MPLS Virtual Private Networks

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

Finding Feature Information

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

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

Prerequisites for MPLS Virtual Private Networks

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

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

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

Restrictions for MPLS Virtual Private Networks

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

Supported Static Routes in an MPLS Environment

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

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

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

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

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

Unsupported Static Routes in an MPLS Environment That Uses the TFIB

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

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

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

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

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

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

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

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

Supported Static Routes in an MPLS VPN Environment

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The following **ip route vrf** command is supported when you configure static routes in an MPLS VPN environment, and the next hop is in the global table on the customer edge (CE) side. For example, the following

command is supported when the destination prefix is the CE device's loopback address, as in external Border Gateway Protocol (EBGP) multihop cases.

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

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

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

Information About MPLS Virtual Private Networks

MPLS Virtual Private Network Definition

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

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

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

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

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

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

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

The figure below shows a basic MPLS VPN.

Figure 1: Basic MPLS VPN Terminology
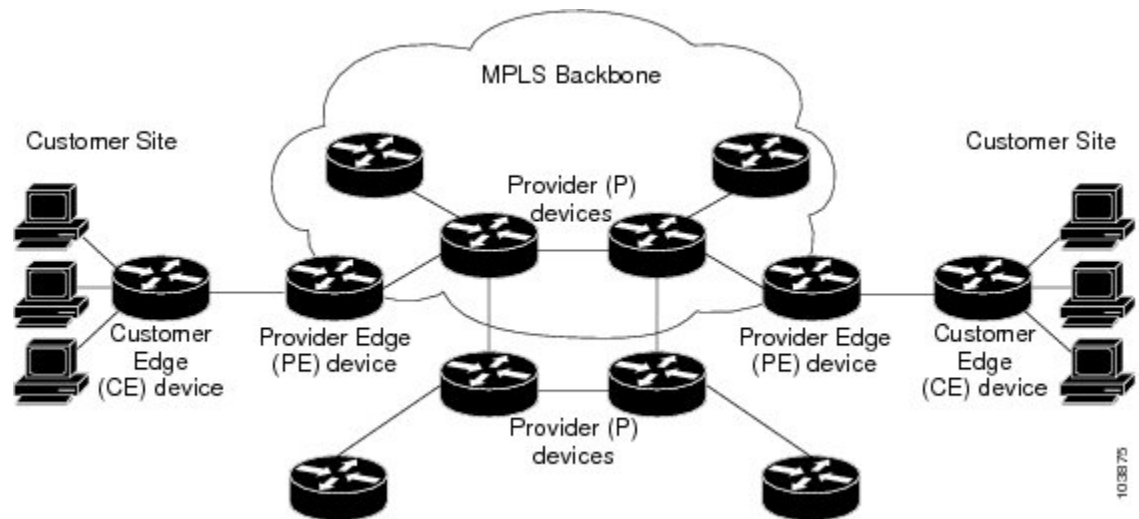
How an MPLS Virtual Private Network Works

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

- Exchanges routing updates with the customer edge (CE) device.

- Translates the CE routing information into VPNv4 routes.

- Exchanges VPNv4 routes with other PE devices through the Multiprotocol Border Gateway Protocol (MP-BGP).

The following sections describe how MPLS VPN works:

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

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

- An IP routing table

- A derived Cisco Express Forwarding table

- A set of interfaces that use the forwarding table

- A set of rules and routing protocol parameters that control the information that is included in the routing table

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

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

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

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

- When a VPN route that is learned from a customer edge (CE) device is injected into BGP, a list of VPN route target extended community attributes is associated with it. Typically the list of route target community extended values is set from an export list of route targets associated with the virtual routing and forwarding (VRF) instance from which the route was learned.

- An import list of route target extended communities is associated with each VRF. The import list defines route target extended community attributes that a route must have in order for the route to be imported into the VRF. For example, if the import list for a particular VRF includes route target extended communities A, B, and C, then any VPN route that carries any of those route target extended communities—A, B, *or* C—is imported into the VRF.

MPLS Forwarding

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

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

- The top label directs the packet to the correct PE device.

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

Major Components of an MPLS Virtual Private Network

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

- VPN route target communities—A VPN route target community is a list of all members of a VPN community. VPN route targets need to be configured for each VPN community member.

- Multiprotocol BGP (MP-BGP) peering of VPN community provider edge (PE) devices—MP-BGP propagates virtual routing and forwarding (VRF) reachability information to all members of a VPN community. MP-BGP peering must be configured on all PE devices within a VPN community.

- MPLS forwarding—MPLS transports all traffic between all VPN community members across a VPN service-provider network.

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

Benefits of an MPLS Virtual Private Network

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

Connectionless Service

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

Centralized Service

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

- Multicast

- Quality of service (QoS)

- Telephony support within a VPN

- Centralized services including content and web hosting to a VPN

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

Scalability

If you create a VPN using connection-oriented, point-to-point overlays, Frame Relay, or ATM virtual connections (VCs), the VPN's key deficiency is scalability. Specifically, connection-oriented VPNs without fully meshed connections between customer sites are not optimal. MPLS-based VPNs, instead, use the peer model and Layer 3 connectionless architecture to leverage a highly scalable VPN solution. The peer model

requires a customer site to peer with only one provider edge (PE) device as opposed to all other customer edge (CE) devices that are members of the VPN. The connectionless architecture allows the creation of VPNs in Layer 3, eliminating the need for tunnels or VCs.

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

- PE devices must maintain VPN routes for those VPNs who are members.

- P devices do not maintain any VPN routes.

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

Security

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

Security is provided in the following areas:

- At the edge of a provider network, ensuring packets received from a customer are placed on the correct VPN.

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

Ease of Creation

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

Flexible Addressing

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

Integrated QoS Support

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

- Predictable performance and policy implementation

- Support for multiple levels of service in an MPLS VPN

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

Straightforward Migration

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

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

How to Configure MPLS Virtual Private Networks

Configuring the Core Network

Assessing the Needs of MPLS Virtual Private Network Customers

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	Identify the size of the network.	Identify the following to determine the number of devices and ports that you need: • How many customers do you need to support? • How many VPNs are needed per customer? • How many virtual routing and forwarding instances are there for each VPN?

	Command or Action	**Purpose**
Step 2	Identify the routing protocols in the core.	Determine which routing protocols you need in the core network.
Step 3	Determine if you need MPLS VPN High Availability support.	MPLS VPN Nonstop Forwarding and Graceful Restart are supported on select devices and Cisco software releases. Contact Cisco Support for the exact requirements and hardware support.
Step 4	Determine if you need Border Gateway Protocol (BGP) load sharing and redundant paths in the MPLS VPN core.	For configuration steps, see the "Load Sharing MPLS VPN Traffic" feature module in the *MPLS Layer 3 VPNs Inter-AS and CSC Configuration Guide*.

Configuring MPLS in the Core

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

- MPLS Label Distribution Protocol (LDP). For configuration information, see the "MPLS Label Distribution Protocol (LDP)" module in the *MPLS Label Distribution Protocol Configuration Guide*.

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

Connecting the MPLS Virtual Private Network Customers

Defining VRFs on the PE Devices to Enable Customer Connectivity

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf vpn1`	Defines the virtual private network (VPN) routing instance by assigning a virtual routing and forwarding (VRF) name and enters VRF configuration mode. • The *vrf-name* argument is the name assigned to a VRF.

Step 4	**rd** *route-distinguisher*
	
	Example:
	`Device(config-vrf)# rd 100:1`
		Creates routing and forwarding tables.
		
		• The *route-distinguisher* argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter a route distinguisher (RD) in either of these formats:
		
		   ◦ 16-bit AS number:your 32-bit number, for example, 101:3
		
		   ◦ 32-bit IP address:your 16-bit number, for example, 10.0.0.1:1

Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Device(config-vrf)# route-target both 100:1`	Creates a route-target extended community for a VRF. • The **import** keyword imports routing information from the target VPN extended community. • The **export** keyword exports routing information to the target VPN extended community. • The **both** keyword imports routing information from and exports routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both route-target extended communities.
Step 6	**exit** **Example:** `Device(config-vrf)# exit`	(Optional) Exits to global configuration mode.

Configuring VRF Interfaces on PE Devices for Each VPN Customer

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip vrf forwarding** *vrf-name*
5. **end**

DETAILED STEPS

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

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

Configuring Routing Protocols Between the PE and CE Devices

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

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

SUMMARY STEPS

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

DETAILED STEPS

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

	Command or Action	**Purpose**
Step 3	**router rip** **Example:** Device(config)# router rip	Enables the Routing Information Protocol (RIP).
Step 4	**version** {**1**	**2**} **Example:** Device(config-router)# version 2	Specifies RIP version used globally by the device.
Step 5	**address-family ipv4** [**multicast**	**unicast**	**vrf** *vrf-name*] **Example:** Device(config-router)# address-family ipv4 vrf vpn1	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specifies the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**network** *ip-address* **Example:** Device(config-router-af)# network 192.168.7.0	Enables RIP on the PE-to-CE link.
Step 7	**redistribute** *protocol* [*process-id*] {**level-1**	**level-1-2**	**level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal**	**external 1**	**external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**] **Example:** Device(config-router-af)# redistribute bgp 200	Redistributes routes from one routing domain into another routing domain. • For the RIPv2 routing protocol, use the **redistribute bgp** *as-number* command.
Step 8	**exit-address-family** **Example:** Device(config-router-af)# exit-address-family	Exits address family configuration mode.
Step 9	**end** **Example:** Device(config-router)# end	(Optional) Exits to privileged EXEC mode.

Configuring Static Routes Between the PE and CE Devices

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route vrf** *vrf-name* **Example:** `Device(config)# ip route vrf vpn1`	Defines static route parameters for every provider edge-to-customer edge (PE-to-CE) session and enters router configuration mode.
Step 4	**address-family ipv4** [**multicast**	**unicast**	**vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 vrf vpn1`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes.

	Command or Action	Purpose
		• The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**] **Example:** Device(config-router-af)# redistribute static	Redistributes routes from one routing domain into another routing domain. • To redistribute virtual routing and forwarding (VRF) static routes into the VRF Border Gateway Protocol (BGP) table, use the **redistribute static** command. See the command reference page for information about other arguments and keywords.
Step 6	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**] **Example:** Device(config-router-af)# redistribute connected	Redistributes routes from one routing domain into another routing domain. • To redistribute directly connected networks into the VRF BGP table, use the **redistribute connected** command.
Step 7	**exit-address-family** **Example:** Device(config-router-af)# exit-address-family	Exits address family configuration mode.
Step 8	**end** **Example:** Device(config-router)# end	(Optional) Exits to privileged EXEC mode.

Verifying the Virtual Private Network Configuration

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

SUMMARY STEPS

1. **show ip vrf**

DETAILED STEPS

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

Verifying Connectivity Between MPLS Virtual Private Network Sites

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

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

SUMMARY STEPS

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

DETAILED STEPS

Step 1   **enable**
Enables privileged EXEC mode.

Step 2   **ping** [*protocol*] {*host-name* | *system-address*}
Diagnoses basic network connectivity on AppleTalk, Connectionless-mode Network Service (CLNS), IP, Novell, Apollo, Virtual Integrated Network Service (VINES), DECnet, or Xerox Network Service (XNS) networks. Use the **ping** command to verify the connectivity from one CE device to another.

Step 3   **trace** [*protocol*] [*destination*]
Discovers the routes that packets take when traveling to their destination. The **trace** command can help isolate a trouble spot if two devices cannot communicate.

Step 4   **show ip route** [*ip-address* [*mask*] [**longer-prefixes**]] | *protocol* [*process-id*]] | [**list** [*access-list-name* | *access-list-number*]
Displays the current state of the routing table. Use the *ip-address* argument to verify that CE1 has a route to CE2. Verify the routes learned by CE1. Make sure that the route for CE2 is listed.

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

SUMMARY STEPS

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

DETAILED STEPS

Step 1   **enable**
Enables privileged EXEC mode.

Step 2   **show ip route vrf** *vrf-name* [*prefix*]
Displays the IP routing table associated with a virtual routing and forwarding (VRF) instance. Check that the loopback addresses of the local and remote customer edge (CE) devices are in the routing table of the provider edge (PE) devices.

Step 3   **show ip cef vrf** *vrf-name* [*ip-prefix*]
Displays the Cisco Express Forwarding forwarding table associated with a VRF. Check that the prefix of the remote CE device is in the Cisco Express Forwarding table.

Configuration Examples for MPLS Virtual Private Networks

Example: Configuring an MPLS Virtual Private Network Using RIP

PE Configuration	CE Configuration

<pre>ip vrf vpn1
rd 100:1
route-target export 100:1
route-target import 100:1
!
ip cef
mpls ldp router-id Loopback0 force
mpls label protocol ldp
!
interface Loopback0
ip address 10.0.0.1 255.255.255.255
!
interface
ip vrf forwarding vpn1
ip address 192.0.2.3 255.255.255.0
no cdp enable
interface
ip address 192.0.2.2 255.255.255.0
mpls label protocol ldp
mpls ip
!
router rip
version 2
timers basic 30 60 60 120
!
address-family ipv4 vrf vpn1
version 2
redistribute bgp 100 metric transparent
network 192.0.2.0
distribute-list 20 in
no auto-summary
exit-address-family
!
router bgp 100
no synchronization
bgp log-neighbor changes
neighbor 10.0.0.3 remote-as 100
neighbor 10.0.0.3 update-source Loopback0
no auto-summary
!
address-family vpnv4
neighbor 10.0.0.3 activate
neighbor 10.0.0.3 send-community extended
bgp scan-time import 5
exit-address-family
!
address-family ipv4 vrf vpn1
redistribute connected
redistribute rip
no auto-summary
no synchronization
exit-address-family</pre>
	<pre>ip cef
	mpls ldp router-id Loopback0 force
	mpls label protocol ldp
	!
	interface Loopback0
	ip address 10.0.0.9 255.255.255.255
	!
	interface
	ip address 192.0.2.1 255.255.255.0
	no cdp enable
	router rip
	version 2
	timers basic 30 60 60 120
	redistribute connected
	network 10.0.0.0
	network 192.0.2.0
	no auto-summary</pre>

Example: Configuring an MPLS Virtual Private Network Using Static Routes

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

Additional References

Related Documents

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

Technical Assistance

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

Feature Information for MPLS Virtual Private Networks

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

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

Table 1: Feature Information for MPLS Virtual Private Networks

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