Multiprotocol Label Switching (MPLS) Configuration Guide, Cisco IOS XE Everest 16.6.x (Catalyst 9400 Switches)

First Published: 2017-09-07

Last Modified: 2017-10-30

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

CHAPTER 1

Configuring Multiprotocol Label Switching (MPLS)

Multiprotocol Label Switching

This module describes Multiprotocol Label Switching and how to configure it on Cisco switches.

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 at the end of this module.

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.

Restrictions for Multiprotocol Label Switching

- Multiprotocol Label Switching (MPLS) fragmentation is not supported.
- MPLS maximum transmission unit (MTU) is not supported.

Information about Multiprotocol Label Switching

Multiprotocol label switching (MPLS) combines the performance and capabilities of Layer 2 (data link layer) switching with the proven scalability of Layer 3 (network layer) routing. MPLS enables you to meet the challenges of explosive growth in network utilization while providing the opportunity to differentiate services without sacrificing the existing network infrastructure. The MPLS architecture is flexible and can be employed in any combination of Layer 2 technologies. MPLS support is offered for all Layer 3 protocols, and scaling is possible well beyond that typically offered in today's networks.

Functional Description of Multiprotocol Label Switching

Label switching is a high-performance packet forwarding technology that integrates the performance and traffic management capabilities of data link layer (Layer 2) switching with the scalability, flexibility, and performance of network layer (Layer 3) routing.

Label Switching Functions

In conventional Layer 3 forwarding mechanisms, as a packet traverses the network, each switch extracts all the information relevant to forwarding the packet from the Layer 3 header. This information is then used as an index for a routing table lookup to determine the next hop for the packet.

In the most common case, the only relevant field in the header is the destination address field, but in some cases, other header fields might also be relevant. As a result, the header analysis must be done independently at each switch through which the packet passes. In addition, a complicated table lookup must also be done at each switch.

In label switching, the analysis of the Layer 3 header is done only once. The Layer 3 header is then mapped into a fixed length, unstructured value called a *label* .

Many different headers can map to the same label, as long as those headers always result in the same choice of next hop. In effect, a label represents a *forwarding equivalence class* --that is, a set of packets which, however different they may be, are indistinguishable by the forwarding function.

The initial choice of a label need not be based exclusively on the contents of the Layer 3 packet header; for example, forwarding decisions at subsequent hops can also be based on routing policy.

Once a label is assigned, a short label header is added at the front of the Layer 3 packet. This header is carried across the network as part of the packet. At subsequent hops through each MPLS switch in the network, labels are swapped and forwarding decisions are made by means of MPLS forwarding table lookup for the label carried in the packet header. Hence, the packet header does not need to be reevaluated during packet transit through the network. Because the label is of fixed length and unstructured, the MPLS forwarding table lookup process is both straightforward and fast.

Distribution of Label Bindings

Each label switching router (LSR) in the network makes an independent, local decision as to which label value to use to represent a forwarding equivalence class. This association is known as a label binding. Each LSR informs its neighbors of the label bindings it has made. This awareness of label bindings by neighboring switches is facilitated by the following protocols:

- Label Distribution Protocol (LDP)--enables peer LSRs in an MPLS network to exchange label binding information for supporting hop-by-hop forwarding in an MPLS network
- Border Gateway Protocol (BGP)--Used to support MPLS virtual private networks (VPNs)

When a labeled packet is being sent from LSR A to the neighboring LSR B, the label value carried by the IP packet is the label value that LSR B assigned to represent the forwarding equivalence class of the packet. Thus, the label value changes as the IP packet traverses the network.

For more information about LDP configuration, see the see MPLS: LDP Configuration Guide at http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mpls/config_library/xe-3s/mp-xe-3s-library.html

Note As the scale of label entries is limited in, especially with ECMP, it is recommended to enable LDP label filtering. LDP labels shall be allocated only for well known prefixes like loopback interfaces of routers and any prefix that needs to be reachable in the global routing table.

MPLS Layer 3 VPN

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

Before configuring MPLS Layer 3 VPNs, you should have MPLS, Label Distribution Protocol (LDP), and Cisco Express Forwarding (CEF) installed in your network. All routers in the core, including the PE routers, must be able to support CEF and MPLS forwarding.

Classifying and Marking MPLS QoS EXP

The QoS EXP Matching feature allows you to classify and mark network traffic by modifying the Multiprotocol Label Switching (MPLS) experimental bits (EXP) field in IP packets.

The QoS EXP Matching feature allows you to organize network traffic by setting values for the MPLS EXP field in MPLS packets. By choosing different values for the MPLS EXP field, you can mark packets so that packets have the priority that they require during periods of congestion. Setting the MPLS EXP value allows you to:

- **Classify traffic:** The classification process selects the traffic to be marked. Classification accomplishes this by partitioning traffic into multiple priority levels, or classes of service. Traffic classification is the primary component of class-based QoS provisioning.
- **Police and mark traffic**: Policing causes traffic that exceeds the configured rate to be discarded or marked to a different drop level. Marking traffic is a way to identify packet flows to differentiate them. Packet marking allows you to partition your network into multiple priority levels or classes of service.

Restrictions

Following is the list of restrictions for classifying and marking MPLS QoS EXP:

- Only Uniform mode and Pipe mode are supported; Short-pipe mode is not supported.
- Support range of QoS-group values range between 0 and 30. (Total 31 QoS-groups).

- EXP marking using QoS policy is supported only on the outer label; inner EXP marking is not supported.

How to Configure Multiprotocol Label Switching

This section explains how to perform the basic configuration required to prepare a switch for MPLS switching and forwarding.

Configuring a Switch for MPLS Switching (CLI)

MPLS switching on Cisco switches requires that Cisco Express Forwarding be enabled.

Note **ip unnumbered** command is not supported in MPLS configuration.

Procedure

	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 cef distributed** **Example:** `Device(config)# ip cef distributed`	Enables Cisco Express Forwarding on the switch.
Step 4	**mpls label range** *minimum-value maximum-value* **Example:** `Device(config)# mpls label range 16 4096`	Configure the range of local labels available for use with MPLS applications on packet interfaces.
Step 5	**mpls label protocol ldp** **Example:** `Device(config)# mpls label protocol ldp`	Specifies the label distribution protocol for the platform.

Configuring a Switch for MPLS Forwarding (CLI)

MPLS forwarding on Cisco switches requires that forwarding of IPv4 packets be enabled.

Note **ip unnumbered** command is not supported in MPLS configuration.

Procedure

	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 slot/subslot /port* **Example:** `Device(config)# interface gigabitethernet 1/0/0`	Specifies the Gigabit Ethernet interface and enters interface configuration mode. For Switch Virtual Interface (SVI), the example is `Device(config)# interface vlan 1000`
Step 4	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables MPLS forwarding of IPv4 packets along routed physical interfaces (Gigabit Ethernet), Switch Virtual Interface (SVI), or port channels.
Step 5	**mpls label protocol ldp** **Example:** `Device(config-if)# mpls label protocol ldp`	Specifies the label distribution protocol for an interface. **Note** MPLS LDP cannot be enabled on a Virtual Routing and Forwarding (VRF) interface.

Step 6	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Verifying Multiprotocol Label Switching Configuration

This section explains how to verify successful configuration of MPLS switching and forwarding.

Verifying Configuration of MPLS Switching

To verify that Cisco Express Forwarding has been configured properly, issue the **show ip cef summary** command, which generates output similar to that shown below:

Procedure

show ip cef summary

Example:

```
Switch# show ip cef summary

IPv4 CEF is enabled for distributed and running
VRF Default
 150 prefixes (149/1 fwd/non-fwd)
 Table id 0x0
 Database epoch:        4 (150 entries at this epoch)
Switch#
```

Verifying Configuration of MPLS Forwarding

To verify that MPLS forwarding has been configured properly, issue the **show mpls interfaces detail** command, which generates output similar to that shown below:

Note The MPLS MTU value is equivalent to the IP MTU value of the port or switch by default. MTU configuration for MPLS is not supported.

Procedure

Step 1 **show mpls interfaces detail**

Example:

```
For physical (Gigabit Ethernet) interface:
Switch# show mpls interfaces detail interface GigabitEthernet 1/0/0

        Type Unknown
        IP labeling enabled
        LSP Tunnel labeling not enabled
        IP FRR labeling not enabled
        BGP labeling not enabled
        MPLS not operational
        MTU = 1500

For Switch Virtual Interface (SVI):
Switch# show mpls interfaces detail interface Vlan1000

        Type Unknown
        IP labeling enabled (ldp) :
          Interface config
        LSP Tunnel labeling not enabled
        IP FRR labeling not enabled
        BGP labeling not enabled
        MPLS operational
        MTU = 1500
```

Step 2 **show running-config interface**

Example:

```
For physical (Gigabit Ethernet) interface:
Switch# show running-config interface interface GigabitEthernet 1/0/0

Building configuration...


Current configuration : 307 bytes
!
interface TenGigabitEthernet1/0/0
no switchport
ip address xx.xx.x.x xxx.xxx.xxx.x
mpls ip
mpls label protocol ldp
end

For Switch Virtual Interface (SVI):
Switch# show running-config interface interface Vlan1000

Building configuration...


Current configuration : 187 bytes
!
interface Vlan1000
ip address xx.xx.x.x xxx.xxx.xxx.x
mpls ip
mpls label protocol ldp
end
```

Step 3 **show mpls forwarding**

Example:

```
For physical (Gigabit Ethernet) interface:
Switch#show mpls forwarding-table
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
500        No Label   l2ckt(3)         0             Gi3/0/22   point2point
501        No Label   l2ckt(1)         12310411816789 none       point2point
502        No Label   l2ckt(2)         0             none       point2point
503        566        15.15.15.15/32   0             Po5        192.1.1.2
504        530        7.7.7.7/32       538728528     Po5        192.1.1.2
505        573        6.6.6.10/32      0             Po5        192.1.1.2
```

```
506        606        6.6.6.6/32       0             Po5        192.1.1.2
507        explicit-n 1.1.1.1/32       0             Po5        192.1.1.2
556        543        19.10.1.0/24     0             Po5        192.1.1.2
567        568        20.1.1.0/24      0             Po5        192.1.1.2
568        574        21.1.1.0/24      0             Po5        192.1.1.2
574        No Label   213.1.1.0/24[V]  0             aggregate/vpn113
575        No Label   213.1.2.0/24[V]  0             aggregate/vpn114
576        No Label   213.1.3.0/24[V]  0             aggregate/vpn115
577        No Label   213:1:1::/64     0             aggregate
594        502        103.1.1.0/24     0             Po5        192.1.1.2
595        509        31.1.1.0/24      0             Po5        192.1.1.2
596        539        15.15.1.0/24     0             Po5        192.1.1.2
597        550        14.14.1.0/24     0             Po5        192.1.1.2
633        614        2.2.2.0/24       0             Po5        192.1.1.2
634        577        90.90.90.90/32   873684        Po5        192.1.1.2
635        608        154.1.1.0/24     0             Po5        192.1.1.2
636        609        153.1.1.0/24     0             Po5        192.1.1.2
Switch#
end
```

Additional References for Multiprotocol Label Switching

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the Multiprotocol Label Switching (MPLS) Commands section of the *Command Reference (Catalyst 9400 Series Switches)*

Technical Assistance

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

Feature Information for Multiprotocol Label Switching

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 Multiprotocol Label Switching

Release	Modification
	This feature was introduced.

CHAPTER 2

Configuring MPLS Layer 3 VPN

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 Layer 3 VPN.

MPLS Layer 3 VPNs

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 at the end of this module.

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.

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. Use the following guidelines when configuring static routes.

Supported Static Routes in an MPLS Environment

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

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

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

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

Unsupported Static Routes in an MPLS Environment That Uses the TFIB

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

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

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

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

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

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

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

Supported Static Routes in an MPLS VPN Environment

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Information About MPLS Virtual Private Networks

MPLS Virtual Private Network Definition

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

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

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

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

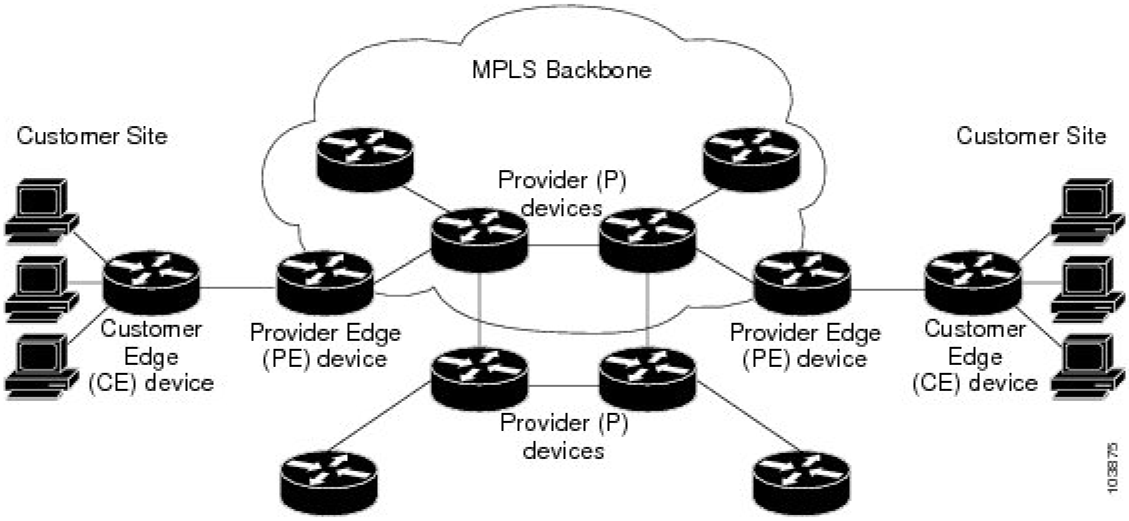
MPLS VPNs are easier to manage and expand than conventional VPNs. When a new site is added to an MPLS VPN, only the service provider's edge device that provides services to the customer site needs to be updated.

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

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

The figure below shows a basic MPLS VPN.

Figure 1: Basic MPLS VPN Terminology


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

How an MPLS Virtual Private Network Works

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

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

The following sections describe how MPLS VPN works:

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.

Procedure

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

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*.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vrf1`	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	**address-family** *ipv4* | *ipv6* **Example:** `Device(config-vrf)# address-family ipv6`	Enters IPv4 or IPv6 address family mode
Step 6	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:**	Creates a route-target extended community for a VRF.

	Command or Action	Purpose
	`Device(config-vrf-af)# route-target both 100:1`	• 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 7	**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.

Procedure

	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 GigabitEthernet 0/0/1`	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.

	Command or Action	Purpose
Step 4	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding vrf1`	Associates a VRF with the specified interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.

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

Configuring Routing Protocols Between the PE and CE Devices

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

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.

Procedure

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

Procedure

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

Procedure

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
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 GigabitEthernet 1/0/1 vrf forwarding vpn1 ip address 192.0.2.3 255.255.255.0 no cdp enable interface GigabitEthernet 1/0/1 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	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 GigabitEthernet 1/0/1 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

Example: Configuring an MPLS Virtual Private Network Using Static Routes

PE Configuration	CE Configuration
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 GigabitEthernet 1/0/1 vrf forwarding vpn1 ip address 192.0.2.3 255.255.255.0 no cdp enable ! interface GigabitEthernet 1/0/1 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 GigabitEthernet 1/0/1 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
For complete syntax and usage information for the commands used in this chapter.	See the MPLS Commands section of the *Command Reference (Catalyst 9400 Series Switches)*
Configuring Cisco Express Forwarding	"Configuring Basic Cisco Express Forwarding" module in the *Cisco Express Forwarding Configuration Guide*
Configuring LDP	"MPLS Label Distribution Protocol (LDP)" module in the *MPLS Label Distribution Protocol 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 2: Feature Information for MPLS Virtual Private Networks

Release	Modification
	This feature was introduced.

CHAPTER 3

Configuring MPLS QoS

Classifying and Marking MPLS EXP

The QoS EXP Matching feature allows you to classify and mark network traffic by modifying the Multiprotocol Label Switching (MPLS) experimental bits (EXP) field. This module contains conceptual information and the configuration tasks for classifying and marking network traffic using the MPLS EXP field.

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 at the end of this module.

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.

Prerequisites for Classifying and Marking MPLS EXP

- The switch must be configured as an MPLS provider edge (PE) or provider (P) router, which can include the configuration of a valid label protocol and underlying IP routing protocols.

Restrictions for Classifying and Marking MPLS EXP

- MPLS classification and marking can only occur in an operational MPLS Network.
- MPLS EXP classification and marking is supported only on MPLS enabled interfaces or MPLS traffic on other interfaces.
- If a packet is classified by IP type of service (ToS) or class of service (CoS) at ingress, it cannot be reclassified by MPLS EXP at egress (imposition case). However, if a packet is classified by MPLS at ingress it can be reclassified by IP ToS, CoS, or Quality of Service (QoS) group at egress (disposition case).

- To apply QoS on traffic across protocol boundaries, use QoS-group. You can classify and assign ingress traffic to the QoS-group. Thereafter, you can the QoS-group at egress to classify and apply QoS.
- If a packet is encapsulated in MPLS, the MPLS payload cannot be checked for other protocols such as IP for classification or marking. Only MPLS EXP marking affects packets encapsulated by MPLS.

Information About Classifying and Marking MPLS EXP

Classifying and Marking MPLS EXP Overview

The QoS EXP Matching feature allows you to organize network traffic by setting values for the MPLS EXP field in MPLS packets. By choosing different values for the MPLS EXP field, you can mark packets so that packets have the priority that they require during periods of congestion. Setting the MPLS EXP value allows you to:

- Classify traffic

  The classification process selects the traffic to be marked. Classification accomplishes this by partitioning traffic into multiple priority levels, or classes of service. Traffic classification is the primary component of class-based QoS provisioning. For more information, see the “Classifying Network Traffic” module.

- Police and mark traffic

  Policing causes traffic that exceeds the configured rate to be discarded or marked to a different drop level. Marking traffic is a way to identify packet flows to differentiate them. Packet marking allows you to partition your network into multiple priority levels or classes of service. For more information, see the “Marking Network Traffic” module.

MPLS Experimental Field

The MPLS experimental bits (EXP) field is a 3-bit field in the MPLS header that you can use to define the QoS treatment (per-hop behavior) that a node should give to a packet. In an IP network, the DiffServ Code Point (DSCP) (a 6-bit field) defines a class and drop precedence. The EXP bits can be used to carry some of the information encoded in the IP DSCP and can also be used to encode the dropping precedence.

By default, Cisco IOS Software copies the three most significant bits of the DSCP or the IP precedence of the IP packet to the EXP field in the MPLS header. This action happens when the MPLS header is initially imposed on the IP packet. However, you can also set the EXP field by defining a mapping between the DSCP or IP precedence and the EXP bits. This mapping is configured using the **set mpls experimental** or **police** commands. For more information, see the “How to Classify and Mark MPLS EXP” section.

You can perform MPLS EXP marking operations using table-maps. It is recommended to assign QoS-group to a different class of traffic in ingress policy and translate QoS-group to DSCP and EXP markings in egress policy using table-map.

Benefits of MPLS EXP Classification and Marking

If a service provider does not want to modify the value of the IP precedence field in packets transported through the network, they can use the MPLS EXP field value to classify and mark IP packets.

By choosing different values for the MPLS EXP field, you can mark critical packets so that those packets have priority if network congestion occurs.

How to Classify and Mark MPLS EXP

Classifying MPLS Encapsulated Packets

You can use the **match mpls experimental topmost** command to define traffic classes based on the packet EXP values, inside the MPLS domain. You can use these classes to define services policies to mark the EXP traffic using the **police** command.

Procedure

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

Step 3	**class-map** [**match-all** | **match-any**] *class-map-name*
	Example:
	`Switch(config)# class-map exp3`
		Creates a class map to be used for matching traffic to a specified class, and enters class-map configuration mode.
		• Enter the class map name.

Step 4	**match mpls experimental topmost** *mpls-exp-value* **Example:** `Switch(config-cmap)# match mpls experimental topmost 3`	Specifies the match criteria. **Note** The **match mpls experimental topmost** command classifies traffic on the basis of the EXP value in the topmost label header.

Step 5	**end**
	Example:
	`Switch(config-cmap)# end`
		(Optional) Returns to privileged EXEC mode.

Marking MPLS EXP on the Outermost Label

Perform this task to set the value of the MPLS EXP field on imposed label entries.

Before you begin

In typical configurations, marking MPLS packets at imposition is used with ingress classification on IP ToS or CoS fields.

Note For IP imposition marking, the IP precedence value is copied to the MPLS EXP value by default.

Note The **set mpls experimental imposition** command works only on packets that have new or additional MPLS labels added to them.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Switch(config)# policy-map mark-up-exp-2`	Specifies the name of the policy map to be created and enters policy-map configuration mode. • Enter the policy map name.

Step 4	**class** *class-map-name*
	Example:
	`Switch(config-pmap)# class prec012`
		Creates a class map to be used for matching traffic to a specified class, and enters class-map configuration mode.
		• Enter the class map name.

Step 5	**set mpls experimental imposition** *mpls-exp-value* **Example:** `Switch(config-pmap-c)# set mpls experimental imposition 2`	Sets the value of the MPLS EXP field on top label.

Step 6	**end**
	Example:
	`Switch(config-pmap-c)# end`
		(Optional) Returns to privileged EXEC mode.

Marking MPLS EXP on Label Switched Packets

Perform this task to set the MPLS EXP field on label switched packets.

Before you begin

Note The **set mpls experimental topmost** command marks EXP for the outermost label of MPLS traffic. Due to this marking at ingress policy, the egress policy must include classification based on the MPLS EXP values.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Switch(config)# policy-map mark-up-exp-2`	Specifies the name of the policy map to be created and enters policy-map configuration mode. • Enter the policy map name.

Step 4	**class** *class-map-name*
	Example:
	`Switch(config-pmap)# class-map exp012`
		Creates a class map to be used for matching traffic to a specified class, and enters class-map configuration mode.
		• Enter the class map name.

Step 5	**set mpls experimental topmost** *mpls-exp-value* **Example:** `Switch(config-pmap-c)# set mpls experimental topmost 2`	Sets the MPLS EXP field value in the topmost label on the output interface.

Step 6	**end**
	Example:
	`Switch(config-pmap-c)# end`
		(Optional) Returns to privileged EXEC mode.

Configuring Conditional Marking

To conditionally set the value of the MPLS EXP field on all imposed label, perform the following task:

Before you begin

Note The **set-mpls-exp-topmost-transmit** action affects MPLS encapsulated packets only. The **set-mpls-exp-imposition-transmit** action affects any new labels that are added to the packet.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Switch(config)# policy-map ip2tag`	Specifies the name of the policy map to be created and enters policy-map configuration mode. • Enter the policy map name.

Step 4	**class** *class-map-name*
	Example:
	`Switch(config-pmap)# class iptcp`
		Creates a class map to be used for matching traffic to a specified class, and enters policy-map class configuration mode.
		• Enter the class map name.

Step 5	**police cir** *bps* **bc pir** *bps* **be**
	Example:
	`Switch(config-pmap-c)# police cir 1000000`
	` pir 2000000`
		Defines a policer for classified traffic and enters policy-map class police configuration mode.

Step 6	**conform-action transmit**
	Example:
	`Switch(config-pmap-c-police)#`
	`conform-action transmit 3`
		Defines the action to take on packets that conform to the values specified by the policer.
		• In this example, if the packet conforms to the committed information rate (cir) or is within the conform burst (bc) size, the MPLS EXP field is set to 3.

Step 7	**exceed-action set-mpls-exp-topmost-transmit dscp table** *dscp-table-value* **Example:**	Defines the action to take on packets that exceed the values specified by the policer.

	Command or Action	Purpose
	`Switch(config-pmap-c-police)# exceed-action set-mpls-exp-topmost-transmit dscp table dscp2exp`	
Step 8	**violate-action drop** **Example:** `Switch(config-pmap-c-police)# violate-action drop`	Defines the action to take on packets whose rate exceeds the peak information rate (pir) and is outside the bc and be ranges. • You must specify the exceed action before you specify the violate action. • In this example, if the packet rate exceeds the pir rate and is outside the bc and be ranges, the packet is dropped.

Step 9	**end**
	Example:
	`Switch(config-pmap-c-police)# end`
		(Optional) Returns to privileged EXEC mode.

Configuration Examples for Classifying and Marking MPLS EXP

Example: Classifying MPLS Encapsulated Packets

Defining an MPLS EXP Class Map

The following example defines a class map named exp3 that matches packets that contains MPLS experimental value 3:

```
Switch(config)# class-map exp3
Switch(config-cmap)# match mpls experimental topmost 3
Switch(config-cmap)# exit
```

Defining a Policy Map and Applying the Policy Map to an Ingress Interface

The following example uses the class map created in the example above to define a policy map. This example also applies the policy map to a physical interface for ingress traffic.

```
Switch(config)# policy-map change-exp-3-to-2
Switch(config-pmap)# class exp3
Switch(config-pmap-c)# set mpls experimental topmost 2
Switch(config-pmap)# exit
Switch(config)# interface GigabitEthernet 0/0/0
Switch(config-if)# service-policy input change-exp-3-to-2
Switch(config-if)# exit
```

Defining a Policy Map and Applying the Policy Map to an Egress Interface

The following example uses the class map created in the example above to define a policy map. This example also applies the policy map to a physical interface for egress traffic.

```
Switch(config)# policy-map WAN-out
Switch(config-pmap)# class exp3
Switch(config-pmap-c)# shape average 10000000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface GigabitEthernet 0/0/0
Switch(config-if)# service-policy output WAN-out
Switch(config-if)# exit
```

Marking MPLS EXP on the Outermost Label

Perform this task to set the value of the MPLS EXP field on imposed label entries.

Before you begin

In typical configurations, marking MPLS packets at imposition is used with ingress classification on IP ToS or CoS fields.

Note For IP imposition marking, the IP precedence value is copied to the MPLS EXP value by default.

Note The **set mpls experimental imposition** command works only on packets that have new or additional MPLS labels added to them.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Switch(config)# policy-map mark-up-exp-2`	Specifies the name of the policy map to be created and enters policy-map configuration mode. • Enter the policy map name.

	Command or Action	Purpose

Step 4	**class** *class-map-name*
	Example:
	`Switch(config-pmap)# class prec012`
		Creates a class map to be used for matching traffic to a specified class, and enters class-map configuration mode.
		• Enter the class map name.

Step 5	**set mpls experimental imposition** *mpls-exp-value* **Example:** `Switch(config-pmap-c)# set mpls experimental imposition 2`	Sets the value of the MPLS EXP field on top label.

Step 6	**end**
	Example:
	`Switch(config-pmap-c)# end`
		(Optional) Returns to privileged EXEC mode.

Example: Marking MPLS EXP on Label Switched Packets

Defining an MPLS EXP Label Switched Packets Policy Map

The following example defines a policy map that sets the MPLS EXP topmost value to 2 according to the MPLS EXP value of the forwarded packet:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# class-map exp012
Switch(config-cmap)# match mpls experimental topmost 0 1 2
Switch(config-cmap)# exit
Switch(config-cmap)# policy-map mark-up-exp-2
Switch(config-pmap)# class exp012
Switch(config-pmap-c)# set mpls experimental topmost 2
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
```

Applying the MPLS EXP Label Switched Packets Policy Map to a Main Interface

The following example shows how to apply the policy map to a main interface:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface GigabitEthernet 0/0/0
Switch(config-if)# service-policy input mark-up-exp-2
Switch(config-if)# exit
```

Example: Configuring Conditional Marking

The example in this section creates a policer for the **iptcp** class, which is part of the **ip2tag** policy map, and attaches the policy map to the Gigabit Ethernet interface.

```
Switch(config)# policy-map ip2tag
Switch(config-pmap)# class iptcp
Switch(config-pmap-c)# police cir 1000000 pir 2000000
Switch(config-pmap-c-police)# conform-action transmit
Switch(config-pmap-c-police)# exceed-action set-mpls-exp-imposition-transmit 2
Switch(config-pmap-c-police)# violate-action drop
Switch(config-pmap-c-police)# exit
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface GigabitEthernet 0/0/1
Switch(config-if)# service-policy input ip2tag
```

Additional References

Related Documents

Related Topic	Document Title
QoS commands	*Cisco IOS Quality of Service Solutions Command Reference*

Standards and RFCs

Standard/RFC	Title
No new or modified standards are supported, and support for existing standards has not been modified.	

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 QoS MPLS EXP

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 QoS MPLS EXP

Release	Modification
	This feature was introduced.

CHAPTER 4

Configuring MPLS Static Labels

MPLS Static Labels

This document describes the Cisco MPLS Static Labels feature. The MPLS Static Labels feature provides the means to configure the binding between a label and an IPv4 prefix statically.

Prerequisites for MPLS Static Labels

The network must support the following Cisco IOS features before you enable MPLS Static Labels:

- Multiprotocol Label Switching (MPLS)
- Cisco Express Forwarding

Restrictions for MPLS Static Labels

- On a provider edge (PE) router for MPLS VPNs, there's no mechanism for statically binding a label to a customer network prefix (VPN IPv4 prefix).
- MPLS Static Crossconnect is not supported.
- MPLS Static Labels is not supported for label-controlled Asynchronous Transfer Mode (lc-atm).
- MPLS static bindings are not supported for local prefixes.
- VRF aware Static Labels is not supported,

Information About MPLS Static Labels

MPLS Static Labels Overview

Generally, label switching routers (LSRs) dynamically learn the labels they should use to label-switch packets. They do this by means of label distribution protocols that include:

- Label Distribution Protocol (LDP), the Internet Engineering Task Force (IETF) standard, used to bind labels to network addresses.
- Resource Reservation Protocol (RSVP) used to distribute labels for traffic engineering (TE)
- Border Gateway Protocol (BGP) used to distribute labels for Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs)

To use a learned label to label-switch packets, an LSR installs the label into its Label Forwarding Information Base (LFIB).

The MPLS Static Labels feature provides the means to configure the binding between a label and an IPv4 prefix statically.

Benefits of MPLS Static Labels

Static Bindings Between Labels and IPv4 Prefixes

You can configure static bindings between labels and IPv4 prefixes to support MPLS hop-by-hop forwarding through neighbor routers that don’t implement LDP label distribution.

How to Configure MPLS Static Labels

Configuring MPLS Static Prefix Label Bindings

To configure MPLS static prefix/label bindings, use the following commands beginning in global configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls label range** *min-label max-label* [**static** *min-static-label max-static-label*] **Example:** `Device(config)# mpls label range 200 100000 static 16 199`	Specifies a range of labels for use with MPLS Static Labels feature. (Default is no labels reserved for static assignment.)
Step 4	**mpls static binding ipv4** *prefix mask* [**input**| **output** *nexthop*] label	Specifies static binding of labels to IPv4 prefixes.

	Command or Action	Purpose
	Example: `Device(config)# mpls static binding ipv4 10.0.0.0 255.0.0.0 55`	Bindings specified are installed automatically in the MPLS forwarding table as routing demands.

Verifying MPLS Static Prefix Label Bindings

To verify the configuration for MPLS static prefix/label bindings, use this procedure:

Procedure

Step 1 Enter **show mpls label range** command. The output shows that the new label ranges do not take effect until a reload occurs:

Example:

```
Device# show mpls label range

Downstream label pool: Min/Max label: 16/100000
   [Configured range for next reload: Min/Max label: 200/100000]
Range for static labels: Min/Max/Number: 16/199
```

The following output from the **show mpls label range** command, executed after a reload, indicates that the new label ranges are in effect:

Example:

```
Device# show mpls label range

Downstream label pool: Min/Max label: 200/100000
Range for static labels: Min/Max/Number: 16/199
```

Step 2 Enter the **show mpls static binding ipv4** command to show the configured static prefix/label bindings:

Example:

```
Device# show mpls static binding ipv4
10.17.17.17/32: Incoming label: 251 (in LIB)
  Outgoing labels:
     10.0.0.1                  18
10.18.18.18/32: Incoming label: 201 (in LIB)
  Outgoing labels:
10.0.0.1 implicit-null
```

Step 3 Use the **show mpls forwarding-table** command to determine which static prefix/label bindings are currently in use for MPLS forwarding.

Example:

```
Device# show mpls forwarding-table
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
201    Pop tag     10.18.18.18/32    0          PO1/1/0    point2point
```

```
       2/35        10.18.18.18/32    0          AT4/1/0.1  point2point
251    18          10.17.17.17/32    0          PO1/1/0    point2point
```

Monitoring and Maintaining MPLS Static Labels

To monitor and maintain MPLS Static Labels, use one or more of the following commands:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Devie> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**show mpls forwarding-table** **Example:** `Device# show mpls forwarding-table`	Displays the contents of the MPLS LFIB.
Step 3	**show mpls label range** **Example:** `Device# show mpls label range`	Displays information about the static label range.
Step 4	**show mpls static binding ipv4** **Example:** `Device# show mpls static binding ipv4`	Displays information about the configured static prefix/label bindings.

Configuration Examples for MPLS Static Labels

Example: Configuring MPLS Static Prefixes Labels

In the following output, the **mpls label range** command reconfigures the range used for dynamically assigned labels 16–100000 to 200–100000. It configures a static label range of 16–199.

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mpls label range 200 100000 static 16 199
% Label range changes take effect at the next reload.
Router(config)# end
```

In the following output, the **show mpls label range** command indicates that the new label ranges don’t take effect until a reload occurs:

```
Device# show mpls label range
```

```
Downstream label pool: Min/Max label: 16/100000
   [Configured range for next reload: Min/Max label: 200/100000]
Range for static labels: Min/Max/Number: 16/199
```

In the following output, the **show mpls label range** command, executed after a reload, indicates that the new label ranges are in effect:

```
Device# show mpls label range

Downstream label pool: Min/Max label: 200/100000
Range for static labels: Min/Max/Number: 16/199
```

In the following output, the **mpls static binding ipv4** commands configure static prefix/label bindings. They also configure input (local) and output (remote) labels for various prefixes:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# mpls static binding ipv4 10.0.0.0 255.0.0.0 55
Device(config)# mpls static binding ipv4 10.0.0.0 255.0.0.0 output 10.0.0.66 2607
Device(config)# mpls static binding ipv4 10.6.0.0 255.255.0.0 input 17
Device(config)# mpls static binding ipv4 10.0.0.0 255.0.0.0 output 10.13.0.8 explicit-null
Device(config)# end
```

In the following output, the **show mpls static binding ipv4** command displays the configured static prefix/label bindings:

```
Device# show mpls static binding ipv4

10.0.0.0/8: Incoming label: none;
  Outgoing labels:
10.13.0.8           explicit-null
10.0.0.0/8: Incoming label: 55 (in LIB)
  Outgoing labels:
     10.0.0.66              2607
10.66.0.0/16: Incoming label: 17 (in LIB)
  Outgoing labels:  None
```

Additional References

Related Documents

Related Topic	Document Title
MPLS commands	*Multiprotocol Label Switching Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature. 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 Static Labels

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

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

Table 4: Feature Information for MPLS Static Labels

Feature Name	Releases	Feature Information
MPLS Static Labels	Cisco IOS XE Everest 16.6.1	The MPLS Static Labels feature provides the means to configure the the binding between a label and an IPv4 prefix statically. The following commands were introduced or modified: **debug mpls static binding**, **mpls label range**, **mpls static binding ipv4**, **show mpls label range**, **show mpls static binding ipv4**

CHAPTER 5

Configuring Multicast VPN

Configuring Multicast VPN

The Multicast VPN (MVPN) feature provides the ability to support multicast over a Layer 3 VPN. As enterprises extend the reach of their multicast applications, service providers can accommodate them over their Multiprotocol Label Switching (MPLS) core network. IP multicast is used to stream video, voice, and data over an MPLS VPN network core.

Historically, point-to-point tunnels were the only way to connect through a service provider network. Although such tunneled networks tend to have scalability issues, they represented the only means of passing IP multicast traffic through a VPN.

Because Layer 3 VPNs support only unicast traffic connectivity, deploying MPLS in conjunction with a Layer 3 VPN allows service providers to offer both unicast and multicast connectivity to Layer 3 VPN customers.

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 at the end of this module.

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.

Prerequisites for Configuring Multicast VPN

Enable IP multicast and configure the PIM interfaces using the tasks described in the “Configuring Basic IP Multicast” module.

Restrictions for Configuring Multicast VPN

- The update source interface for the Border Gateway Protocol (BGP) peerings must be the same for all BGP peerings configured on the device in order for the default multicast distribution tree (MDT) to be configured properly. If you use a loopback address for BGP peering, PIM sparse mode must be enabled on the loopback address.

- MVPN does not support multiple BGP peering update sources.
- Multiple BGP update sources are not supported, and configuring them can break MVPN reverse path forwarding (RPF) checking. The source IP address of the MVPN tunnels is determined by the highest IP address used for the BGP peering update source. If this IP address is not the IP address used as the BGP peering address with the remote provider edge (PE) device, MVPN will not function properly.

Information About Configuring Multicast VPN

Multicast VPN Operation

MVPN IP allows a service provider to configure and support multicast traffic in an MPLS VPN environment. This feature supports routing and forwarding of multicast packets for each individual VRF instance, and it also provides a mechanism to transport VPN multicast packets across the service provider backbone.

A VPN is network connectivity across a shared infrastructure, such as an ISP. Its function is to provide the same policies and performance as a private network, at a reduced cost of ownership, thus creating many opportunities for cost savings through operations and infrastructure.

An MVPN allows an enterprise to transparently interconnect its private network across the network backbone of a service provider. The use of an MVPN to interconnect an enterprise network in this way does not change the way that enterprise network is administered, nor does it change general enterprise connectivity.

Benefits of Multicast VPN

- Provides a scalable method to dynamically send information to multiple locations.
- Provides high-speed information delivery.
- Provides connectivity through a shared infrastructure.

Multicast VPN Routing and Forwarding and Multicast Domains

MVPN introduces multicast routing information to the VPN routing and forwarding table. When a provider edge (PE) device receives multicast data or control packets from a customer edge (CE) router, forwarding is performed according to the information in the Multicast VPN routing and forwarding instance (MVRF). MVPN does not use label switching.

A set of MVRFs that can send multicast traffic to each other constitutes a multicast domain. For example, the multicast domain for a customer that wanted to send certain types of multicast traffic to all global employees would consist of all CE routers associated with that enterprise.

Multicast Distribution Trees

MVPN establishes a static default multicast distribution tree (MDT) for each multicast domain. The default MDT defines the path used by PE routers to send multicast data and control messages to every other PE router in the multicast domain.

If Source Specific Multicast (SSM) is used as the core multicast routing protocol, the multicast IP addresses used for the default and data MDT must be configured within the SSM range on all PE routers.

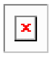
MVPN also supports the dynamic creation of MDTs for high-bandwidth transmission. Data MDTs are a feature unique to Cisco IOS software. Data MDTs are intended for high-bandwidth sources such as full-motion video inside the VPN to ensure optimal traffic forwarding in the MPLS VPN core. The threshold at which the data MDT is created can be configured on a per-router or a per-VRF basis. When the multicast transmission exceeds the defined threshold, the sending PE router creates the data MDT and sends a UDP message, which contains information about the data MDT, to all routers on the default MDT. The statistics to determine whether a multicast stream has exceeded the data MDT threshold are examined once every second. After a PE router sends the UDP message, it waits 3 more seconds before switching over; 13 seconds is the worst case switchover time, and 3 seconds is the best case.

Data MDTs are created only for (S, G) multicast route entries within the VRF multicast routing table. They are not created for (*, G) entries regardless of the value of the individual source data rate.

In the following example, a service provider has a multicast customer with offices in San Jose, New York, and Dallas. A one-way multicast presentation is occurring in San Jose. The service provider network supports all three sites associated with this customer, in addition to the Houston site of a different enterprise customer.

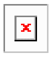
The default MDT for the enterprise customer consists of provider routers P1, P2, and P3 and their associated PE routers. PE4 is not part of the default MDT, because it is associated with a different customer. The figure shows that no data flows along the default MDT, because no one outside of San Jose has joined the multicast.

Figure 2: Default Multicast Distribution Tree Overview



An employee in New York joins the multicast session. The PE router associated with the New York site sends a join request that flows across the default MDT for the multicast domain of the customer. PE1, the PE router associated with the multicast session source, receives the request. The figure depicts that the PE router forwards the request to the CE router associated with the multicast source (CE1a).

Figure 3: Initializing the Data MDT



The CE router (CE1a) begins to send the multicast data to the associated PE router (PE1), which sends the multicast data along the default MDT. Immediately sending the multicast data, PE1 recognizes that the multicast data exceeds the bandwidth threshold for which a data MDT should be created. Therefore, PE1 creates a data MDT, sends a message to all routers using the default MDT, which contains information about the data MDT, and, three seconds later, begins sending the multicast data for that particular stream using the data MDT. Only PE2 has interested receivers for this source, so only PE2 will join the data MDT and receive traffic on it.

PE routers maintain a PIM relationship with other PE routers over the default MDT and a PIM relationship with directly attached PE routers.

Multicast Tunnel Interface

An MVRF, which is created per multicast domain, requires the device to create a tunnel interface from which all MVRF traffic is sourced. A multicast tunnel interface is an interface that the MVRF uses to access the multicast domain. It can be thought of as a conduit that connects an MVRF and the global MVRF. One tunnel interface is created per MVRF.

MDT Address Family in BGP for Multicast VPN

The **mdt** keyword has been added to the **address-family ipv4** command to configure an MDT address-family session. MDT address-family sessions are used to pass the source PE address and MDT group address to PIM using Border Gateway Protocol (BGP) MDT Subaddress Family Identifier (SAFI) updates.

BGP Advertisement Methods for Multicast VPN Support

In a single autonomous system, if the default MDT for an MVPN is using PIM sparse mode (PIM-SM) with a rendezvous point (RP), then PIM is able to establish adjacencies over the Multicast Tunnel Interface (MTI) because the source PE and receiver PE discover each other through the RP. In this scenario, the local PE (the source PE) sends register messages to the RP, which then builds a shortest-path tree (SPT) toward the source PE. The remote PE, which acts as a receiver for the MDT multicast group, then sends (*, G) joins toward the RP and joins the distribution tree for that group.

However, if the default MDT group is configured in a PIM Source Specific Multicast (PIM-SSM) environment rather than a PIM-SM environment, the receiver PE needs information about the source PE and the default MDT group. This information is used to send (S, G) joins toward the source PE to build a distribution tree from the source PE (without the need for an RP). The source PE address and default MDT group address are sent using BGP.

BGP Extended Community

When BGP extended communities are used, the PE loopback (source address) information is sent as a VPNv4 prefix using Route Distinguisher (RD) Type 2 (to distinguish it from unicast VPNv4 prefixes). The MDT group address is carried in a BGP extended community. Using a combination of the embedded source in the VPNv4 address and the group in the extended community, PE routers in the same MVRF instance can establish SSM trees to each other.

Note Prior to the introduction of MDT SAFI support, the BGP extended community attribute was used as an interim solution to advertise the IP address of the source PE and default MDT group before IETF standardization. A BGP extended community attribute in an MVPN environment, however, has certain limitations: it cannot be used in inter-AS scenarios (because the attribute is nontransitive), and it uses RD Type 2 (which is not a supported standard).

How to Configure Multicast VPN

Configuring the Data Multicast Group

A data MDT group can include a maximum of 256 multicast groups per VPN per VRF per PE device. Multicast groups used to create the data MDT group are dynamically chosen from a pool of configured IP addresses. Use the following procedure to configure data multicast group on the device.

Procedure

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

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vrf1`	Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 1:1`	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	**route-target both** *ASN:nn or IP-address:nn* **Example:** `Device(config-vrf)# route-target both 1:1`	Creates a route-target extended community for a VRF. The **both** keyword specifies to import both import and export routing information to the target VPN extended community.
Step 6	**address family ipv4 unicast** *value* **Example:** `Device(config-vrf)# address family ipv4 unicast`	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
Step 7	**mdt default** *group-address* **Example:** `Device(config-vrf-af)# mdt default 226.10.10.10`	Configures the multicast group address range for data MDT groups for a VRF. • A tunnel interface is created as a result of this command. • The default MDT group address configuration must be the same on all PEs in the same VRF.

	Command or Action	Purpose
Step 8	**mdt data** *group number* **Example:** `Device(config-vrf-af)# mdt data 232.0.1.0 0.0.0.31`	Specifies a range of addresses to be used in the data MDT pool.
Step 9	**mdt data threshold** *kbps* **Example:** `Device(config-vrf-af)# mdt data threshold 50`	Specifies the threshold in *kbps*. The range is from 1 to 4294967.
Step 10	**mdt log-reuse** **Example:** `Device(config-vrf-af)# mdt log-reuse`	(Optional) Enables the recording of data MDT reuse and generates a syslog message when a data MDT has been reused.

Step 11	**end**
	Example:
	`Device(config-vrf-af)# end`
		Returns to privileged EXEC mode.

Configuring a Default MDT Group for a VRF

Perform this task to configure a default MDT group for a VRF.

The default MDT group must be the same group configured on all devices that belong to the same VPN. The source IP address will be the address used to source the BGP sessions.

Procedure

	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 multicast-routing** **Example:** `Device(config)# ip multicast-routing`	Enables multicast routing.

	Command or Action	Purpose
Step 4	**ip multicast-routing vrf** *vrf-name* **Example:** `Device(config)# ip multicast-routing vrf vrf1`	Supports the MVPN VRF instance.
Step 5	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vrf1`	Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name.
Step 6	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 1:1`	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 7	**route-target both** *ASN:nn or IP-address:nn* **Example:** `Device(config-vrf)# route-target both 1:1`	Creates a route-target extended community for a VRF. The **both** keyword specifies to import both import and export routing information to the target VPN extended community.
Step 8	**address family ipv4 unicast** *value* **Example:** `Device(config-vrf)# address family ipv4 unicast`	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
Step 9	**mdt default** *group-address* **Example:** `Device(config-vrf-af)# mdt default 226.10.10.10`	Configures the multicast group address range for data MDT groups for a VRF. • A tunnel interface is created as a result of this command. • The default MDT group address configuration must be the same on all PEs in the same VRF.

	Command or Action	Purpose

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

Step 11	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 12	**ip pim vrf** *vrf-name***rp-address** *value* **Example:** `Device(config-vrf-af)# ip pim vrf vrf1 rp-address 1.1.1.1`	Enters the RP configuration mode.

Configuring the MDT Address Family in BGP for Multicast VPN

Perform this task to configure an MDT address family session on PE devices to establish MDT peering sessions for MVPN.

Before you begin

Before MVPN peering can be established through an MDT address family, MPLS and Cisco Express Forwarding (CEF) must be configured in the BGP network and multiprotocol BGP on PE devices that provide VPN services to CE devices.

Note The following policy configuration parameters are not supported:

- Route-originator attribute
- Network Layer Reachability Information (NLRI) prefix filtering (prefix lists, distribute lists)
- Extended community attributes (route target and site of origin)

Procedure

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

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65535`	Enters router configuration mode and creates a BGP routing process.
Step 4	**address-family ipv4 mdt** **Example:** `Device(config-router)# address-family ipv4 mdt`	Enters address family configuration mode to create an IP MDT address family session.
Step 5	**neighbor** *neighbor-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.1.1 activate`	Enables the MDT address family for this neighbor.
Step 6	**neighbor** *neighbor-address* **send-community** [**both** | **extended** | **standard**] **Example:** `Device(config-router-af)# neighbor 192.168.1.1 send-community extended`	Enables community and (or) extended community exchange with the specified neighbor.
Step 7	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode and returns to router configuration mode.
Step 8	**address-family vpnv4** **Example:** `Device(config-router)# address-family vpnv4`	Enters address family configuration mode to create a VPNv4 address family session.
Step 9	**neighbor** *neighbor-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.1.1 activate`	Enables the VPNv4 address family for this neighbor.
Step 10	**neighbor** *neighbor-address* **send-community** [**both** | **extended** | **standard**] **Example:**	Enables community and (or) extended community exchange with the specified neighbor.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor 192.168.1.1 send-community extended`	

Step 11	**end**
	Example:
	`Device(config-router-af)# end`
		Exits address family configuration mode and enters privileged EXEC mode.

Verifying Information for the MDT Default Group

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show ip pim** [**vrf** *vrf-name*] **mdt bgp**

Example:

```
Device# show ip pim mdt bgp

MDT-default group 232.2.1.4
rid:1.1.1.1 next_hop:1.1.1.1
```

Displays information about the BGP advertisement of the RD for the MDT default group.

Step 3 **show ip pim** [**vrf** *vrf-name*] **mdt send**

Example:

```
Device# show ip pim mdt send

MDT-data send list for VRF:vpn8
  (source, group)                     MDT-data group      ref_count
  (10.100.8.10, 225.1.8.1)            232.2.8.0           1
  (10.100.8.10, 225.1.8.2)            232.2.8.1           1
  (10.100.8.10, 225.1.8.3)            232.2.8.2           1
  (10.100.8.10, 225.1.8.4)            232.2.8.3           1
  (10.100.8.10, 225.1.8.5)            232.2.8.4           1
  (10.100.8.10, 225.1.8.6)            232.2.8.5           1
  (10.100.8.10, 225.1.8.7)            232.2.8.6           1
  (10.100.8.10, 225.1.8.8)            232.2.8.7           1
  (10.100.8.10, 225.1.8.9)            232.2.8.8           1
  (10.100.8.10, 225.1.8.10)           232.2.8.9           1
```

Displays detailed information about the MDT data group incluidng MDT advertisements that the specified device has made.

Step 4 **show ip pim vrf** *vrf-name* **mdt history interval** *minutes*

Example:

```
Device# show ip pim vrf vrf1 mdt history interval 20

   MDT-data send history for VRF - vrf1 for the past 20 minutes
MDT-data group        Number of reuse
     10.9.9.8           3
     10.9.9.9           2
```

Displays the data MDTs that have been reused during the past configured interval.

Configuration Examples for Multicast VPN

Example: Configuring MVPN and SSM

In the following example, PIM-SSM is configured in the backbone. Therefore, the default and data MDT groups are configured within the SSM range of IP addresses. Inside the VPN, PIM-SM is configured and only Auto-RP announcements are accepted.

```
ip vrf vrf1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
 mdt default 232.0.0.1
 mdt data 232.0.1.0 0.0.0.255 threshold 500 list 101
!
ip pim ssm default
ip pim vrf vrf1 accept-rp auto-rp
```

Example: Enabling a VPN for Multicast Routing

In the following example, multicast routing is enabled with a VPN routing instance named vrf1:

```
ip multicast-routing vrf1
```

Example: Configuring the Multicast Group Address Range for Data MDT Groups

In the following example, the VPN routing instance is assigned a VRF named blue. The MDT default group for a VPN VRF is 239.1.1.1, and the multicast group address range for MDT groups is 239.1.2.0 with wildcard bits of 0.0.0.3:

```
ip vrf blue
 rd 55:1111
 route-target both 55:1111
 mdt default 239.1.1.1
 mdt data 239.1.2.0 0.0.0.3
 end
```

Example: Limiting the Number of Multicast Routes

In the following example, the number of multicast routes that can be added to a multicast routing table is set to 200,000 and the threshold value of the number of mroutes that will cause a warning message to occur is set to 20,000:

```
!
ip multicast-routing
ip multicast-routing vrf cisco
ip multicast cache-headers
ip multicast route-limit 200000 20000
ip multicast vrf cisco route-limit 200000 20000
no mpls traffic-eng auto-bw timers frequency 0
!
```

Additional References for Configuring Multicast VPN

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	

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 Configuring Multicast VPN

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

Table 5: Feature Information for Multicast VPN

Release	Modification
	This feature was introduced.

INDEX