packet-forwarding process in an MPLS Multi-VRF CE-enabled network, as illustrated in the figure above :

- When the CE receives a packet from a VPN, it looks up the routing table based on the input interface. When a route is found, the CE imposes the MPLS label it received from the PE for that route and forwards the packet to the PE.
- When the ingress PE receives a packet from the CE, it swaps the incoming label with the corresponding label stack and sends it to the MPLS network.
- When an egress PE receives a packet from the network, it swaps the VPN label with the label it earlier had received for the route from the CE, and forwards it to the CE.
- When a CE receives a packet from an egress PE, it uses the incoming label on the packet to forward the packet to the correct VPN.

To configure Multi-VRF, you create a VRF table and then specify the Layer 3 interface associated with that VRF. Next, you configure the routing protocols within the VPN, and between the CE and the PE. BGP is the preferred routing protocol for distributing VPN routing information across the provider's backbone. For more information, see the How to Configure MPLS Multi-VRF, page 144 .

The Multi-VRF network has three major components:

- VPN route target communities: These are lists of all other members of a VPN community. You need to configure VPN route targets for each VPN community member.
- Multiprotocol BGP peering of VPN community PE routers: This propagates VRF reachability information to all members of a VPN community. You need to configure BGP peering in all PE routers within a VPN community.
- VPN forwarding: This transports all traffic between VPN community members across a VPN service-provider network.

Considerations for Configuring MPLS Multi-VRF

When BGP is used as the routing protocol, it can also be used for MPLS label exchange between the PE and CE routers. By contrast, if Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP), RIP, or static routing is used, LDP must be used to signal labels.

To configure the MPLS Multi-VRF feature, create a VRF table, specify the Layer 3 interface associated with that VRF, and then configure the routing protocols within the VPN and between the CE and the PE routers.

Consider these points when configuring the MPLS Multi-VRF feature in your network:

- A router with the MPLS Multi-VRF feature is shared by several customers, and each customer has its own routing table.
- Because each customer uses a different VRF table, the same IP addresses can be reused. Overlapping IP addresses are allowed in different VPNs.
- The MPLS Multi-VRF feature lets several customers share the same physical link between the PE and CE routers. Trunk ports with several VLANs separate packets among the customers. Each customer has its own VLAN.
- For the PE router, there is no difference between using the MPLS Multi-VRF feature or using several CE routers. In the figure above, for example, four virtual Layer 3 interfaces are connected to the MPLS Multi-VRF CE router.
- The MPLS Multi-VRF feature does not affect the packet switching rate.

How to Configure MPLS Multi-VRF

Configuring VRFs

Perform the following task to configure VRFs on both the PE and CE routers:

If a VRF has not been configured, the router has the following default configuration:

- No VRFs have been defined.
- No import maps, export maps, or route maps have been defined.
- No VRF maximum routes exist.
- Only the global routing table exists on the interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip routing**
4. **ip vrf** *vrf-name*
5. **rd** *route-distinguisher*
6. **route-target** {**export** | **import** | **both**} *route-target-ext-community*
7. **import map** *route-map*
8. **exit**
9. **interface** *type-number*
10. **ip vf forwarding** *vrf-name*
11. **end**
12. **show ip vrf**

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 routing** **Example:** `Router(config)# ip routing`	Enables IP routing.
Step 4	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf v1`	Names the VRF and enters VRF configuration mode.

	Command or Action	Purpose
Step 5	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 100:1`	Creates a VRF table by specifying a route distinguisher. • Enter either an autonomous system number and an arbitrary number (xxx:y), or an IP address and an arbitrary number (A.B.C.D:y).
Step 6	**route-target** {**export** | **import** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf)# route-target export 100:1`	Creates a list of import, export, or import and export route target communities for the specified VRF. • Enter either an autonomous system number and an arbitrary number (xxx:y), or an IP address and an arbitrary number (A.B.C.D:y). **Note** This command works only if BGP is running.
Step 7	**import map** *route-map* **Example:** `Router(config-vrf)# import map importmap1`	(Optional) Associates a route map with the VRF.
Step 8	**exit** **Example:** `Router(config-vrf)# exit`	Returns to global configuration mode.
Step 9	**interface** *type-number* **Example:** `Router(config)# interface fastethernet3/0.10`	Specifies the Layer 3 interface to be associated with the VRF and enters interface configuration mode. • The interface can be a routed port or an SVI.
Step 10	**ip vf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding v1`	Associates the VRF with the Layer 3 interface.
Step 11	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 12	**show ip vrf** **Example:** `Router# show ip vrf`	Displays the settings of the VRFs.

Configuring BGP as the Routing Protocol

Most routing protocols can be used between the CE and the PE routers. However, external BGP (eBGP) is recommended, because:

- ◦ BGP does not require more than one algorithm to communicate with many CE routers.
  ◦ BGP is designed to pass routing information between systems run by different administrations.
  ◦ BGP makes it easy to pass attributes of the routes to the CE router.

When BGP is used as the routing protocol, it can also be used to handle the MPLS label exchange between the PE and CE routers. By contrast, if OSPF, EIGRP, RIP, or static routing is used, LDP must be used to signal labels.

To configure a BGP PE-to-CE routing session, perform the following steps on the CE and on the PE routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **network** *ip-address* **mask** *network-mask*
5. **redistribute ospf** *process-id* **match internal**
6. **network** *ip-address* **area** *area-id*
7. **address-family ipv4 vrf** *vrf-name*
8. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
9. **neighbor** *address* **activate**

DETAILED STEPS

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

Command or Action	Purpose
Step 2 **configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3 **router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 100`	Configures the BGP routing process with the autonomous system number passed to other BGP routers, and enters router configuration mode.
Step 4 **network** *ip-address* **mask** *network-mask* **Example:** `Router(config-router)# network 10.0.0.0 mask 255.255.255.0`	Specifies a network and mask to announce using BGP.
Step 5 **redistribute ospf** *process-id* **match internal** **Example:** `Router(config-router)# redistribute ospf 2 match internal`	Sets the router to redistribute OSPF internal routes.
Step 6 **network** *ip-address* **area** *area-id* **Example:** `Router(config-router)# network 10.0.0.0 255.255.255.0 area 0`	Identifies the network address and mask on which OSPF is running, and the area ID of that network address.
Step 7 **address-family ipv4 vrf** *vrf-name* **Example:** `Router(config-router)# address-family ipv4 vrf v12`	Identifies the name of the VRF instance that will be associated with the next two commands, and enters VRF address-family mode.
Step 8 **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor 10.0.0.3 remote-as 100`	Informs this router's BGP neighbor table of the neighbor's address (or peer group name) and the neighbor's autonomous system number.

Command or Action	Purpose
Step 9 **neighbor** *address* **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.3 activate`	Activates the advertisement of the IPv4 address-family neighbors.

Configuring PE-to-CE MPLS Forwarding and Signaling with BGP

If BGP is used for routing between the PE and CE routers, configure BGP to signal the labels on the VRF interfaces of both the CE and PE routers. You must globally enable signaling at the router-configuration level and for each interface:

- To enable MPLS label signaling via BGP at the router-configuration level, use the **neighbor send-label** command.
- To enable MPLS forwarding on the interface used for the PE-to-CE eBGP session at the interface level, use the **mpls bgp forwarding** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4 vrf** *vrf-name*
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** *address* **send-label** [**explicit-null**]
7. **neighbor** *address* **activate**
8. **end**
9. **configure terminal**
10. **interface** *type number*
11. **mpls bgp forwarding**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 100`	Configures the BGP routing process with the autonomous system number passed to other BGP routers and enters router configuration mode.
Step 4	**address-family ipv4 vrf** *vrf-name* **Example:** `Router(config-router)# address-family ipv4 vrf v12`	Identifies the name of the VRF instance that will be associated with the next two commands and enters address family configuration mode.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor 10.0.0.3 remote-as 100`	Informs this router's BGP neighbor table of the neighbor's address (or peer group name) and the neighbor's autonomous system number.
Step 6	**neighbor** *address* **send-label** [**explicit-null**] **Example:** `Router(config-router-af)# neighbor 10.0.0.3 send-label`	Enables the router to use BGP to distribute MPLS labels along with the IPv4 routes to the peer routers. • If a BGP session is running when you issue this command, the BGP session flaps immediately after the command is issued.
Step 7	**neighbor** *address* **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.3 activate`	Activates the advertisement of the IPv4 address-family neighbors.
Step 8	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 9	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface fastethernet3/0.10`	Enters interface configuration mode for the interface to be used for the BGP session. • The interface can be a routed port or an SVI.
Step 11	**mpls bgp forwarding** **Example:** `Router(config-if)# mpls bgp forwarding`	Enables MPLS forwarding on the interface.

Configuring a Routing Protocol Other than BGP

You can use RIP, EIGRP, OSPF or with static routing. This configuration uses OSPF, but the process is the same for other protocols.

If you use OSPF as the routing protocol between the PE and the CE routers, issue the **capability vrf-lite** command in router configuration mode. See *OSPF Support for Multi-VRF in CE Routers* for more information.

Note If OSPF, EIGRP, RIP, or static routing is used, LDP must be used to signal labels.

The MPLS Multi-VRF feature is not supported by IGRP nor IS-IS.

Multicast cannot be configured on the same Layer 3 interface as the MPLS Multi-VRF feature is configured.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id* [**vrf** *vpn-name*]
4. **log-adjacency-changes**
5. **redistribute bgp** *autonomous-system-number* **subnets**
6. **network** *ip-address subnet-mask* **area** *area-id*
7. **end**
8. **show ip ospf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `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 100 vrf v1`	Enables OSPF routing, specifies a VRF table, and enters router configuration mode.
Step 4	**log-adjacency-changes** **Example:** `Router(config-router)# log-adjacency-changes`	(Optional) Logs changes in the adjacency state. This is the default state.
Step 5	**redistribute bgp** *autonomous-system-number* **subnets** **Example:** `Router(config-router)# redistribute bgp 800 subnets`	Sets the router to redistribute information from the BGP network to the OSPF network.

	Command or Action	Purpose
Step 6	**network** *ip-address subnet-mask* **area** *area-id* **Example:** `Router(config-router)# network 10.0.0.0 255.255.255.0 area 0`	Indicates the network address and mask on which OSPF runs, and the area ID of that network address.
Step 7	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 8	**show ip ospf** **Example:** `Router# show ip ospf`	Displays information about the OSPF routing processes.

Configuring PE-to-CE MPLS Forwarding and Signaling with LDP

If OSPF, EIGRP, RIP, or static routing is used, LDP must be used to signal labels. Perform the following steps to configure PE-to-CE MPLS forwarding and signaling with LDP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **mpls ip**

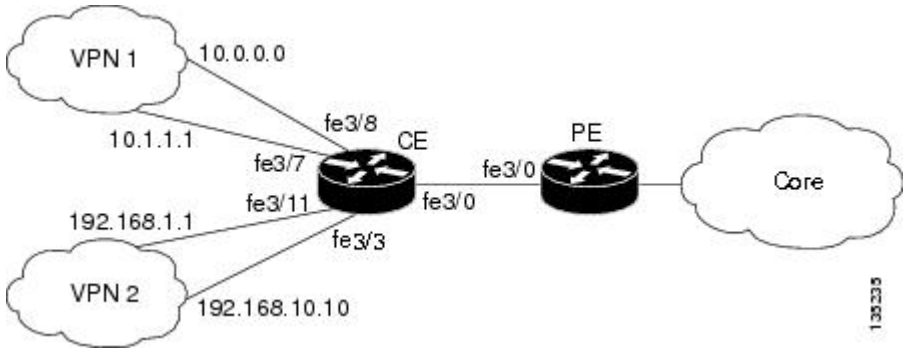
DETAILED STEPS

	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	**interface** *type number* **Example:** `Router(config)# interface fastethernet3/0.10`	Enters subinterface configuration mode for the interface associated with the VRF. • The interface can be a routed port or an SVI.
Step 4	**mpls ip** **Example:** `Router(config-subif)# mpls ip`	Enables MPLS forwarding of IPv4 packets along normally routed paths for this interface.

Configuration Examples for MPLS Multi-VRF

The figure below shows an example MPLS Multi-VRF configuration.



- Example Configuring MPLS Multi-VRF on the PE Router, page 154
- Example Configuring MPLS Multi-VRF on the CE Router, page 155

Example Configuring MPLS Multi-VRF on the PE Router

Configuring VRFs

```
configure terminal
ip vrf v1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
 exit
ip vrf v2
 rd 100:2
 route-target export 100:2
 route-target import 100:2
 exit
```

Configuring PE-to-CE Connections Using BGP for Both Routing and Label Exchange

```
router bgp 100
 address-family ipv4 vrf v2
  neighbor 10.0.0.8 remote-as 800
  neighbor 10.0.0.8 send-label
  neighbor 10.0.0.8 activate
  exit
 address-family ipv4 vrf vl
  neighbor 10.0.0.8 remote-as 800
  neighbor 10.0.0.8 send-label
  neighbor 10.0.0.8 activate
  end
configure terminal
 interface fastethernet3/0.10
  ip vrf forwarding v1
  ip address 10.0.0.3 255.255.255.0
  mpls bgp forwarding
  exit
 interface fastethernet3/0.20
  ip vrf forwarding v2
  ip address 10.0.0.3 255.255.255.0
  mpls bgp forwarding
  exit
```

Configuring PE-to-CE Connections Using OSPF for Routing and LDP for Label Exchange

```
router ospf 100 vrf v1
 network 10.0.0.0 255.255.255.0 area 0
 exit
router ospf 101 vrf v2
 network 10.0.0.0 255.255.255.0 area 0
 exit
interface fastethernet3/0.10
 ip vrf forwarding v1
 ip address 10.0.0.3 255.255.255.0
 mpls ip
 exit
interface fastethernet3/0.20
 ip vrf forwarding v2
 ip address 10.0.0.3 255.255.255.0
 mpls ip
 exit
```

Example Configuring MPLS Multi-VRF on the CE Router

Configuring VRFs

```
configure terminal
 ip routing
 ip vrf v11
  rd 800:1
  route-target export 800:1
  route-target import 800:1
  exit
 ip vrf v12
  rd 800:2
  route-target export 800:2
  route-target import 800:2
  exit
```

Configuring CE Router VPN Connections

```
interface fastethernet3/8
ip vrf forwarding v11
```

```
ip address 10.0.0.8 255.255.255.0
exit
interface fastethernet3/11
ip vrf forwarding v12
ip address 10.0.0.8 255.255.255.0
exit
  router ospf 1 vrf v11
 network 10.0.0.0 255.255.255.0 area 0
 network 10.0.0.0 255.255.255.0 area 0
 exit
  router ospf 2 vrf v12
 network 10.0.0.0 255.255.255.0 area 0
 network 10.0.0.0 255.255.255.0 area 0
 exit
```

Note If BGP is used for routing between the PE and CE routers, the BGP-learned routes from the PE router can be redistributed into OSPF using the commands in the following example.

```
  router ospf 1 vrf v11
 redistribute bgp 800 subnets
 exit
  router ospf 2 vrf v12
 redistribute bgp 800 subnets
 exit
```

Configuring PE-to-CE Connections Using BGP for Both Routing and Label Exchange

```
  router bgp 800
 address-family ipv4 vrf v12
neighbor 10.0.0.3 remote-as 100
neighbor 10.0.0.3 send-label
neighbor 10.0.0.3 activate
 exit
 address-family ipv4 vrf vl1
neighbor 10.0.0.3 remote-as 100
neighbor 10.0.0.3 send-label
neighbor 10.0.0.3 activate
 end
  interface fastethernet3/0.10
  ip vrf forwarding v11
  ip address 10.0.0.8 255.255.255.0
  mpls bgp forwarding
  exit
  interface fastethernet3/0.20
  ip vrf forwarding v12
  ip address 10.0.0.8 255.255.255.0
  mpls bgp forwarding
  exit
```

Configuring PE-to-CE Connections Using OSPF for Routing and LDP for Label Exchange

```
  router ospf 1 vrf v11
 network 10.0.0.0 255.255.255.0 area 0
 exit
  router ospf 2 vrf v12
 network 10.0.0.0 255.255.255.0 area 0
 exit
  interface fastethernet3/0.10
  ip vrf forwarding v11
  ip address 10.0.0.3 255.255.255.0
  mpls ip
  exit
  interface fastethernet3/0.20
  ip vrf forwarding v12
  ip address 10.0.0.3 255.255.255.0
```

```
mpls ip
exit
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Description of commands associated with MPLS and MPLS application	*Cisco IOS Multiprotocol Label Switching Command Reference*
OSPF with Multi-VRF	*OSPF Support for Multi-VRF in CE Routers*

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 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 Multi-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 11 Feature Information for MPLS Multi-VRF

Feature Name	Releases	Feature Information
MPLS Multi-VRF	12.1(11)EA1 12.1(20)EW 12.2(4)T 12.2(8)YN 12.2(18)SXD 12.2(25)EWA 12.2(28)SB	The MPLS Multi-VRF feature allows you to configure and maintain more than one instance of a routing and forwarding table within the same CE router. In Cisco IOS Release 12.1(11)EA1, the Multi-VRF feature was introduced. The feature was integrated into Cisco IOS Release 12.1(20)EW. The feature was integrated into Cisco IOS Release 12.2(4)T. The feature was integrated into Cisco IOS Release 12.2(8)YN. The feature was integrated into Cisco IOS Release 12.2(18)SXD. The feature was integrated into Cisco IOS Release 12.2(25)EWA. Multiprotocol Label Switching support was added in Cisco IOS Release 12.2(28)SB.

BGP Best External

The BGP Best External feature provides the network with a backup external route to avoid loss of connectivity of the primary external route. The BGP Best External feature advertises the most preferred route among those received from external neighbors as a backup route. This feature is beneficial in active-backup topologies, where service providers use routing policies that cause a border router to choose a path received over an internal BGP (iBGP) session (of another border router) as the best path for a prefix even if it has an external BGP (eBGP) learned path. This active-backup topology defines one exit or egress point for the prefix in the autonomous system and uses the other points as backups if the primary link or eBGP peering is unavailable. The policy causes the border router to hide the paths learned over its eBGP sessions from the autonomous system because it does not advertise any path for such prefixes. To cope with this situation, some routers advertise one externally learned path called the best external path.

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 for BGP Best External, page 171.

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

Contents

Prerequisites for BGP Best External

- The Bidirectional Forwarding Detection (BFD) protocol must be enabled to quickly detect link failures.
- Ensure that the BGP and the Multiprotocol Label Switching (MPLS) network is up and running with the customer site connected to the provider site by more than one path (multihomed).
- The backup path must have a unique next hop that is not the same as the next hop of the best path.
- BGP must support lossless switchover between operational paths.

Restrictions for BGP Best External

- The BGP Best External feature will not install a backup path if BGP Multipath is installed and a multipath exists in the BGP table. One of the multipaths automatically acts as a backup for the other paths.
- The BGP Best External feature is not supported with the following features:
  - MPLS VPN Carrier Supporting Carrier
  - MPLS VPN Inter-Autonomous Systems, option B
  - MPLS VPN Per Virtual Routing and Forwarding (VRF) Label
- The BGP Best External feature cannot be configured with Multicast or L2VPN VRF address families.
- The BGP Best External feature cannot be configured on route reflectors.
- The BGP Best External feature does not support NSF/SSO. However, ISSU is supported if both Route Processors have the BGP Best External feature configured.
- The BGP Best External feature can only be configured on VPNv4, VPNv6, IPv4 VRF, and IPv6 VRF address families.
- When you configure the BGP Best External feature using the **bgp advertise-best-external** command, you need not enable the BGP PIC feature with the **bgp additional-paths install** command. The BGP PIC feature is automatically enabled by the BGP Best External feature.
- When you configure the BGP Best External feature, it will override the functionality of the MPLS VPN--BGP Local Convergence feature. However, you do not have to remove the **protection local-prefixes** command from the configuration.

Information About BGP Best External

BGP Best External Overview

Service providers use routing policies that cause a border router to choose a path received over an iBGP session (of another border router) as the best path for a prefix even if it has an eBGP learned path. This practice is popularly known as active-backup topology and is done to define one exit or egress point for the

prefix in the autonomous system and to use the other points as backups if the primary link or eBGP peering is unavailable.

The policy, though beneficial, causes the border router to hide the paths learned over its eBGP sessions from the autonomous system because the border router does not advertise any path for such prefixes. To cope with this situation, some routers advertise one externally learned path called the best external path. The best external behavior causes the BGP selection process to select two paths to every destination:

- The best path is selected from the complete set of routes known to that destination.
- The best external path is selected from the set of routes received from its external peers.

BGP advertises the best path to external peers. Instead of withdrawing the best path from its internal peers when it selects an iBGP path as the best path, BGP advertises the best external path to the internal peers.

The BGP Best External feature is an essential component of the Prefix-Independent Convergence (PIC) edge for both Internet access and MPLS VPN scenarios and makes alternate paths available in the network in the active-backup topology.

What the Best External Route Means

The BGP Best External feature uses a "best external route" as a backup path, which, according to *draft-marques-idr-best-external*, is the most preferred route among those received from external neighbors. The most preferred route from external neighbors can be the following:

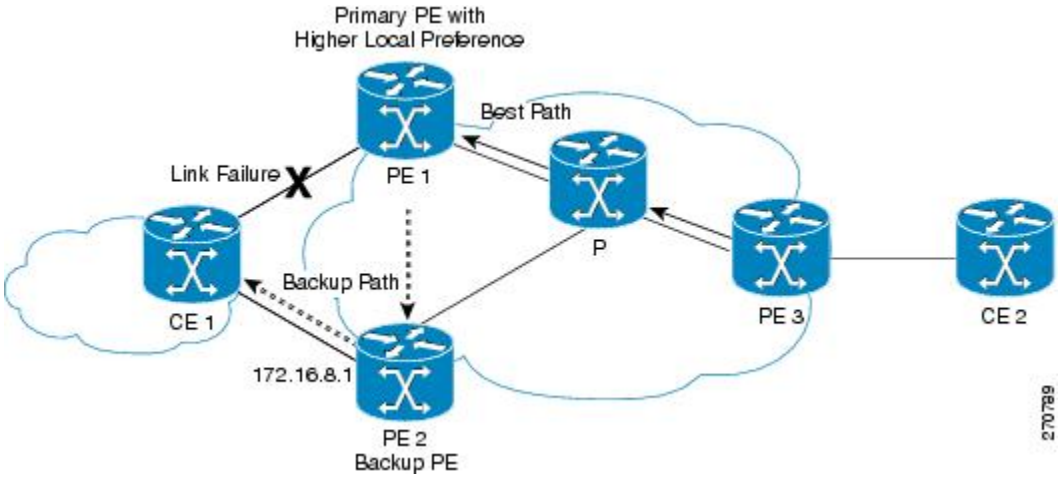
- Two routers in different clusters that have an iBGP session between them.
- Two routers in different autonomous systems of a confederation that have an eBGP session between them.

The best external route might be different from the best route installed in the routing information base (RIB). The best route could be an internal route. By allowing the best external route to be advertised and stored, in addition to the best route, networks gain faster restoration of connectivity by providing additional paths that may be used if the primary path fails.

How the BGP Best External Feature Works

The BGP Best External feature is based on Internet Engineering Task Force (IETF) draft-marques-idr-best-external.txt. The BGP Best External feature advertises a best external route to its internal peers as a backup route. The backup route is stored in the RIB and Cisco Express Forwarding. If the primary path fails, the BGP PIC functionality enables the best external path to take over, enabling faster restoration of connectivity.

Figure 9 MPLS VPN: Best External at the Edge of MPLS VPN



How the BGP Best External Feature Works, page 163 shows an MPLS VPN using the BGP Best External feature. The network includes the following components:

- eBGP sessions exist between the provider edge (PE) and customer edge (CE) routers.
- PE1 is the primary router and has a higher local preference setting.
- Traffic from CE2 uses PE1 to reach router CE1.
- PE1 has two paths to reach CE1.
- CE1 is dual-homed with PE1 and PE2.
- PE1 is the primary path and PE2 is the backup path.

In How the BGP Best External Feature Works, page 163, traffic in the MPLS cloud flows through PE1 to reach CE1. Therefore, PE2 uses PE1 as the best path and PE2 as the backup path.

PE1 and PE2 are configured with the BGP Best External feature. BGP computes both the best path (the PE1-CE1 link) and a backup path (PE2) and installs both paths into the RIB and Cisco Express Forwarding. The best external path (PE2) is advertised to the peer routers, in addition to the best path.

When Cisco Express Forwarding detects a link failure on the PE1-CE1 link, Cisco Express Forwarding immediately switches to the backup path PE2. Traffic is quickly rerouted due to local Fast Convergence in Cisco Express Forwarding using the backup path. Thus, traffic loss is minimized and fast convergence is achieved.

Configuration Modes for Enabling BGP Best External

You can enable the BGP Best External feature in different modes, each of which protects VRFs in its own way:

- If you issue the **bgp advertise-best-external** command in VPNv4 address family configuration mode, it applies to all IPv4 VRFs. If you issue the command in this mode, you need not issue it for specific VRFs.
- If you issue the **bgp advertise-best-external** command in IPv4 address family configuration mode, it applies only that VRF.

How to Configure BGP Best External

Enabling the BGP Best External Feature

Perform the following task to enable the BGP Best External feature. This task shows how to configure the BGP Best External feature in either IPv4 or VPNv4 address family. In VPNv4 address family configuration mode, the BGP Best External feature applies to all IPv4 VRFs; you do not have to configure it for specific VRFs. If you issue the **bgp advertise-best-external** command in IPv4 VRF address family configuration mode, the BGP Best External feature applies only that VRF.

- Configure the MPLS VPN and verify that it is working properly before configuring the BGP Best External feature. See "Configuring MPLS Layer 3 VPNs" for more information.
- Configure multiprotocol VRFs, which allow you to share route-target policies (import and export) between IPv4 and IPv6 or configure separate route-target policies for IPv4 and IPv6 VPNs. For

information about configuring multiprotocol VRFs, see "MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs".

- Ensure that the CE router is connected to the network by at least two paths.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. Do one of the following:
   - **address-family ipv4** [**unicast** | **vrf** *vrf-name*]
   - or
   - **address-family vpnv4** [**unicast**]
5. **bgp advertise-best-external**
6. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
7. **neighbor** *ip-address* **activate**
8. **neighbor** *ip-address* **fall-over** [**b fd**|**route-map** *map-name*]
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** *autonomous-system-number* **Example:** `Router(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose

Step 4	Do one of the following:
	• **address-family ipv4** [**unicast** | **vrf** *vrf-name*]
	• or
	• **address-family vpnv4** [**unicast**]
	Example:
	`Router(config-router)# address-family ipv4 unicast`
	Example:
	`Router(config-router)# address-family vpnv4`
		Specifies the IPv4 or VPNv4 address family and enters address family configuration mode.
		• The **unicast**keyword specifies the IPv4 or VPNv4 unicast address family.
		• The **vrf**keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.

Step 5	**bgp advertise-best-external** **Example:** `Router(config-router-af)# bgp advertise-best-external`	Calculates and uses an external backup path and installs it into the RIB and Cisco Express Forwarding.
Step 6	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • By default, neighbors that are defined using the **neighbor remote-as**command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, neighbors must also be activated using the **neighbor activate**command in address family configuration mode for the other prefix types.
Step 7	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local router.

	Command or Action	Purpose
Step 8	**neighbor** *ip-address* **fall-over** [**b fd**|**route-map** *map-name*] **Example:** `Router(config-router-af)# neighbor 192.168.1.1 fall-over bfd`	Configures the BGP peering to use fast session deactivation and enables BFD protocol support for failover. • BGP will remove all routes learned through this peer if the session is deactivated.
Step 9	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits address family configuration mode and returns to privileged EXEC mode.

Verifying the BGP Best External Feature

Perform the following task to verify that the BGP Best External feature is configured correctly.

SUMMARY STEPS

1. **enable**
2. **show vrf detail**
3. **show ip bgp ipv4** {**mdt** {**all** | **rd** | **vrf**} | **multicast** | **tunnel**| **unicast**} or**show ip bgp vpnv4**{**all**| **rd** *route-distinguisher* | **vrf** *vrf-name*}[**rib-failure**][*ip-prefix/length* [**longer-prefixes**]][*network-address*[*mask*][**longer-prefixes**]][**cidr-only**][**community**][*community-list*][**dampened-paths**][**filter-list**] [**flap-statistics**][**inconsistent-as**][**neighbors**][**paths** [**line**]][**peer-group**][**quote-regexp**][**regexp**] [**summary**][**labels**]
4. **show bgp vpnv4 unicast vrf** *vrf-name ip-address*
5. **show ip route vrf** *vrf-name* **repair-paths** *ip-address*
6. **show ip cef vrf** *vrf-name ip-address* **detail**

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 vrf detail**

Use this command to verify that the BGP Best External feature is enabled. The following **show vrf detail** command output shows that the BGP Best External feature is enabled.

Example:

```
Router# show vrf detail
VRF test1 (VRF Id = 1); default RD 400:1; default VPNID <not set>
  Interfaces:
    Se4/0
Address family ipv4 (Table ID = 1 (0x1)):
  Export VPN route-target communities
    RT:100:1                RT:200:1                RT:300:1
    RT:400:1
  Import VPN route-target communities
    RT:100:1                RT:200:1                RT:300:1
    RT:400:1
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix

 Prefix protection with additional path enabled
Address family ipv6 not active.
```

Step 3 **show ip bgp ipv4** {**mdt** {**all** | **rd** | **vrf**} | **multicast** | **tunnel**| **unicast**} or**show ip bgp vpnv4**{**all**| **rd** *route-distinguisher* | **vrf** *vrf-name*}[**rib-failure**][*ip-prefix/length* [**longer-prefixes**]][*network-address*[*mask*][**longer-prefixes**]][**cidr-only**][**community**][*community-list*][**dampened-paths**][**filter-list**] [**flap-statistics**][**inconsistent-as**] [**neighbors**][**paths** [**line**]][**peer-group**][**quote-regexp**][**regexp**] [**summary**][**labels**]

Use this command to verify that the best external route is advertised. In the command output, the code b indicates a backup path and the code x designates the best external path.

Example:

```
Router# show ip bgp vpnv4 all
BGP table version is 1104964, local router ID is 10.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, multipath,
b backup-path, x best-external
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 11:12 (default for vrf blue)
*>i1.0.0.1/32       10.10.3.3                0     200     0 1 ?
* i                 10.10.3.3                0     200     0 1 ?
*                   10.0.0.1                               0 1 ?
*bx                 10.0.0.1                 0             0 1 ?
*                   10.0.0.1                               0 1 ?
```

Step 4 **show bgp vpnv4 unicast vrf** *vrf-name ip-address*

Use this command to verify that the best external route is advertised.

Example:

```
Router# show bgp vpnv4 unicast vrf vpn1 10.10.10.10
BGP routing table entry for 10:10:10.10.10.10/32, version 10
Paths: (2 available, best #1, table vpn1)
  Advertise-best-external
  Advertised to update-groups:
    1         2
  200
    10.6.6.6 (metric 21) from 10.6.6.6 (10.6.6.6)
    Origin incomplete, metric 0, localpref 200, valid, internal, best
    Extended Community: RT:1:1
    mpls labels in/out 23/23
  200
    10.1.2.1 from 10.1.2.1 (10.1.1.1)
    Origin incomplete, metric 0, localpref 100, valid,
external, backup/repair,     advertise-best-external
```

```
    Extended Community: RT:1:1 , recursive-via-connected
    mpls labels in/out 23/nolabel
```

Step 5 **show ip route vrf** *vrf-name* **repair-paths** *ip-address*

Use this command to display the repair route.

Example:

```
Router# show ip route vrf vpn1 repair-paths
Routing Table: vpn1
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP
       + - replicated route, % - next hop override
Gateway of last resort is not set
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
B        10.1.1.0/24 [200/0] via 10.6.6.6, 00:38:33
                     [RPR][200/0] via 10.1.2.1, 00:38:33
B        10.1.1.1/32 [200/0] via 10.6.6.6, 00:38:33
                     [RPR][200/0] via 10.1.2.1, 00:38:33
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.1.2.0/24 is directly connected, Ethernet0/0
L        10.1.2.2/32 is directly connected, Ethernet0/0
B        10.1.6.0/24 [200/0] via 10.6.6.6, 00:38:33
                      [RPR][200/0] via 10.1.2.1, 00:38:33
```

Step 6 **show ip cef vrf** *vrf-name ip-address* **detail**

Use this command to display the best external route.

Example:

```
Router# show ip cef vrf test 10.71.8.164 detail

10.71.8.164/30, epoch 0, flags rib defined all labels
 recursive via 10.249.0.102 label 35
   nexthop 10.249.246.101 Ethernet0/0 label 25
 recursive via 10.249.0.104 label 28,
repair
   nexthop 10.249.246.101 Ethernet0/0 label 24
```

Configuration Examples for BGP Best External

Example Configuring the BGP Best External Feature

The following example shows how to configure the BGP Best External feature in VPNv4 mode:

```
vrf definition test1
 rd 400:1
 route-target export 100:1
 route-target export 200:1
```

```
 route-target export 300:1
 route-target export 400:1
 route-target import 100:1
 route-target import 200:1
 route-target import 300:1
 route-target import 400:1
 address-family ipv4
 exit-address-family
 exit
!
interface Ethernet1/0
 vrf forwarding test1
 ip address 10.0.0.1 255.0.0.0
 exit
!
router bgp 64500
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.5.5.5 remote-as 64500
 neighbor 10.5.5.5 update-source Loopback0
 neighbor 10.6.6.6 remote-as 64500
 neighbor 10.6.6.6 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4

bgp advertise-best-external
  neighbor 10.5.5.5 activate
  neighbor 10.5.5.5 send-community extended
  neighbor 10.6.6.6 activate
  neighbor 10.6.6.6 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf test1
  no synchronization
bgp recursion host
  neighbor 192.168.13.2 remote-as 64511
  neighbor 192.168.13.2 fall-over bfd
  neighbor 192.168.13.2 activate
  neighbor 192.168.13.2 as-override
 exit-address-family
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: BGP Command Reference*
Basic MPLS VPNs	"Configuring MPLS Layer 3 VPNs"
Multiprotocol VRFs	"MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs"
A failover feature that creates a new path after a link or node failure	"MPLS VPN--BGP Local Convergence"

Standards

Standard	Title
draft-marques-idr-best-external	BGP Best External, Advertisement of the best external route to iBGP

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 2547	*BGP/MPLS VPNs*
RFC 1771	A Border Gateway Protocol 4 (BGP-4)

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 BGP Best External

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 12 Feature Information for BGP Best External

Feature Name	Releases	Feature Information
BGP Best External	12.2(33)SRE	The BGP Best External feature provides the network with a backup external route to avoid loss of connectivity of the primary external route. This feature advertises the most preferred route among those received from external neighbors as a backup route. In Cisco IOS Release12.2(33)SRE, this feature was introduced. The following sections provide information about this feature: The following commands were introduced or modified: **bgp advertise-best-external**, **bgp recursion host**, **show ip bgp**, **show ip bgp vpnv4**, **show ip cef**, **show ip cef vrf**, **show ip route**, **show ip route vrf**.

BGP PIC Edge for IP and MPLS-VPN

First Published: November 20, 2009

Last Updated: March 31, 2011

The BGP PIC Edge for IP and MPLS-VPN feature improves BGP convergence after a network failure. This convergence is applicable to both core and edge failures and can be used in both IP and MPLS networks. The BGP PIC Edge for IP and MPLS-VPN feature creates and stores a backup/alternate path in the routing information base (RIB), forwarding information base (FIB), and Cisco Express Forwarding so that when a failure is detected, the backup/alternate path can immediately take over, thus enabling fast failover.

Note In this document, the BGP PIC Edge for IP and MPLS-VPN feature is called BGP PIC.

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 for BGP PIC, page 190.

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

Contents

Prerequisites for BGP PIC

- Ensure that the Border Gateway Protocol (BGP) and the IP or Multiprotocol Label Switching (MPLS) network is up and running with the customer site connected to the provider site by more than one path (multihomed).
- Ensure that the backup/alternate path has a unique next hop that is not the same as the next hop of the best path.
- Enable the Bidirectional Forwarding Detection (BFD) protocol to quickly detect link failures of directly connected neighbors.

Restrictions for BGP PIC

- With BGP Multipath, the BGP Prefix-Independent Convergence (PIC) feature is already supported.
- In MPLS VPNs, the BGP PIC feature is not supported with MPLS VPN Inter-Autonomous Systems Option B.
- The BGP PIC feature supports prefixes only for IPv4, IPv6, VPNv4, and VPNv6 address families.
- The BGP PIC feature cannot be configured with Multicast or L2VPN Virtual Routing and Forwarding (VRF) address families.
- If the route reflector is only in the control plane, then you do not need BGP PIC, because BGP PIC addresses data plane convergence.
- When two PE routers become each other's backup/alternate path to a CE router, traffic might loop if the CE router fails. Neither router will reach the CE router, and traffic will continue to be forwarded between the PE routers until the time-to-live (TTL) timer expires.
- The BGP PIC feature does not support Nonstop Forwarding with Stateful Switchover (NSF/SSO). However, ISSU is supported if both Route Processors have the BGP PIC feature configured.
- The BGP PIC feature solves the traffic forwarding only for a single network failure at both the edge and the core.
- The BGP PIC feature does not work with the BGP Best External feature. If you try to configure the BGP PIC feature after configuring the BGP Best External feature, you receive an error.

Information About BGP PIC

- Cisco Express Forwarding Recursion, page 182

Benefits of the BGP PIC Edge for IP and MPLS-VPN Feature

- An additional path for failover allows faster restoration of connectivity if a primary path is invalid or withdrawn.
- Reduction of traffic loss.
- Constant convergence time so that the switching time is the same for all prefixes.

How BGP Converges Under Normal Circumstances

Under normal circumstances, BGP can take several seconds to a few minutes to converge after a network change. At a high level, BGP goes through the following process:

1. BGP learns of failures through either Interior Gateway Protocol (IGP) or BFD events or interface events.
2. BGP withdraws the routes from the routing information base (RIB), and the RIB withdraws the routes from the forwarding information base (FIB) and distributed FIB (dFIB). This process clears the data path for the affected prefixes.
3. BGP sends withdraw messages to its neighbors.
4. BGP calculates the next best path to the affected prefixes.
5. BGP inserts the next best path for affected prefixes into the RIB, and the RIB installs them in the FIB and dFIB.

This process takes a few seconds or a few minutes to complete, depending on the latency of the network, the convergence time across the network, and the local load on the devices. The data plane converges only after the control plane converges.

How BGP PIC Improves Convergence

The BGP PIC functionality is achieved by an additional functionality in the BGP, RIB, Cisco Express Forwarding, and MPLS.

- BGP Functionality

BGP PIC affects prefixes under IPv4 and VPNv4 address families. For those prefixes, BGP calculates an additional second best path, along with the primary best path. (The second best path is called the backup/alternate path.) BGP installs the best and backup/alternate paths for the affected prefixes into the BGP RIB. The backup/alternate path provides a fast reroute mechanism to counter a singular network failure. BGP also includes the alternate/backup path in its application programming interface (API) to the IP RIB.

- RIB Functionality

For BGP PIC, RIB installs an alternate path per route if one is available. With the BGP PIC functionality, if the RIB selects a BGP route containing a backup/alternate path, it installs the backup/alternate path with the best path. The RIB also includes the alternate path in its API with the FIB.

- Cisco Express Forwarding Functionality

With BGP PIC, Cisco Express Forwarding stores an alternate path per prefix. When the primary path goes down, Cisco Express Forwarding searches for the backup/alternate path in a prefix independent manner. Cisco Express Forwarding also listens to BFD events to rapidly detect local failures.

- MPLS Functionality

MPLS Forwarding is similar to Cisco Express Forwarding, in that it stores alternate paths and switches to an the alternate path if the primary path goes down.

When the BGP PIC feature is enabled, BGP calculates a backup/alternate path per prefix and installs it into BGP RIB, IP RIB, and FIB. This improves convergence after a network failure. There are two types of network failures that the BGP PIC feature detects:

- Core node/link failure (internal Border Gateway Protocol [iBGP] node failure): If a PE node/link fails, then the failure is detected through IGP convergence. IGP conveys the failure through the RIB to the FIB.
- Local link/immediate neighbor node failure (external Border Gateway Protocol [eBGP] node/link failure): To detect a local link failure or eBGP single-hop peer node failure in less than a second, you must enable BFD. Cisco Express Forwarding looks for BFD events to detect a failure of an eBGP single-hop peer.

Convergence in the Data Plane

Upon detection of a failure, Cisco Express Forwarding detects the alternate next hop for all prefixes affected by the failure. The data plane convergence is achieved in subseconds depending on whether the BGP PIC implementation exists in the software or hardware.

Convergence in the Control Plane

Upon detection of failure, BGP learns about the failure through IGP convergence or BFD events and sends withdraw messages for the prefixes, recalculating the best and backup/alternate paths, and advertising the next best path across the network.

- BGP Fast Reroute's Role in the BGP PIC Feature, page 176

BGP Fast Reroute's Role in the BGP PIC Feature

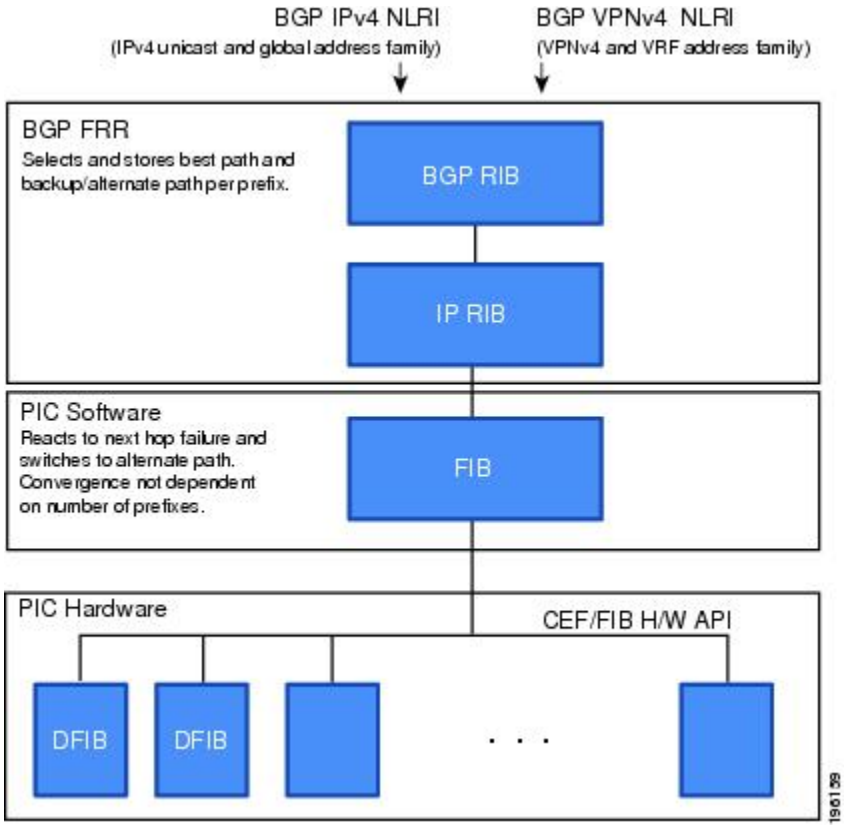
BGP Fast Reroute (FRR) provides a best path and a backup/alternate path in BGP, RIB, and Cisco Express Forwarding. BGP FRR provides a very fast reroute mechanism into the RIB and Cisco Express Forwarding on the backup BGP next hop to reach a destination when the current best path is not available.

BGP FRR precomputes a second best path in BGP and gives it to the RIB and Cisco Express Forwarding as a backup/alternate path, and Cisco Express Forwarding programs it into line cards.

Therefore, BGP FRR sets up the best path and backup/alternate path. The BGP PIC feature provides the ability for Cisco Express Forwarding to quickly switch the traffic to the other egress ports if the current

next hop or the link to this next hop goes down. This is illustrated in GUID-EA7473E8-C436-4913-94F1-AF34801AD8B20.

Figure 10 BGP PIC Edge and BGP FRR



How a Failure Is Detected

A failure in the iBGP (remote) peer is detected by IGP; it may take a few seconds to detect the failure. Convergence can occur in subseconds or seconds, depending on whether PIC is enabled on the line cards.

If the failure is with directly connected neighbors (eBGP), and if you use BFD to detect when a neighbor has gone down, the detection happens within a subsecond and the convergence can occur in subseconds or seconds, depending on whether PIC is enabled on the line cards.

How BGP PIC Achieves Subsecond Convergence

The BGP PIC feature works at the Cisco Express Forwarding level, and Cisco Express Forwarding can be processed in both hardware line cards and in the software.

- For platforms that support Cisco Express Forwarding processing in the line cards, the BGP PIC feature can converge in subseconds. The Cisco 7600 router and Cisco 10000 router supports Cisco Express Forwarding processing in the line cards and in the software, and thus can attain subsecond convergence.
- For platforms that do not use Cisco Express Forwarding in hardware line cards, Cisco Express Forwarding is achieved in the software. The BGP PIC feature will work with the Cisco Express

Forwarding through the software and achieve convergence within seconds. The Cisco 7200 router supports Cisco Express Forwarding in the software and thus can achieve convergence in seconds rather than milliseconds.

How BGP PIC Improves Upon the Functionality of MPLS VPN--BGP Local Convergence

The BGP PIC feature is an enhancement to the MPLS VPN--BGP Local Convergence feature, which provides a failover mechanism that recalculates the best path and installs the new path in forwarding after a link failure. The feature maintains the local label for 5 minutes to ensure that the traffic uses the backup/alternate path, thus minimizing traffic loss.

The BGP PIC feature improves the LoC time to under a second by calculating a backup/alternate path in advance. When a link failure occurs, the traffic is sent to the backup/alternate path.

When you configure the BGP PIC feature, it will override the functionality of the MPLS VPN--BGP Local Convergence feature. You do not have to remove the **protection local-prefixes** command from the configuration.

Configuration Modes for Enabling BGP PIC

Because many service provider networks contain many VRFs, the BGP PIC feature allows you to configure the BGP PIC feature for all VRFs at once.

- VPNv4 address family configuration mode protects all the VRFs.
- VRF-IPv4 address family configuration mode protects only IPv4 VRFs.
- Router configuration mode protects prefixes in the global routing table.

BGP PIC Scenarios

The following scenarios explain how you can configure the BGP PIC functionality to achieve fast convergence:

- IP PE-CE Link and Node Protection on the CE Side (Dual PEs), page 178
- IP PE-CE Link and Node Protection on the CE Side (Dual CEs and Dual PE Primary and Backup Nodes), page 179
- IP MPLS PE-CE Link Protection for the Primary or Backup-Alternate Path, page 180
- IP MPLS PE-CE Node Protection for Primary or Backup-Alternate Path, page 181

IP PE-CE Link and Node Protection on the CE Side (Dual PEs)

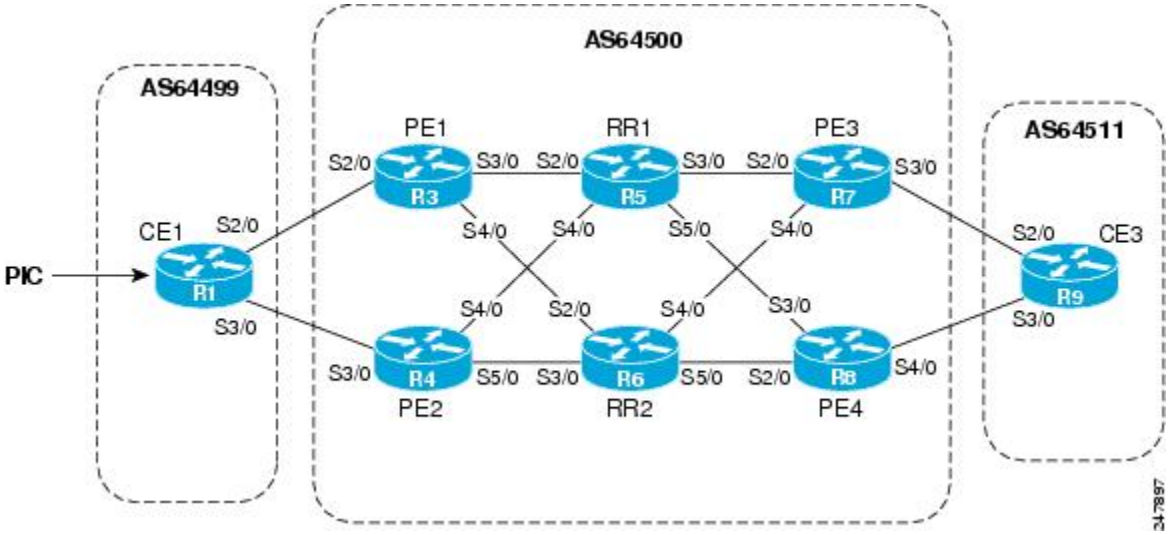
GUID-A26C9E51-ADA4-4FAF-B50F-6B056638A7A96 shows a network that uses the BGP PIC feature. The network includes the following components:

- eBGP sessions exist between the PE and CE routers.
- Traffic from CE1 uses PE1 to reach network 192.168.9.0/24 through router CE3.
- CE1 has two paths:
  - PE1 as the primary path.
  - PE2 as the backup/alternate path.

CE1 is configured with the BGP PIC feature. BGP computes PE1 as the best path and PE2 as the backup/alternate path and installs both routes into the RIB and Cisco Express Forwarding plane. When the CE1-

PE1 link goes down, Cisco Express Forwarding detects the link failure and points the forwarding object to the backup/alternate path. Traffic is quickly rerouted due to local fast convergence in Cisco Express Forwarding.

Figure 11 Using BGP PIC to Protect the PE-CE Link



IP PE-CE Link and Node Protection on the CE Side (Dual CEs and Dual PE Primary and Backup Nodes)

GUID-545AAEB0-E65E-4B56-911C-CF58851684926 shows a network that uses the BGP PIC feature on CE1. The network includes the following components:

- eBGP sessions exist between the PE and CE routers.
- Traffic from CE1 uses PE1 to reach network 192.168.9.0/24 through router CE3.
- CE1 has two paths:
  - PE1 as the primary path.
  - PE2 as the backup/alternate path.
- An iBGP session exists between the CE1 and CE2 routers.

In this example, CE1 and CE2 are configured with the BGP PIC feature. BGP computes PE1 as the best path and PE2 as the backup/alternate path and installs both the routes into the RIB and Cisco Express Forwarding plane.

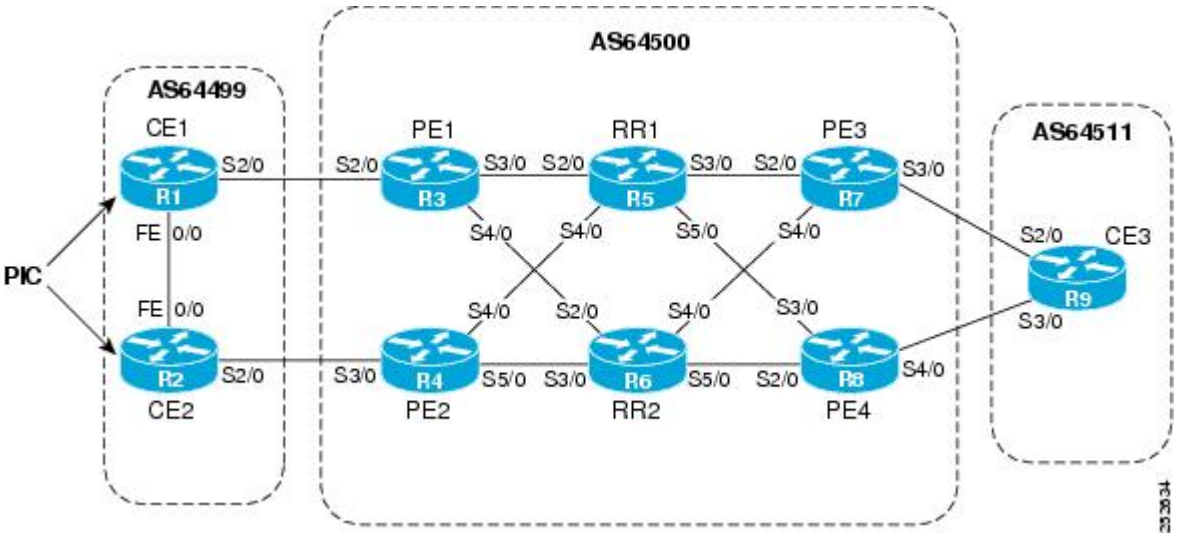
There should not be any policies set on CE1 and CE2 for the eBGP peers PE1 and PE2. Both CE routers must point to the eBGP route as next hop. On CE1, the next hop to reach CE3 is through PE1, so PE1 is the best path to reach CE3. On CE2, the best path to reach CE3 is PE2. CE2 advertises itself as the next hop to CE1, and CE1 does the same to CE2. As a result, CE1 has two paths for the specific prefix and it usually selects the directly connected eBGP path over the iBGP path according to the best path selection rules. Similarly, CE2 has two paths--an eBGP path through PE2 and an iBGP path through CE1-PE1.

When the CE1-PE1 link goes down, Cisco Express Forwarding detects the link failure and points the forwarding object to the backup/alternate node CE2. Traffic is quickly rerouted due to local fast convergence in Cisco Express Forwarding.

If the CE1-PE1 link or PE1 goes down and BGP PIC is enabled on CE1, BGP recomputes the best path, removing the next hop PE1 from RIB and reinstalling CE2 as the next hop into the RIB and Cisco Express

Forwarding. CE1 automatically gets a backup/alternate repair path into Cisco Express Forwarding and the traffic loss during forwarding is now in subseconds, thereby achieving fast convergence.

Figure 12 Using BGP PIC in a Dual CE, Dual PE Network



IP MPLS PE-CE Link Protection for the Primary or Backup-Alternate Path

IP MPLS PE-CE Link Protection for the Primary or Backup-Alternate Path, page 180 shows a network that uses the BGP PIC feature on CE1 and CE2. The network includes the following components:

- eBGP sessions exist between the PE and CE routers.
- The PE routers are VPNv4 iBGP peers with reflect routers in the MPLS network.
- Traffic from CE1 uses PE1 to reach the network 192.168.9.0/24 through router CE3.
- CE3 is dual-homed with PE3 and PE4.
- PE1 has two paths to reach CE3 from the reflect routers:
  - PE3 is the primary path with the next hop as a PE3 address.
  - PE4 is the backup/alternate path with the next hop as a PE4 address.

In this example, all the PE routers can be configured with the BGP PIC feature under IPv4 or VPNv4 address families.

For BGP PIC to work in BGP for PE-CE link protection, set the policies on PE3 and PE4 for prefixes received from CE3 so that one of the PE routers acts as the primary and the other as the backup/alternate. Usually, this is done using local preference and giving better local preference to PE3. In the MPLS cloud, traffic internally flows through PE3 to reach CE3. Thus, PE1 has PE3 as the best path and PE4 as the second path.

When the PE3-CE3 link goes down, Cisco Express Forwarding detects the link failure, and PE3 recomputes the best path, selects PE4 as the best path, and sends a withdraw message for the PE3 prefix to the reflect routers. Some of the traffic goes through PE3-PE4 until BGP installs PE4 as the best path route into the RIB and Cisco Express Forwarding. PE1 receives the withdraw, recomputes the best path, selects PE4 as the best path, and installs the routes into the RIB and Cisco Express Forwarding plane.

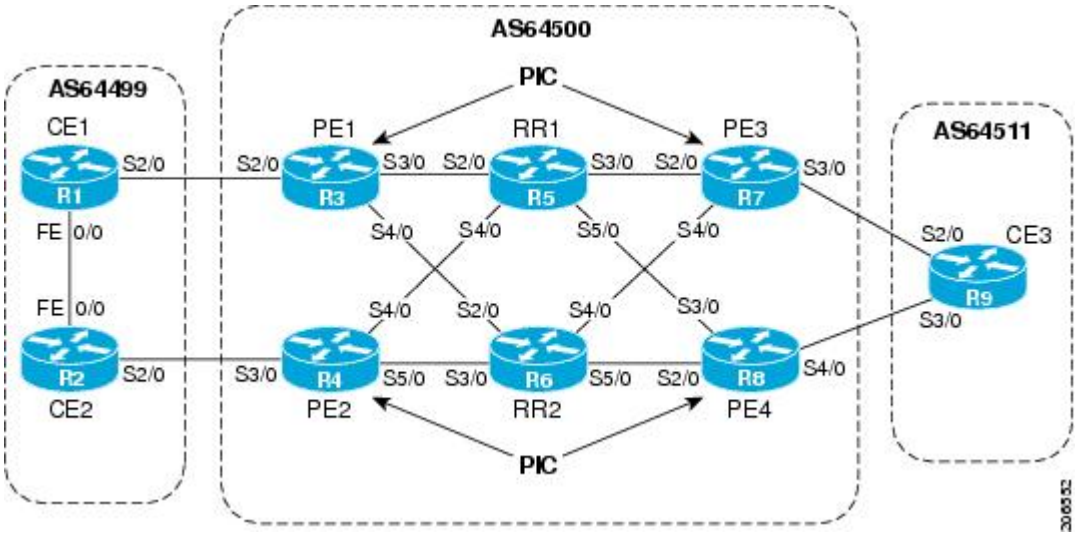
Thus, with BGP PIC enabled on PE3 and PE4, Cisco Express Forwarding detects the link failure and does in-place modification of the forwarding object to the backup/alternate node PE4 that already exists in Cisco

Express Forwarding. PE4 knows that the backup/alternate path is locally generated and routes the traffic to the egress port connected to CE3. This way, traffic loss is minimized and fast convergence is achieved.

IP MPLS PE-CE Node Protection for Primary or Backup-Alternate Path

IP MPLS PE-CE Node Protection for Primary or Backup-Alternate Path, page 181 shows a network that uses the BGP PIC feature on all the PE routers in an MPLS network.

Figure 13 Enabling BGP PIC on All PEs Routers in the MPLS Network



The network includes the following components:

- eBGP sessions exist between the PE and CE routers.
- The PE routers are VPNv4 iBGP peers with reflect routers in the MPLS network.
- Traffic from CE1 uses PE1 to reach the network 192.168.9.0/24 through router CE3.
- CE3 is dual-homed with PE3 and PE4.
- PE1 has two paths to reach CE3 from the reflect routers:
  - PE3 is the primary path with the next hop as a PE3 address.
  - PE4 is the backup/alternate path with the next hop as a PE4 address.

In this example, all the PE routers are configured with the BGP PIC feature under IPv4 and VPNv4 address families.

For BGP PIC to work in BGP for the PE-CE node protection, set the policies on PE3 and PE4 for the prefixes received from CE3 such that one of the PE routers acts as primary and the other as backup/alternate. Usually, this is done using local preference and giving better local preference to PE3. In the MPLS cloud, traffic internally flows through PE3 to reach CE3. So, PE1 has PE3 as the best path and PE4 as the second path.

When PE3 goes down, PE1 knows about the removal of the host prefix by IGPs in subseconds, recomputes the best path, selects PE4 as the best path, and installs the routes into the RIB and Cisco Express Forwarding plane. Normal BGP convergence will happen while BGP PIC is redirecting the traffic through PE4, and packets are not lost.

Thus, with BGP PIC enabled on PE3, Cisco Express Forwarding detects the node failure on PE3 and points the forwarding object to the backup/alternate node PE4. PE4 knows that the backup/alternate path is locally

generated and routes the traffic to the egress port using the backup/alternate path. This way, traffic loss is minimized.

No Local Policies Set on the PE Routers

PE1 and PE2 point to the eBGP CE paths as the next hop with no local policy. Each of the PE routers receives the other's path, and BGP calculates the backup/alternate path and installs it into Cisco Express Forwarding, along with its own eBGP path towards CE as the best path. The limitation of the MPLS PE-CE link and node protection solutions is that you cannot change BGP policies. They should work without the need for a best-external path.

Local Policies Set on the PE Routers

Whenever there is a local policy on the PE routers to select one of the PE routers as the primary path to reach the egress CE, the **bgp advertise-best-external** command is needed on the backup/alternate node PE3 to propagate the external CE routes with a backup/alternate label into the route reflectors and the far-end PE routers.

Cisco Express Forwarding Recursion

Recursion is the ability to find the next longest matching path when the primary path goes down.

When the BGP PIC feature is not installed, and if the next hop to a prefix fails, Cisco Express Forwarding finds the next path to reach the prefix by recursing through the FIB to find the next longest matching path to the prefix. This is useful if the next hop is multiple hops away and there is more than one way of reaching the next hop.

However, with the BGP PIC feature, you may want to disable Cisco Express Forwarding recursion for the following reasons:

- Recursion slows down convergence when Cisco Express Forwarding searches all the FIB entries.
- BGP PIC Edge already precomputes an alternate path, thus eliminating the need for Cisco Express Forwarding recursion.

When the BGP PIC functionality is enabled, Cisco Express Forwarding recursion is disabled by default for two conditions:

- For next hops learned with a /32 network mask (host routes)
- For next hops that are directly connected

For all other cases, Cisco Express Forwarding recursion is enabled.

As part of the BGP PIC functionality, you can issue the **bgp recursion host** command to disable or enable Cisco Express Forwarding recursion for BGP host routes.

To disable or enable Cisco Express Forwarding recursion for BGP directly connected next hops, you can issue the **disable-connected-check** command.

How to Configure BGP PIC

- Configuring BGP PIC, page 183

Configuring BGP PIC

Because many service provider networks contain many VRFs, the BGP PIC feature allows you to configure the BGP PIC feature for all VRFs at once.

- VPNv4 address family configuration mode protects all the VRFs.
- VRF-IPv4 address family configuration mode protects only IPv4 VRFs.
- Router configuration mode protects prefixes in the global routing table.

For a full configuration example that includes configuring multiprotocol VRFs and shows output to verify that the feature is enabled, see the Example Configuring BGP PIC, page 186.

- If you are implementing the BGP PIC feature in an MPLS VPN, ensure that the network is working properly before configuring the BGP PIC feature. See Configuring MPLS Layer 3 VPNs for more information.
- If you are implementing the BGP PIC feature in an MPLS VPN, configure multiprotocol VRFs, which allow you to share route-target policies (import and export) between IPv4 and IPv6 or to configure separate route-target policies for IPv4 and IPv6 VPNs. For information about configuring multiprotocol VRFs, see MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs .
- Ensure that the CE router is connected to the network by at least two paths.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. Do one of the following:
   - **address-family ipv4** [**unicast** | **vrf** *vrf-name*]
   -
   - or
   - **address-family vpnv4** [**unicast**]
5. **bgp additional-paths install**
6. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
7. **neighbor** *ip-address* **activate**
8. **bgp recursion host**
9. **neighbor** *ip-address* **fall-over** [**bfd** |**route-map** *map-name*]
10. **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	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.

Step 4	Do one of the following:
	• **address-family ipv4** [**unicast** | **vrf** *vrf-name*]
	•
	• or
	• **address-family vpnv4** [**unicast**]
	
	Example:
	`Router(config-router)# address-family ipv4 unicast`
	
	Example:
	
	Example:
	
	Example:
	`Router(config-router)# address-family vpnv4`
		Specifies the IPv4 or VPNv4 address family and enters address family configuration mode.
		• The **unicast** keyword specifies the IPv4 or VPNv4 unicast address family.
		• The **vrf** keyword and *vrf-name* argument specify the name of the virtual routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.

Step 5	**bgp additional-paths install** **Example:** `Router(config-router-af)# bgp additional-paths install`	Calculates a backup/alternate path and installs it into the RIB and Cisco Express Forwarding.

	Command or Action	Purpose
Step 6	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types.
Step 7	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local router.
Step 8	**bgp recursion host** **Example:** `Router(config-router-af)# bgp recursion host`	(Optional) Enables the recursive-via-host flag for IPv4, VPNv4, and VRF address families. • When the BGP PIC feature is enabled, Cisco Express Forwarding recursion is disabled. Under most circumstances, you do not want to enable recursion when BGP PIC is enabled.
Step 9	**neighbor** *ip-address* **fall-over** [**bfd** |**route-map** *map-name*] **Example:** `Router(config-router-af)# neighbor 192.168.1.1 fall-over bfd`	Enables BFD protocol support to detect when a neighbor has gone away, which can occur within a subsecond.
Step 10	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuration Examples for BGP PIC

- Example Configuring BGP PIC, page 186
- Example Displaying Backup Alternate Paths for BGP PIC, page 187

Example Configuring BGP PIC

The following example shows how to configure the BGP PIC feature in VPNv4 address family configuration mode, which enables the feature on all VRFs. In the following example, there are two VRFs defined: blue and green. All the VRFs, including those in VRFs blue and green, are protected by backup/alternate paths.

```
vrf definition test1
 rd 400:1
 route-target export 100:1
 route-target export 200:1
 route-target export 300:1
 route-target export 400:1
 route-target import 100:1
 route-target import 200:1
 route-target import 300:1
 route-target import 400:1
 address-family ipv4
 exit-address-family
 exit
!
interface Ethernet1/0
 vrf forwarding test1
 ip address 10.0.0.1 255.0.0.0
 exit
router bgp 3
 no synchronization
 bgp log-neighbor-changes
 redistribute static
 redistribute connected
 neighbor 10.6.6.6 remote-as 3
 neighbor 10.6.6.6 update-source Loopback0
 neighbor 10.7.7.7 remote-as 3
 neighbor 10.7.7.7 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
  bgp additional-paths install
  neighbor 10.6.6.6 activate
  neighbor 10.6.6.6 send-community both
  neighbor 10.7.7.7 activate
  neighbor 10.7.7.7 send-community both
 exit-address-family
 !
 address-family ipv4 vrf blue
  import path selection all
  import path limit 10
  no synchronization
  neighbor 10.11.11.11 remote-as 1
  neighbor 10.11.11.11 activate
 exit-address-family
 !
 address-family ipv4 vrf green
  import path selection all
  import path limit 10
  no synchronization
  neighbor 10.13.13.13 remote-as 1
  neighbor 10.13.13.13 activate
 exit-address-family
```

The following **show vrf detail** command output shows that the BGP PIC feature is enabled:

```
Router# show vrf detail
VRF test1 (VRF Id = 1); default RD 400:1; default VPNID <not set>
  Interfaces:
    Se4/0
Address family ipv4 (Table ID = 1 (0x1)):
  Export VPN route-target communities
    RT:100:1                RT:200:1                RT:300:1
```

```
    RT:400:1
  Import VPN route-target communities
    RT:100:1                RT:200:1                RT:300:1
    RT:400:1
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
  Prefix protection with additional path enabled
Address family ipv6 not active.
```

Example Displaying Backup Alternate Paths for BGP PIC

The command output in the following example shows that the VRFs in VRF blue have backup/alternate paths:

```
Router# show ip bgp vpnv4 vrf blue 10.0.0.0
BGP routing table entry for 10:12:12.0.0.0/24, version 88
Paths: (4 available, best #1, table blue)
  Additional-path
  Advertised to update-groups:
     6
  1, imported path from 12:23:12.0.0.0/24
    10.3.3.3 (metric 21) from 10.6.6.6 (10.6.6.6)
      Origin incomplete, metric 0, localpref 200, valid, internal, best
      Extended Community: RT:12:23
      Originator: 10.3.3.3, Cluster list: 10.0.0.1 , recursive-via-host
      mpls labels in/out nolabel/37
  1, imported path from 12:23:12.0.0.0/24
    10.13.13.13 (via green) from 10.13.13.13 (10.0.0.2)
      Origin incomplete, metric 0, localpref 100, valid, external
      Extended Community: RT:12:23 , recursive-via-connected
  1, imported path from 12:23:12.0.0.0/24
    10.3.3.3 (metric 21) from 10.7.7.7 (10.7.7.7)
      Origin incomplete, metric 0, localpref 200, valid, internal
      Extended Community: RT:12:23
      Originator: 10.3.3.3, Cluster list: 10.0.0.1 , recursive-via-host
      mpls labels in/out nolabel/37
  1
    10.11.11.11 from 10.11.11.11 (1.0.0.1)
      Origin incomplete, metric 0, localpref 100, valid, external, backup/repair
      Extended Community: RT:11:12 , recursive-via-connected
```

The command output in the following example shows that the VRFs in VRF green have backup/alternate paths:

```
Router# show ip bgp vpnv4 vrf green 12.0.0.0
BGP routing table entry for 12:23:12.0.0.0/24, version 87
Paths: (4 available, best #4, table green)
  Additional-path
  Advertised to update-groups:
     5
  1, imported path from 11:12:12.0.0.0/24
    10.11.11.11 (via blue) from 10.11.11.11 (1.0.0.1)
      Origin incomplete, metric 0, localpref 100, valid, external
      Extended Community: RT:11:12 , recursive-via-connected
  1
    10.3.3.3 (metric 21) from 10.7.7.7 (10.7.7.7)
      Origin incomplete, metric 0, localpref 200, valid, internal
      Extended Community: RT:12:23
      Originator: 10.3.3.3, Cluster list: 10.0.0.1 , recursive-via-host
      mpls labels in/out nolabel/37
  1
    10.13.13.13 from 10.13.13.13 (10.0.0.2)
      Origin incomplete, metric 0, localpref 100, valid, external, backup/repair
      Extended Community: RT:12:23 , recursive-via-connected
  1
    10.3.3.3 (metric 21) from 10.6.6.6 (10.6.6.6)
      Origin incomplete, metric 0, localpref 200, valid, internal, best
      Extended Community: RT:12:23
```

```
      Originator: 10.3.3.3, Cluster list: 10.0.0.1 , recursive-via-host
      mpls labels in/out nolabel/37
```

The command output in the following example shows the BGP routing table entries for the backup and alternate paths:

```
Router# show ip bgp 10.0.0.0 255.255.0.0
BGP routing table entry for 10.0.0.0/16, version 123
Paths: (4 available, best #3, table default)
  Additional-path
  Advertised to update-groups:
     2          3
  Local
    10.0.101.4 from 10.0.101.4 (10.3.3.3)
      Origin IGP, localpref 100, weight 500, valid, internal
  Local
    10.0.101.3 from 10.0.101.3 (10.4.4.4)
      Origin IGP, localpref 100, weight 200, valid, internal
  Local
    10.0.101.2 from 10.0.101.2 (10.1.1.1)
      Origin IGP, localpref 100, weight 900, valid, internal, best
  Local
    10.0.101.1 from 10.0.101.1 (10.5.5.5)
      Origin IGP, localpref 100, weight 700, valid, internal, backup/repair
```

The command output in the following example shows the routing information base entries for the backup and alternate paths:

```
Router# show ip route repair-paths 10.0.0.0 255.255.0.0
Routing entry for 10.0.0.0/16
  Known via "bgp 10", distance 200, metric 0, type internal
  Last update from 10.0.101.2 00:00:56 ago
  Routing Descriptor Blocks:
  * 10.0.101.2, from 10.0.101.2, 00:00:56 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
      MPLS label: none
    [RPR]10.0.101.1, from 10.0.101.1, 00:00:56 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
      MPLS label: none
```

The command output in the following example shows the Cisco Express Forwarding/forwarding information base entries for the backup and alternate paths:

```
Router# show ip cef 10.0.0.0 255.255.0.0 detail
10.0.0.0/16, epoch 0, flags rib only nolabel, rib defined all labels
  recursive via 10.0.101.2
    attached to GigabitEthernet0/2
  recursive via 10.0.101.1, repair
    attached to GigabitEthernet0/2
```

Additional References

Related Documents

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

Related Topic	Document Title
BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Routing: BGP Command Reference
Basic MPLS VPNs	Configuring MPLS Layer 3 VPNs
A failover feature that creates a new path after a link or node failure	MPLS VPN--BGP Local Convergence
Configuring multiprotocol VRFs	MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs
BGP routing	BGP Feature Roadmap

Standards

Standard	Title
draft-walton-bgp-add-paths-04.txt	*Advertisement of Multiple Paths in BGP*

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 2547	*BGP/MPLS VPNs*
RFC 1771	A Border Gateway Protocol 4 (BGP-4)

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 BGP PIC

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

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

Table 13 Feature Information for BGP PIC

Feature Name	Releases	Feature Information
BGP PIC Edge for IP and MPLS-VPN	12.2(33)SRE 12.2(33)XNE 15.0(1)S	The BGP PIC Edge for IP and MPLS-VPN feature improves BGP convergence after a network failure. This convergence is applicable to both core and edge failures and can be used in both IP and MPLS networks. The BGP PIC Edge for IP and MPLS-VPN feature creates and stores a backup/alternate path in the routing information base (RIB), forwarding information base (FIB), and Cisco Express Forwarding so that when a failure is detected, the backup/alternate path can immediately take over, thus enabling fast failover. In 12.2(33)SRE, this feature was introduced on the Cisco 7200 and Cisco 7600 routers. In 12.2(33)XNE, support was added for the Cisco 10000 router. The following sections provide information about this feature: The following commands were introduced or modified: **bgp additional-paths install**, **bgp recursion host**, **show ip bgp**, **show ip cef**, **show ip route**, **show vrf**.

MPLS VPN--L3VPN over GRE

The MPLS VPN--L3VPN over GRE feature provides a mechanism for tunneling Multiprotocol Label Switching (MPLS) packets over a non-MPLS network.

The MPLS VPN--L3VPN over GRE feature utilizes MPLS over generic routing encapsulation (MPLSoGRE) to encapsulate MPLS packets inside IP tunnels. This action creates a virtual point-to-point link across non-MPLS networks.

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--L3VPN over GRE

Before you configure the MPLS VPN--L3VPN over GRE feature, ensure that your MPLS Virtual Private Network (VPN) is configured and working properly. See the Configuring MPLS Layer 3 VPNs module for information about setting up MPLS VPNs.

Ensure that the following routing protocols are configured and working properly:

- Label Distribution Protocol (LDP)--for MPLS label distribution. See MPLS Label Distribution Protocol Overview
- Multiprotocol Border Gateway Protocol (MP-BGP)--for VPN route and label distribution. See Configuring MPLS Layer 3 VPNs

Restrictions for MPLS VPN--L3VPN over GRE

The MPLS VPN--L3VPN over GRE feature does not support the following:

- Quality of service (QoS) service policies configured on the tunnel interface; they are supported on the physical or subinterface
- GRE options: sequencing, checksum, and source route
- IPv6 GRE
- Advanced features such as Carrier Supporting Carrier (CSC) and Interautonomous System (Inter-AS)
- For PE-to-PE tunneling, configure tunnels with the same source address if you are running a release earlier than Cisco IOS Release 15.2(1)S.
- For PE-to-PE tunneling, configure tunnels with the same destination address

Information About MPLS VPN--L3VPN over GRE

The MPLS VPN--L3VPN over GRE feature provides a mechanism for tunneling MPLS packets over non-MPLS networks.

MPLS VPN--L3VPN over GRE allows you to create a GRE tunnel across a non-MPLS network. The MPLS packets are encapsulated within the GRE tunnel packets, and the encapsulated packets traverse the non-MPLS network through the GRE tunnel. When GRE tunnel packets are received at the other side of the non-MPLS network, the GRE tunnel packet header is removed and the inner MPLS packet is forwarded to its final destination.

The MPLS VPN--L3VPN over GRE feature supports three GRE tunnel configurations:

- PE-to-PE Tunneling, page 194
- P-to-PE Tunneling, page 195
- P-to-P Tunneling, page 195

PE-to-PE Tunneling

The provider edge-to-provider edge (PE-to-PE) tunneling configuration provides a scalable way to connect multiple customer networks across a non-MPLS network. With this configuration, traffic that is destined to multiple customer networks is multiplexed through a single GRE tunnel.

Note A similar nonscalable alternative is to connect each customer network through separate GRE tunnels (for example, connecting one customer network for each GRE tunnel).

As shown in the figure below, the PE routers assign VPN routing and forwarding (VRF) numbers to the customer edge (CE) routers on each side of the non-MPLS network.

The PE routers use routing protocols such as BGP, OSPF, or Routing Information Protocol (RIP) to learn about the IP networks behind the CE routers. The routes to the IP networks behind the CE routers are stored in the associated CE router's VRF routing table.

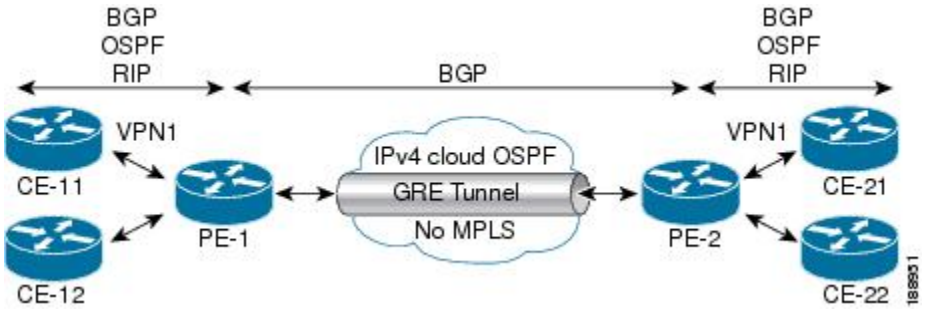
The PE router on one side of the non-MPLS network uses the routing protocols (that are operating within the non-MPLS network) to learn about the PE router on the other side of the non-MPLS network. The

learned routes that are established between the PE routers are then stored in the main or default routing table.

The opposing PE router uses BGP to learn about the routes that are associated with the customer networks behind the PE routers. These learned routes are not known to the non-MPLS network.

For this example, BGP defines a static route to the BGP neighbor (the opposing PE router) through the GRE tunnel that spans the non-MPLS network. Because the routes that are learned by the BGP neighbor include the GRE tunnel next hop, all customer network traffic is sent using the GRE tunnel.

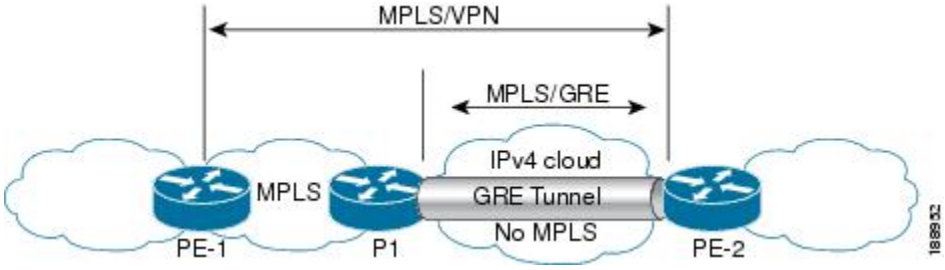
Figure 14 PE-to-PE Tunneling



P-to-PE Tunneling

As shown in the figure below, the provider-to-provider edge (P-to-PE) tunneling configuration provides a way to connect a PE router (P1) to an MPLS segment (PE-2) across a non-MPLS network. In this configuration, MPLS traffic that is destined to the other side of the non-MPLS network is sent through a single GRE tunnel.

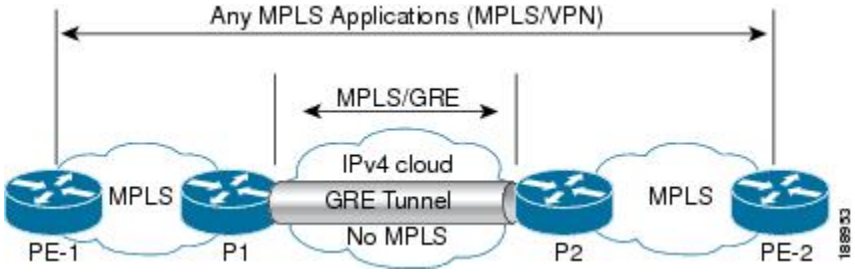
Figure 15 P-to-PE Tunneling



P-to-P Tunneling

As shown in the figure below, the provider-to-provider (P-to-P) configuration provides a method of connecting two MPLS segments (P1 to P2) across a non-MPLS network. In this configuration, MPLS traffic that is destined to the other side of the non-MPLS network is sent through a single GRE tunnel.

Figure 16 P-to-P Tunneling



How to Configure MPLS VPN--L3VPN over GRE

Configuring the MPLS VPN--L3VPN over GRE Tunnel Interface

To configure the MPLS VPN--L3VPN over GRE feature, you must create a GRE tunnel to span the non-MPLS networks. You must perform this procedure on the devices located at both ends of the GRE tunnel.

Before configuring the MPLS VPN--L3VPN over GRE feature, ensure that your MPLS VPN and the appropriate routing protocols are configured and working properly. See the Prerequisites for MPLS VPN--L3VPN over GRE, page 193.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **ip address** *ip-address*
5. **tunnel source** *source-address*
6. **tunnel destination** *destination-address*
7. **mpls ip**

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 tunnel** *tunnel-number* **Example:** `Router(config)# interface tunnel 1`	Creates a tunnel on the specified interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip address** *ip-address* **Example:** `Router(config-if)# ip address 10.0.0.1 255.255.255.0`	Assigns an IP address to the tunnel interface.
Step 5	**tunnel source** *source-address* **Example:** `Router(config-if)# tunnel source 10.1.1.1`	Specifies the tunnel's source IP address.
Step 6	**tunnel destination** *destination-address* **Example:** `Router(config-if)# tunnel destination 10.1.1.2`	Specifies the tunnel's destination IP address.
Step 7	**mpls ip** **Example:** `Router(config-if)# mpls ip`	Enables MPLS on the tunnel's physical interface.

- Examples, page 197

Examples

The following example shows a GRE tunnel configuration that spans a non-MPLS network. This example shows the tunnel configuration on the PE devices (PE1 and PE2) located at both ends of the tunnel:

PE1 Configuration

```
Router# configure terminal
Router(config)# interface Tunnel 1
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# tunnel source 10.0.0.1
Router(config-if)# tunnel destination 10.0.0.2
Router(config-if)# mpls ip
```

PE2 Configuration

```
Router# configure terminal
Router(config)# interface Tunnel 1
Router(config-if)# ip address 10.1.1.2 255.255.255.0
Router(config-if)# tunnel source 10.0.0.2
Router(config-if)# tunnel destination 10.0.0.1
Router(config-if)# mpls ip
```

Configuration Examples for MPLS VPN--L3VPN over GRE

MPLS Configuration with MPLS VPN--L3VPN over GRE Example

The following basic MPLS configuration example uses a GRE tunnel to span a non-MPLS network. This example is similar to the configuration shown in the first figure above.

PE1 Configuration

```
!
mpls ip
!
ip vrf vpn1
rd 100:1
route-target import 100:1
route-target export 100:1
!
interface loopback 0
ip address 10.2.2.2 255.255.255.255
!
interface GigabitEthernet 0/1/2
ip address 10.1.1.1 255.255.255.0
!
interface Tunnel 1
ip address 10.0.0.1 255.255.255.0
tunnel source 10.1.1.1
tunnel destination 10.1.1.2
mpls ip
!
interface GigabitEthernet 0/1/3
ip vrf forwarding vpn1
ip address 10.10.0.1 255.255.255.0
!
router bgp 100
neighbor 10.5.5.5 remote-as 100
neighbor 10.5.5.5 update-source loopback0
!
address-family vpnv4
neighbor 10.5.5.5 activate
neighbor 10.5.5.5 send community-extended
!
address-family ipv4 vrf vpn1
neighbor 10.10.0.2 remote-as 20
neighbor 10.10.0.2 activate
!
```

PE2 Configuration

```
!
mpls ip
!
ip vrf vpn1
rd 100:1
route-target import 100:1
route-target export 100:1
!
interface loopback 0
ip address 10.5.5.5 255.255.255.255
!
interface GigabitEthernet 0/1/1
ip address 10.1.1.2 255.255.255.0
!
```

```
interface Tunnel 1
ip address 10.0.0.2 255.255.255.0
tunnel source 10.1.1.2
tunnel destination 10.1.1.1
mpls ip
!
interface GigabitEthernet 0/0/5
ip vrf forwarding vpn1
ip address 10.1.2.1 255.255.255.0
!
router bgp 100
neighbor 10.2.2.2 remote-as 100
neighbor 10.2.2.2 update-source loopback0
!
address-family vpnv4
neighbor 10.2.2.2 activate
neighbor 10.2.2.2 send community-extended
!
address-family ipv4 vrf vpn1
neighbor 10.1.2.2 remote-as 30
neighbor 10.1.2.2 activate
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Multiprotocol Label Switching (MPLS) commands	*Cisco IOS Multiprotocol Label Switching Command Reference*
Setting up MPLS VPN networks	Configuring MPLS Layer 3 VPNs
Label Distribution Protocol	MPLS Label Distribution Protocol Overview
Multiprotocol Border Gateway Protocol (MP-BGP)	Configuring MPLS Layer 3 VPNs
Configuring L3 VPN over mGRE Tunnels	Dynamic Layer-3 VPNs with Multipoint GRE Tunnels

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
None	--

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--L3VPN over GRE

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

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

Table 14 Feature Information for MPLS VPN--L3VPN over GRE

Feature Name	Releases	Feature Information
MPLS VPN--L3VPN over GRE feature	12.0(22)S 12.2(13)T 12.0(26)S 12.2(33)SRE Cisco IOS XE Release 2.1 15.2(1)S	The MPLS VPN--L3VPN over GRE feature provides a mechanism for tunneling Multiprotocol Label Switching (MPLS) packets over a non-MPLS network. In Cisco IOS Release 15.2(1)S, you can configure tunnels with the same source address in a PE-to-PE tunneling configuration. This feature uses no new or modified commands.

Dynamic Layer 3 VPNs with Multipoint GRE Tunnels

The Dynamic Layer 3 VPNs with Multipoint GRE Tunnels feature provides a Layer 3 (L3) transport mechanism based on an enhanced multipoint generic routing encapsulation (mGRE) tunneling technology for use in IP networks. The dynamic Layer 3 tunneling transport can also be used within IP networks to transport Virtual Private Network (VPN) traffic across service provider and enterprise networks, and to provide interoperability for packet transport between IP and Multiprotocol Label Switching (MPLS) VPNs. This feature provides support for RFC 2547, which defines the outsourcing of IP backbone services for enterprise networks.

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 for Dynamic L3 VPNs with mGRE Tunnels, page 224.

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

Prerequisites for Dynamic L3 VPNs with mGRE Tunnels

Before you configure the Dynamic Layer 3 VPNs with Multipoint GRE Tunnels feature, ensure that your MPLS VPN is configured and working properly. See the "Configuring MPLS Layer 3 VPNs" module for information about setting up MPLS VPNs.

Restrictions for Dynamic L3 VPNs with mGRE Tunnels

- The deployment of a MPLS VPN using both IP/GRE and MPLS encapsulation within a single network is not supported.
- Each provider edge (PE) router supports one tunnel configuration only.

Information About Dynamic L3 VPNs with mGRE Tunnels

You can configure mGRE tunnels to create a multipoint tunnel network that overlays an IP backbone. This overlay connects PE routers to transport VPN traffic. To deploy L3 VPN mGRE tunnels, you create a VRF instance, create the mGRE tunnel, redirect the VPN IP traffic to the tunnel, and set up the BGP VPNv4 exchange so that updates are filtered through a route map and interesting prefixes are resolved in the VRF table.

In addition, when MPLS VPNs are configured over mGRE, you can deploy L3 PE-based VPN services using a standards-based IP core. This allows you to provision the VPN services without using the overlay method. When an MPLS VPN over mGRE is configured, the system uses IPv4-based mGRE tunnels to encapsulate VPN-labeled IPv4 and IPv6 packets between PEs.

- Layer 3 mGRE Tunnels, page 204

Layer 3 mGRE Tunnels

By configuring mGRE tunnels, you create a multipoint tunnel network as an overlay to the IP backbone. This overlay interconnects the PE routers to transport VPN traffic through the backbone. This multipoint tunnel network uses Border Gateway Protocol (BGP) to distribute VPNv4 routing information between PE routers, maintaining the peer relationship between the service provider or enterprise network and customer sites. The advertised next hop in BGP VPNv4 triggers tunnel endpoint discovery. This feature provides the ability for multiple service providers to cooperate and offer a joint VPN service with traffic tunneled directly from the ingress PE router at one service provider directly to the egress PE router at a different service provider site.

In addition to providing the VPN transport capability, the mGRE tunnels create a full-mesh topology and reduce the administrative and operational overhead previously associated with a full mesh of point-to-point tunnels used to interconnect multiple customer sites. The configuration requirements are greatly reduced and enable the network to grow with minimal additional configuration.

Dynamic L3 tunnels provide for better scaling when creating partial-mesh or full-mesh VPNs. Adding new remote VPN peers is simplified because only the new router needs to be configured. The new address is learned dynamically and propagated to the nodes in the network. The dynamic routing capability dramatically reduces the size of configuration needed on all routers in the VPN, such that with the use of multipoint tunnels, only one tunnel interface needs to be configured on a PE that services many VPNs. The L3 mGRE tunnels need to be configured only on the PE router. Features available with GRE are still available with mGRE, including dynamic IP routing and IP multicast and Cisco Express Forwarding (CEF) switching of mGRE/Next Hop Routing Protocol (NHRP) tunnel traffic.

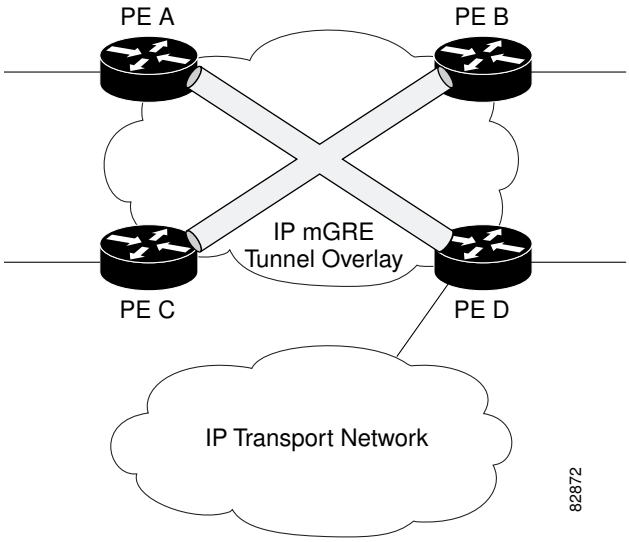
The following sections describe how the mGRE tunnels are used:

- Interconnecting Provider Edge Routers Within an IP Network, page 205
- Packet Transport Between IP and MPLS Networks, page 205
- BGP Next Hop Verification, page 206

Interconnecting Provider Edge Routers Within an IP Network

The Dynamic Layer 3 VPNs with Multipoint GRE Tunnels feature allows you to create a multiaccess tunnel network to interconnect the PE routers that service your IP network. This tunnel network transports IP VPN traffic to all of the PE routers. The figure below illustrates the tunnel overlay network used in an IP network to transport VPN traffic between the PE routers.

Figure 17 mGRE Tunnel Overlay Connecting PE Routers Within an IP Network



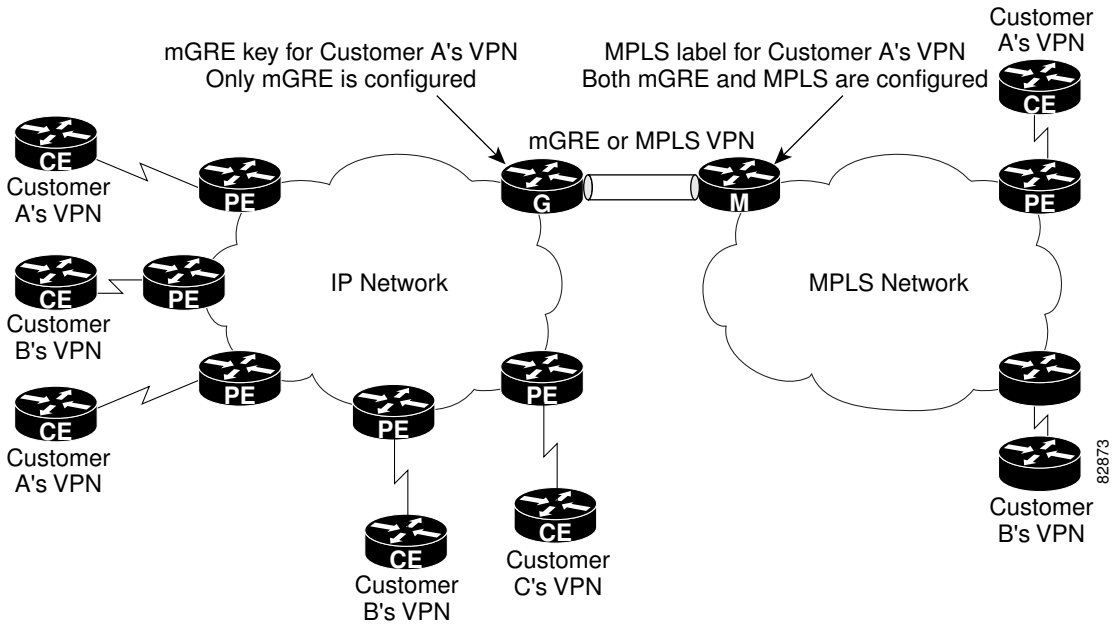
The multiaccess tunnel overlay network provides full connectivity between PE routers. The PE routers exchange VPN routes by using BGP as defined in RFC 2547. IP traffic is redirected through the multipoint tunnel overlay network using distinct IP address spaces for the overlay and transport networks and by changing the address space instead of changing the numerical value of the address.

Packet Transport Between IP and MPLS Networks

Layer 3 mGRE tunnels can be used as a packet transport mechanism between IP and MPLS networks. To enable the packet transport between the two different protocols, one PE router on one side of the

connection between the two networks must run MPLS. The figure below shows how mGRE tunnels can be used to transport VPN traffic between PE routers.

Figure 18 mGRE Used to Transport VPN Traffic Between IP and MPLS Network



For the packet transport to occur between the IP and MPLS network, the MPLS VPN label is mapped to the GRE key. The mapping takes place on the router where both mGRE and MPLS are configured. In the figure above the mapping of the label to the key occurs on Router M, which sits on the MPLS network.

BGP Next Hop Verification

BGP performs the BGP path selection, or next hop verification, at the PE. For a BGP path to a network to be considered in the path selection process, the next hop for the path must be reachable in the Interior Gateway Protocol (IGP). When an IP prefix is received and advertised as the next hop IP address, the IP traffic is tunneled from the source to the destination by switching the address space of the next hop.

How to Configure L3 VPN mGRE Tunnels

- Creating the VRF and mGRE Tunnel, page 207
- Setting Up BGP VPN Exchange, page 209
- Enabling the MPLS VPN over mGRE Tunnels and Configuring an L3VPN Encapsulation Profile, page 211
- Defining the Address Space and Specifying Address Resolution for MPLS VPNs over mGRE, page 214

Creating the VRF and mGRE Tunnel

The tunnel that transports the VPN traffic across the service provider network resides in its own address space. A special VRF instance must be created called Resolve in VRF (RiV). This section describes how to create the VRF and GRE tunnel.

The IP address on the interface should be the same as that of the source interface specified in the configuration. The source interface specified should match that used by BGP as a source for the VPNv4 update.

Note Tunnel mode IPSec is not supported on MPLS over GRE Tunnel.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd 1:1**
5. **interface tunnel** *tunnel-name*
6. **ip address** *ip-address subnet-id*
7. **tunnel source loopback** *n*
8. **tunnel mode gre multipoint l3vpn**
9. **tunnel key** *gre-ke* y
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	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf customer a riv`	Creates the special Resolve in VRF (RiV) VRF instance and table that will be used for the tunnel and redirection of the IP address.

	Command or Action	Purpose
Step 4	**rd 1:1** **Example:** `Router(config-vrf)# rd 1:1`	Enters the VRF configuration mode and specifies a route distinguisher (RD) for a VPN VRF instance.
Step 5	**interface tunnel** *tunnel-name* **Example:** `Router(config-vrf)# interface tunnel 1`	Enters interface configuration mode to create the tunnel.
Step 6	**ip address** *ip-address subnet-id* **Example:** `Router(config-if)# ipaddress 209.165.200.225 255.255.255.224`	Specifies the IP address for the tunnel.
Step 7	**tunnel source loopback** *n* **Example:** `Router(config-if)# tunnel source loopback test1`	Creates the loopback interface.
Step 8	**tunnel mode gre multipoint l3vpn** **Example:** `Router(config-if)# tunnel mode gre multipoint l3vpn`	Sets the mode for the tunnel as "gre multipoint l3vpn".
Step 9	**tunnel key** *gre-ke* y **Example:** `Router(config-if)# tunnel key 18`	Specifies the GRE key for the tunnel.
Step 10	**end** **Example:** `Router(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Setting Up BGP VPN Exchange

The configuration task described in this section sets up the BGP VPNv4 exchange so that the updates are filtered through a route map and interesting prefixes are resolved in the VRF table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-name*
4. **ip route vrf** *riv-vrf-name ip-address subnet- mask* **tunnel** *n*
5. **router bgp** *as-number*
6. **network** *network-id*
7. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
8. **neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type*
9. **address-family vpnv4** [**unicast**]
10. **neighbor** {*ip-address* | *peer-group-name*} **activate**
11. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
12. **set ip next-hop resolve-in-vrf** *vrf-name*
13. **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 tunnel** *tunnel-name* **Example:** `Router(config)# interface tunnel 1`	Enters interface configuration mode for the tunnel.

	Command or Action	Purpose
Step 4	**ip route vrf** *riv-vrf-name ip-address subnet- mask* **tunnel** *n* **Example:** `Router(config-if)# ip route vrf vrf1 209.165.200.226 255.255.255.224 tunnel 1`	Sets the packet forwarding to the special RiV VRF.
Step 5	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Specifies the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along.
Step 6	**network** *network-id* **Example:** `Router(config)# network 209.165.200.255`	Specifies the network ID for the networks to be advertised by the BGP and multiprotocol BGP routing processes.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config)# neighbor 209.165.200.227 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type* **Example:** `Router(config)# neighbor 209.165.200.228 update-source FastEthernet0/1`	Specifies a specific operational interface that BGP sessions use for TCP connections.
Step 9	**address-family vpnv4** [**unicast**] **Example:** `Router(config)# address-family vpnv4`	Specifies address family configuration mode for configuring routing sessions, such as BGP, that use standard VPN4 address prefixes.
Step 10	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config)# neighbor 209.165.200.229 activate`	Enables the exchange of information with a neighboring router.

	Command or Action	Purpose
Step 11	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **Example:** `Router(config)# neighbor 209.165.200.230 route-map mpt in`	Applies a route map to incoming or outgoing routes. • Use once for each inbound route.
Step 12	**set ip next-hop resolve-in-vrf** *vrf-name* **Example:** `Router(config)# set ip next-hop resolve-in-vrf vrft`	Specifies that the next hop is to be resolved in the VRF table for the specified VRF.
Step 13	**end** **Example:** `Router(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Enabling the MPLS VPN over mGRE Tunnels and Configuring an L3VPN Encapsulation Profile

This section describes how to define the VRF, enable MPLS VPN over mGRE, and configure an L3VPN encapsulation profile.

Note Transport protocols such as IPv6, MPLS, IP, and Layer 2 Tunneling Protocol version 3 (L2TPv3) can also be used in this configuration.

To enable and configure MPLS VPN over mGRE, you must first define the VRF for tunnel encapsulation and enable L3VPN encapsulation in the system.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd 1:1**
5. **exit**
6. **ip cef**
7. **ipv6** *unicast-routing*
8. **ipv6 cef**
9. **l3vpn encapsulation ip** *profile-name*
10. **transport ipv4 source** *interface n*
11. **protocol gre [key** *gre-key* **]**
12. **exit**
13. **interface** *type number*
14. **ip address** *ip-address mask*
15. **ip router isis**
16. **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 tunnel encap`	Configures a VPN VRF routing table instance and enters VRF configuration mode.

	Command or Action	Purpose
Step 4	**rd 1:1** **Example:** `Router(config-vrf)# rd 1:1`	Specifies an RD for a VPN VRF instance.
Step 5	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode.
Step 6	**ip cef** **Example:** `Router(config)# ip cef`	Enables Cisco Express Forwarding on the router.
Step 7	**ipv6** *unicast-routing* **Example:** `Router(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 8	**ipv6 cef** **Example:** `Router(config)# ipv6 cef`	Enables Cisco Express Forwarding for IPv6 on the router.
Step 9	**l3vpn encapsulation ip** *profile-name* **Example:** `Router(config)# l3vpn encapsulation ip tunnel encap`	Enters L3 VPN encapsulation configuration mode to create the tunnel.
Step 10	**transport ipv4 source** *interface n* **Example:** `Router(config-l3vpn-encap-ip)# transport ipv4 source loopback 0`	Specifies IPv4 transport source mode and defines the transport source interface.

	Command or Action	Purpose
Step 11	**protocol gre [key** *gre-key* **]** **Example:** `Router(config-l3vpn-encap-ip)# protocol gre key 1234`	Specifies GRE as the tunnel mode and sets the GRE key.
Step 12	**exit** **Example:** `Router(config-l3vpn-encap-ip)# exit`	Exits L3 VPN encapsulation configuration mode.
Step 13	**interface** *type number* **Example:** `Router(config)# interface loopback 0`	Enters interface configuration mode to configure the interface type.
Step 14	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.10.10.4 255.255.255.255`	Specifies the primary IP address and mask for the interface.
Step 15	**ip router isis** **Example:** `Router(config-if)# ip router isis`	Configures an Intermediate System-to-Intermediate System (IS-IS) routing process for IP on the interface and attaches a null area designator to the routing process.
Step 16	**end** **Example:** `Router(config-if)#end`	Exits the current configuration mode and returns to privileged EXEC mode.

Defining the Address Space and Specifying Address Resolution for MPLS VPNs over mGRE

This section describes how to define the address space and specify the address resolution for MPLS VPNs over mGRE. The following steps also enable you to link the route map to the application template and set up the BGP VPNv4 and VPNv6 exchange so that updates are filtered through the route map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **bgp log-neighbor-changes**
5. **neighbor** *ip-address* **remote-as** *as-number*
6. **neighbor** *ip-address* **update-source** *interface-type interface-name*
7. **address-family vpn4**
8. **no synchronization**
9. **redistribute connected**
10. **neighbor** *ip-address* **activate**
11. **no auto-summary**
12. **exit**
13. **address-family vpnv4**
14. **neighbor** *ip-address* **activate**
15. **neighbor** *ip-address* **send-community both**
16. **neighbor** *ip-address* **route-map** *map-name* **in**
17. **exit**
18. **address-family vpnv6**
19. **neighbor** *ip-address* **activate**
20. **neighbor** *ip-address* **send-community both**
21. **neighbor** *ip-address* **route-map** *ip-address* **in**
22. **exit**
23. **route-map** *map-tag* **permit** *position*
24. **set ip next-hop encapsulate l3vpn** *tunnel encap*
25. **set ipv6 next-hop encapsulate l3vpn** *profile name*
26. **end**
27. **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	**router bgp** *as-number* **Example:** `Router (config)# router bgp 100`	Specifies the number of an autonomous system that identifies the router to other BGP routers, tags the routing information passed along, and enters router configuration mode.
Step 4	**bgp log-neighbor-changes** **Example:** `Router (config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 5	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Router (config-router)# neighbor 10.10.10.6 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table.
Step 6	**neighbor** *ip-address* **update-source** *interface-type interface-name* **Example:** `Router (config-router)# neighbor 10.10.10.6 update-source loopback 0`	Allows BGP sessions to use any operational interface for TCP connections.
Step 7	**address-family vpn4** **Example:** `Router (config-router)# address-family vpnv4`	Enters address family configuration mode to configure routing sessions, that use IPv4 address prefixes.
Step 8	**no synchronization** **Example:** `Router (config-router-af)# no synchronization`	Enables the Cisco IOS software to advertise a network route without waiting for an IGP.

	Command or Action	Purpose
Step 9	**redistribute connected** **Example:** `Router (config-router-af)# redistribute connected`	Redistributes routes from one routing domain into another routing domain and allows the target protocol to redistribute routes learned by the source protocol and connected prefixes on those interfaces over which the source protocol is running.
Step 10	**neighbor** *ip-address* **activate** **Example:** `Router (config-router-af)# neighbor 10.10.10.6 activate`	Enables the exchange of information with a BGP neighbor.
Step 11	**no auto-summary** **Example:** `Router (config-router-af)# no auto-summary`	Disables automatic summarization and sends subprefix routing information across classful network boundaries
Step 12	**exit** **Example:** `Router (config-router-af)# exit`	Exits address family configuration mode.
Step 13	**address-family vpnv4** **Example:** `Router (config-router)# address-family vpnv4`	Enters address family configuration mode to configure routing sessions, such as BGP, that use standard VPNv4 address prefixes.
Step 14	**neighbor** *ip-address* **activate** **Example:** `Router (config-router-af)# neighbor 10.10.10.6 activate`	Enables the exchange of information with a BGP neighbor.
Step 15	**neighbor** *ip-address* **send-community both** **Example:** `Router (config-router-af)# neighbor 10.10.10.6 send-community both`	Specifies that a community attribute, for both standard and extended communities, should be sent to a BGP neighbor.

Command or Action	Purpose
Step 16 **neighbor** *ip-address* **route-map** *map-name* **in** **Example:** `Router (config-router-af)# neighbor 10.10.10.6 route-map SELECT UPDATE FOR L3VPN in`	Applies the named route map to the incoming route.
Step 17 **exit** **Example:** `Router (config-router-af)# exit`	Exits address family configuration mode.
Step 18 **address-family vpnv6** **Example:** `6Router (config-router)# address-family vpnv4`	Enters address family configuration mode to configure routing sessions, such as BGP, that use VPNv6 address prefixes.
Step 19 **neighbor** *ip-address* **activate** **Example:** `Router (config-router-af)# neighbor 209.165.200.252 activate`	Enables the exchange of information with a BGP neighbor.
Step 20 **neighbor** *ip-address* **send-community both** **Example:** `Router (config-router-af)# neighbor 209.165.200.252 send-community both`	Specifies that a communities attribute, for both standard and extended communities, should be sent to a BGP neighbor.
Step 21 **neighbor** *ip-address* **route-map** *ip-address* **in** **Example:** `Router (config-router-af)# neighbor 209.165.200.252 route-map SELECT UPDATE FOR L3VPN in`	Applies the named route map to the incoming route.
Step 22 **exit** **Example:** `Router (config-router-af)# exit`	Exits address family configuration mode.

	Command or Action	Purpose
Step 23	**route-map** *map-tag* **permit** *position* **Example:** `Router (config-router)# route-map 192.168.10.1 permit 10`	Enters route-map configuration mode and defines the conditions for redistributing routes from one routing protocol into another. • The **redistribute** router configuration command uses the specified map tag to reference this route map. Multiple route maps may share the same map tag name. • If the match criteria are met for this route map, the route is redistributed as controlled by the set actions. • If the match criteria are not met, 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 *position* argument indicates the position that new route map will have in the list of route maps already configured with the same name.
Step 24	**set ip next-hop encapsulate l3vpn** *tunnel encap* **Example:** `Router (config-route-map)# set ip next-hop encapsulate l3vpn my profile`	Indicates that output IPv4 packets that pass a match clause of the route map are sent to the VRF for tunnel encapsulation.
Step 25	**set ipv6 next-hop encapsulate l3vpn** *profile name* **Example:** `Router (config-route-map)# set ip next-hop encapsulate l3vpn tunnel encap` **Example:**	Indicates that output IPv6 packets that pass a match clause of the route map are sent to the VRF for tunnel encapsulation.
Step 26	**end** **Example:** `Router (config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 27	**end** **Example:** `Router (config)# exit`	Exits global configuration mode.

• What to Do Next, page 220

What to Do Next

You can perform the following to make sure that the configuration is working properly.

Check the VRF Prefix

Verify that the specified VRF prefix has been received by BGP. The BGP table entry should show that the route map has worked and that the next hop is showing in the RiV. Use the **show ip bgp vpnv4** command as shown in this example.

```
Router# show ip bgp vpnv4 vrf customer 209.165.200.250
BGP routing table entry for 100:1:209.165.200.250/24, version 12
Paths: (1 available, best #1)
  Not advertised to any peer
  Local
209.165.200.251 in "my riv" from 209.165.200.251 (209.165.200.251)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:1
```

Confirm that the same information has been propagated to the routing table:

```
Router# show ip route vrf customer 209.165.200.250

Routing entry for 209.165.200.250
/24
  Known via "bgp 100", distance 200, metric 0, type internal
  Last update from 209.165.200.251 00:23:07 ago
  Routing Descriptor Blocks:
  * 209.165.200.251 (my riv), from 209.165.200.251, 00:23:07 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
```

CEF Switching

You can also verify that CEF switching is working as expected:

```
Router# show ip cef vrf customer
 209.165.200.250

209.165.200.250
/24, version 6, epoch 0
0 packets, 0 bytes
  tag information set
    local tag: VPN-route-head
    fast tag rewrite with Tu1, 123.1.1.2, tags imposed: {17}
  via 209.165.200.251, 0 dependencies, recursive
    next hop 209.165.200.251, Tunnel1 via 209.165.200.251/32 (my riv)
    valid adjacency
    tag rewrite with Tu1, 209.165.200.251, tags imposed: {17}
```

Endpoint Creation

Note that in this example display the tunnel endpoint has been created correctly:

```
Router# show tunnel endpoint tunnel 1
Tunnel1 running in multi-GRE/IP mode
  RFC2547/L3VPN Tunnel endpoint discovery is active on Tu1
  Transporting l3vpn traffic to all routes recursing through "my riv"
 Endpoint 209.165.200.251 via destination 209.165.200.251
 Endpoint 209.165.200.254 via destination 209.165.200.254
```

Adjacency

Confirm that the corresponding adjacency has been created.

```
Router# show adjacency Tunnel 1 interface
Protocol Interface                 Address
TAG      Tunnel1                   209.165.200.251(4)
                                   15 packets, 1980 bytes
                                   45000000000000000FF2FC3C77B010103
                                   7B01010200008847
                                   Epoch: 0
                                   Fast adjacency disabled
                                   IP redirect disabled
                                   IP mtu 1472 (0x0)
                                   Fixup enabled (0x2)
                                         GRE tunnel
                                   Adjacency pointer 0x624A1580, refCount 4
                                   Connection Id 0x0
                                   Bucket 121
```

Note that because MPLS is being transported over mGRE, the LINK_TAG adjacency is the relevant adjacency. The MTU reported in the adjacency is the payload length (including the MPLS label) that the packet will accept. The MAC string shown in the adjacency display can be interpreted as follows:

```
45000000 -> Beginning of IP Header (Partially populated, tl & chksum
00000000    are fixed up per packet)
FF2FC3C7
7B010103 -> Source IP Address in transport network 209.165.200.253
7B010102 -> Destination IP address in transport network 209.165.200.252
00008847 -> GRE Header
```

Refer to the Cisco IOS Multiprotocol Label Switching Configuration Guide for information about configuring MPLS Layer 3 VPNs.

You can use the **show l3vpn encapsulation** *profile-name* command to get information on the basic state of the application. The output of this command provides you details on the references to the tunnel and VRF.

Configuration Examples for Dynamic L3 VPNs Support Using mGRE Tunnels

Configuring Layer 3 VPN mGRE Tunnels Example

This example shows the configuration sequence for creating mGRE tunnels. It includes the definition of the special VRF instance.

```
ip vrf my riv
 rd 1:1
interface Tunnel1
 ip vrf forwarding my_riv
 ip address 209.165.200.250 255.255.255.224
 tunnel source Loopback0
 tunnel mode gre multipoint l3vpn
 tunnel key 123
end
ip route vrf my riv ip address subnet mask Tunnel1
router bgp 100
 network 209.165.200.251
 neighbor 209.165.200.250 remote-as 100
```

```
 neighbor 209.165.200.250 update-source Loopback0
 !
 address-family vpnv4
 neighbor 209.165.200.250 activate
 neighbor 209.165.200.250 route-map SELECT_UPDATES_FOR_L3VPN_OVER_MGRE in
!
route-map SELECT UPDATES FOR L3VPN OVER MGRE permit 10
 set ip next-hop in-vrf my riv
```

This example shows the configuration to link a route map to the application:

```
vrf definition Customer A
 rd 100:110
 route-target export 100:1000
 route-target import 100:1000
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
vrf definition tunnel encap
 rd 1:1
!
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
!
ip cef
!
ipv6 unicast-routing
ipv6 cef
!
!
l3vpn encapsulation ip profile name
 transport source loopback 0
 protocol gre key 1234
!
!
 interface Loopback0
  ip address 209.165.200.252 255.255.255.224
  ip router isis
!
interface Serial2/0
 vrf forwarding Customer A
 ip address 209.165.200.253 255.255.255.224
 ipv6 address 3FFE:1001::/64 eui-64
 no fair-queue
 serial restart-delay 0
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 209.165.200.254 remote-as 100
 neighbor 209.165.200.254 update-source Loopback0
 !
 address-family ipv4
  no synchronization
  redistribute connected
  neighbor 209.165.200.254 activate
  no auto-summary
 exit-address-family
 !
 address-family vpnv4
  neighbor 209.165.200.254 activate
  neighbor 209.165.200.254 send-community both
  neighbor 209.165.200.254 route-map SELECT UPDATE FOR L3VPN in
 exit-address-family
 !
 address-family vpnv6
```

```
  neighbor 209.165.200.254 activate
  neighbor 209.165.200.254 send-community both
  neighbor 209.165.200.254 route-map SELECT UPDATE FOR L3VPN in
 exit-address-family
 !
 address-family ipv4 vrf Customer A
  no synchronization
  redistribute connected
 exit-address-family
 !
 address-family ipv6 vrf Customer A
  redistribute connected
  no synchronization
 exit-address-family
!
!
route-map SELECT UPDATE FOR L3VPN permit 10
set ip next-hop encapulate <profile_name>
set ipv6 next-hop encapsulate <profile_name>
```

Additional References

For additional information related to dynamic L3 VPN mGRE tunnels, refer to the following references:

Related Documents

Related Topic	Document Title
Configuring MPLS Layer 3 VPNs	*Cisco IOS Multiprotocol Label Switching Configuration Guide*
MPLS VPN Over mGRE	*Cisco IOS Interface and Hardware Component Configuration Guide*
Cisco Express Forwarding	*Cisco IOS IP Switching Configuration Guide*
Generic Routing Encapsulation	*Cisco IOS Interface and Hardware Component Configuration Guide*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
IETF-PPVPN-MPLS-VPN-MIB	To locate and download MIBs for selected platforms, Cisco IOS 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*
RFC 2784	*Generic Routing Encapsulation (GRE)*
RFC 2890	Key Sequence Number Extensions to GRE
RFC 4023	Encapsulating MPLS in IP or Generic Routing Encapsulation
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 Dynamic L3 VPNs with mGRE Tunnels

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

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

Table 15 Feature Information for Dynamic L3 VPNs with mGRE Tunnels

Feature Name	Releases	Feature Information
Dynamic Layer 3 VPNs with Multipoint GRE Tunnels	12.0(23)S	This feature provides an L3 transport mechanism based on an enhanced mGRE tunneling technology for use in IP networks.