## CONTENTS

### CHAPTER 1

**Configuring Multiprotocol Label Switching (MPLS)**  
- Multiprotoocol Label Switching 1  
- Finding Feature Information 1  
- Restrictions for Multiprotocol Label Switching 1  
- Information about Multiprotocol Label Switching 2  
  - Functional Description of Multiprotocol Label Switching 2  
  - Label Switching Functions 2  
  - Distribution of Label Bindings 2  
  - MPLS Layer 3 VPN 3  
  - Classifying and Marking MPLS QoS EXP 3  
- How to Configure Multiprotocol Label Switching 4  
  - Configuring a Switch for MPLS Switching 4  
  - Configuring a Switch for MPLS Forwarding 5  
- Verifying Multiprotocol Label Switching Configuration 6  
  - Verifying Configuration of MPLS Switching 6  
  - Verifying Configuration of MPLS Forwarding 6  
- Additional References for Multiprotocol Label Switching 8  
- Feature Information for Multiprotocol Label Switching 9

### CHAPTER 2

**Configuring eBGP Multipath** 11  
- BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN 11  
  - Prerequisites for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN 11  
  - Restrictions for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN 12  
- Information About BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN 12  
- Multipath Load Sharing Between eBGP and iBGP 12  
- eBGP and iBGP Multipath Load Sharing in a BGP MPLS Network 13
Contents

Prerequisites for EoMPLS 30
Restrictions for EoMPLS 30
Configuring Port-Mode EoMPLS 31
  Xconnect Mode 31
  Protocol CLI Method 32
Configuration Examples for EoMPLS 35
Configuring Pseudowire Redundancy 39
  Information About Pseudowire Redundancy 39
Prerequisites for Pseudowire Redundancy 39
Restrictions for Pseudowire Redundancy 39
Configuring Pseudowire Redundancy 39
  Xconnect Mode 40
  Protocol CLI Method 41
Configuration Examples for Pseudowire Redundancy 44

CHAPTER 5
Configuring IPv6 Provider Edge over MPLS (6PE) 47
  Finding Feature Information 47
  Configuring 6PE 47
    Information About 6PE 47
    Scale Numbers 48
  Prerequisites for 6PE 48
  Restrictions for 6PE 48
  Configuring 6PE 48
  Configuration Examples for 6PE 51

CHAPTER 6
Configuring IPv6 VPN Provider Edge over MPLS (6VPE) 55
  Finding Feature Information 55
  Configuring 6VPE 55
    Information About 6VPE 55
    Scale Numbers 56
  Restrictions for 6VPE 56
  Information About 6VPE 56
  Configuration Examples for 6VPE 57
CHAPTER 11

Configuring MPLS VPN Route Target Rewrite 121
Finding Feature Information 121
Prerequisites for MPLS VPN Route Target Rewrite 121
Restrictions for MPLS VPN Route Target Rewrite 121
Information About MPLS VPN Route Target Rewrite 122
Route Target Replacement Policy 122
Route Maps and Route Target Replacement 122
How to Configure MPLS VPN Route Target Rewrite 123
Configuring a Route Target Replacement Policy 123
Applying the Route Target Replacement Policy 126
Associating Route Maps with Specific BGP Neighbors 126
Verifying the Route Target Replacement Policy 128
Configuration Examples for MPLS VPN Route Target Rewrite 129
Examples: Configuring Route Target Replacement Policies 129
Examples: Applying Route Target Replacement Policies 130
Examples: Associating Route Maps with Specific BGP Neighbor 130

CHAPTER 12

Configuring Multicast Virtual Private Network 131
Configuring Multicast VPN 131
Finding Feature Information 131
Prerequisites for Configuring Multicast VPN 131
Restrictions for Configuring Multicast VPN 131
Information About Configuring Multicast VPN 132
Multicast VPN Operation 132
Benefits of Multicast VPN 132
CHAPTER 1

Configuring Multiprotocol Label Switching (MPLS)

- Multiprotocol Label Switching, on page 1
- Finding Feature Information, on page 1
- Restrictions for Multiprotocol Label Switching, on page 2
- Information about Multiprotocol Label Switching, on page 2
- How to Configure Multiprotocol Label Switching, on page 4
- Verifying Multiprotocol Label Switching Configuration, on page 6
- Additional References for Multiprotocol Label Switching, on page 8
- Feature Information for Multiprotocol Label Switching, on page 9

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

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

SUMMARY STEPS

1. enable
2. configure terminal
3. ip cef distributed
4. mpls label range minimum-value maximum-value
5. mpls label protocol ldp

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ip cef distributed</td>
<td>Enables Cisco Express Forwarding on the switch.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# ip cef distributed</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> mpls label range minimum-value maximum-value</td>
<td>Configure the range of local labels available for use with MPLS applications on packet interfaces.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# mpls label range 16 4096</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> mpls label protocol ldp</td>
<td>Specifies the label distribution protocol for the platform.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# mpls label protocol ldp</td>
<td></td>
</tr>
</tbody>
</table>
Configuring a Switch for MPLS Forwarding

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

SUMMARY STEPS

1. enable
2. configure terminal
3. interface type slot/subslot/port
4. mpls ip
5. mpls label protocol ldp
6. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td>Device&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

| Step 2 | configure terminal                            | Enters global configuration mode.                                       |
| Example: |                                               |                                                                         |
|        | Device# configure terminal                    |                                                                         |

| Step 3 | interface type slot/subslot/port             | Specifies the Gigabit Ethernet interface and enters interface configuration mode. For Switch Virtual Interface (SVI), the example is |
| Example: |                                               | Device(config)# interface gigabitethernet 1/0/0                        |
|        |                                               | Device(config)# interface vlan 1000                                   |

| Step 4 | mpls ip                                       | Enables MPLS forwarding of IPv4 packets along routed physical interfaces (Gigabit Ethernet), Switch Virtual Interface (SVI), or port channels. |
| Example: |                                               |                                                                         |
|        | Device(config-if)# mpls ip                   |                                                                         |

| Step 5 | 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. |
| Example: |                                               |                                                                         |
|        | Device(config-if)# mpls label protocol ldp   |                                                                         |

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

**SUMMARY STEPS**

1. `show ip cef summary`

**DETAILED STEPS**

```
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.

**SUMMARY STEPS**

1. `show mpls interfaces detail`
2. `show running-config interface`
3. `show mpls forwarding`
DETAILED STEPS

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

<table>
<thead>
<tr>
<th>Local</th>
<th>Outgoing Prefix</th>
<th>Bytes Label</th>
<th>Switched Label</th>
<th>Outgoing Label or Tunnel Id</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>12ckt(3)</td>
<td>0</td>
<td>12310411816789</td>
<td>Gi3/0/22</td>
<td>point2point</td>
</tr>
<tr>
<td>501</td>
<td>12ckt(1)</td>
<td>0</td>
<td>none</td>
<td>point2point</td>
<td></td>
</tr>
<tr>
<td>502</td>
<td>12ckt(2)</td>
<td>0</td>
<td>none</td>
<td>point2point</td>
<td></td>
</tr>
<tr>
<td>503</td>
<td>15.15.15.15/32</td>
<td>0</td>
<td>Po5</td>
<td>192.1.1.2</td>
<td></td>
</tr>
<tr>
<td>504</td>
<td>7.7.7.7/32</td>
<td>538728528</td>
<td>Po5</td>
<td>192.1.1.2</td>
<td></td>
</tr>
<tr>
<td>505</td>
<td>6.6.6.10/32</td>
<td>0</td>
<td>Po5</td>
<td>192.1.1.2</td>
<td></td>
</tr>
<tr>
<td>506</td>
<td>6.6.6.6/32</td>
<td>0</td>
<td>Po5</td>
<td>192.1.1.2</td>
<td></td>
</tr>
<tr>
<td>507</td>
<td>1.1.1.1/32</td>
<td>0</td>
<td>Po5</td>
<td>192.1.1.2</td>
<td></td>
</tr>
<tr>
<td>556</td>
<td>19.10.1.0/24</td>
<td>0</td>
<td>Po5</td>
<td>192.1.1.2</td>
<td></td>
</tr>
<tr>
<td>668</td>
<td>21.1.1.0/24</td>
<td>0</td>
<td>Po5</td>
<td>192.1.1.2</td>
<td></td>
</tr>
<tr>
<td>574</td>
<td>213.1.1.0/24(V)</td>
<td>0</td>
<td>aggregate/vpn113</td>
<td></td>
<td></td>
</tr>
<tr>
<td>575</td>
<td>213.1.2.0/24(V)</td>
<td>0</td>
<td>aggregate/vpn114</td>
<td></td>
<td></td>
</tr>
<tr>
<td>576</td>
<td>213.1.3.0/24(V)</td>
<td>0</td>
<td>aggregate/vpn115</td>
<td></td>
<td></td>
</tr>
<tr>
<td>577</td>
<td>213.1.111/64</td>
<td>0</td>
<td>aggregate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>594</td>
<td>103.1.1.0/24</td>
<td>0</td>
<td>Po5</td>
<td>192.1.1.2</td>
<td></td>
</tr>
<tr>
<td>595</td>
<td>31.1.1.0/24</td>
<td>0</td>
<td>Po5</td>
<td>192.1.1.2</td>
<td></td>
</tr>
<tr>
<td>596</td>
<td>15.15.1.0/24</td>
<td>0</td>
<td>Po5</td>
<td>192.1.1.2</td>
<td></td>
</tr>
<tr>
<td>597</td>
<td>14.14.1.0/24</td>
<td>0</td>
<td>Po5</td>
<td>192.1.1.2</td>
<td></td>
</tr>
<tr>
<td>633</td>
<td>2.2.2.0/24</td>
<td>0</td>
<td>Po5</td>
<td>192.1.1.2</td>
<td></td>
</tr>
<tr>
<td>634</td>
<td>90.90.90.90/32</td>
<td>873684</td>
<td>Po5</td>
<td>192.1.1.2</td>
<td></td>
</tr>
<tr>
<td>635</td>
<td>154.1.1.0/24</td>
<td>0</td>
<td>Po5</td>
<td>192.1.1.2</td>
<td></td>
</tr>
<tr>
<td>636</td>
<td>153.1.1.0/24</td>
<td>0</td>
<td>Po5</td>
<td>192.1.1.2</td>
<td></td>
</tr>
</tbody>
</table>

Switch# end

Additional References for Multiprotocol Label Switching

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Command List, All Releases</td>
</tr>
</tbody>
</table>

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 9300 Series Switches)
Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.5.1a</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>
BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

The BGP Multipath Load Sharing for eBGP and iBGP feature allows you to configure multipath load balancing with both external BGP (eBGP) and internal BGP (iBGP) paths in Border Gateway Protocol (BGP) networks that are configured to use Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). This feature provides improved load balancing deployment and service offering capabilities and is useful for multi-homed autonomous systems and Provider Edge (PE) routers that import both eBGP and iBGP paths from multihomed and stub networks.

Prerequisites for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

Cisco Express Forwarding (CEF) or distributed CEF (dCEF) must be enabled on all participating devices.
Restrictions for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

Address Family Support
This feature is configured on a per VPN routing and forwarding instance (VRF) basis. This feature can be configured under both IPv4 and IPv6 VRF address families.

Memory Consumption Restriction
Each BGP multipath routing table entry will use additional memory. We recommend that you do not use this feature on a device with a low amount of available memory and especially if the device carries full Internet routing tables.

Number of Paths Limitation
The number of paths supported are limited to 2 BGP multipaths. This could either be 2 iBGP multipaths or 1 iBGP multipath and 1 eBGP multipath.

Unsupported Commands
`ip unnumbered` command is not supported in MPLS configuration.

Information About BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

Multipath Load Sharing Between eBGP and iBGP
A BGP routing process will install a single path as the best path in the routing information base (RIB) by default. The `maximum-paths` command allows you to configure BGP to install multiple paths in the RIB for multipath load sharing. BGP uses the best path algorithm to select a single multipath as the best path and advertise the best path to BGP peers.

Note
The number of paths of multipaths that can be configured is documented on the `maximum-paths` command reference page.

Load balancing over the multipaths is performed by CEF. CEF load balancing is configured on a per-packet round robin or on a per session (source and destination pair) basis. For information about CEF, refer to Cisco IOS IP Switching Configuration Guide documentation: http://ciscosystems.com/en/US/docs/ios/ipswitch/configuration/guide/12_2sx/sw_12_2sx_book.html. The BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature is enabled under the IPv4 VRF address family and IPv6 VRF address family configuration modes. When enabled, this feature can perform load balancing on eBGP and/or iBGP paths that are imported into the VRF. The number of multipaths is configured on a per VRF basis. Separate VRF multipath configurations are isolated by unique route distinguisher.
The BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature operates within the parameters of configured outbound routing policy.

**eBGP and iBGP Multipath Load Sharing in a BGP MPLS Network**

The following figure shows a service provider BGP MPLS network that connects two remote networks to PE router 1 and PE router 2. PE router 1 and PE router 2 are both configured for VPNv4 unicast iBGP peering. Network 2 is a multihomed network that is connected to PE router 1 and PE router 2. Network 2 also has extranet VPN services configured with Network 1. Both Network 1 and Network 2 are configured for eBGP peering with the PE routers.

*Figure 1: Service Provider BGP MPLS Network*

PE router 1 can be configured with the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature so that both iBGP and eBGP paths can be selected as multipaths and imported into the VRF. The multipaths will be used by CEF to perform load balancing. IP traffic that is sent from Network 1 to Network 2, PE router 1 will Load Sharing with eBGP paths as IP traffic & iBGP path will be sent as MPLS traffic.

**Note**
- eBGP session between local CE & local PE is not supported.
- • eBGP session from a local PE to a remote CE is supported.

**Benefits of Multipath Load Sharing for Both eBGP and iBGP**

The BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature allows multihomed autonomous systems and PE routers to be configured to distribute traffic across both eBGP and iBGP paths.

**How to Configure BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN**

This section contains the following procedures:
Configuring Multipath Load Sharing for Both eBGP and iBGP

**SUMMARY STEPS**

1. `enable`
2. `configure {terminal | memory | network}`
3. `router bgp as-number`
4. `address-family ipv4 vrf vrf-name`
5. `address-family ipv6 vrf vrf-name`
6. `maximum-paths eibgp number [import number]`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td><code>Device&gt;</code> <code>enable</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> `configure {terminal</td>
<td>memory</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Device#$ configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>router bgp as-number</code></td>
<td>Enters router configuration mode to create or configure a BGP routing process.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Device(config)# router bgp 40000</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>address-family ipv4 vrf vrf-name</code></td>
<td>Places the router in address family configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Separate VRF multipath configurations are isolated by unique route distinguisher.</td>
</tr>
<tr>
<td><code>Device(config-router)# address-family ipv4 vrf RED</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>address-family ipv6 vrf vrf-name</code></td>
<td>Places the router in address family configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Separate VRF multipath configurations are isolated by unique route distinguisher.</td>
</tr>
<tr>
<td><code>Device(config-router)# address-family ipv6 vrf RED</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> <code>maximum-paths eibgp number [import number]</code></td>
<td>Configures the number of parallel iBGP and eBGP routes that can be installed into a routing table.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Device(config-router-af)# maximum-paths eibgp 2</code></td>
<td>Note: The maximum-paths eibgp command can be configured only under the IPv4 VRF address family configuration mode and cannot be configured in any other address family configuration mode.</td>
</tr>
</tbody>
</table>
Verifying Multipath Load Sharing for Both eBGP and iBGP

SUMMARY STEPS

1. enable
2. show ip bgp neighbors
3. show ip bgp vpnv4 vrf vrf name
4. show ip route vrf vrf-name

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>show ip bgp neighbors</td>
<td>Displays information about the TCP and BGP connections to neighbors.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device# show ip bgp neighbors</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>show ip bgp vpnv4 vrf vrf name</td>
<td>Displays VPN address information from the BGP table. This command is used to verify that the VRF has been received by BGP.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device# show ip bgp vpnv4 vrf RED</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>show ip route vrf vrf-name</td>
<td>Displays the IP routing table associated with a VRF instance. The show ip route vrf command is used to verify that the VRF is in the routing table.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device# show ip route vrf RED</td>
<td></td>
</tr>
</tbody>
</table>

Example

What to do next

Configuration Examples for the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN Feature

The following examples show how to configure and verify this feature:
eBGP and iBGP Multipath Load Sharing Configuration Example

This following configuration example configures a router in IPv4 address-family mode to select two BGP routes (eBGP or iBGP) as multipaths:

```
Device
  router bgp 40000
  address-family ipv4 vrf RED
  maximum-paths eibgp 2
Device
end
```

This following configuration example configures a router in IPv6 address-family mode to select two BGP routes (eBGP or iBGP) as multipaths:

```
Device
  router bgp 40000
  address-family ipv6 vrf RED
  maximum-paths eibgp 2
Device
end
```

Additional References

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples</td>
<td>Cisco IOS IP Command Reference, Volume 2 of 4: Routing Protocols, Release 12.3T</td>
</tr>
<tr>
<td>BGP configuration tasks</td>
<td>Cisco IOS IP Configuration Guide, Release 12.3</td>
</tr>
<tr>
<td>Comprehensive BGP link bandwidth configuration examples and tasks</td>
<td>BGP Link Bandwidth</td>
</tr>
<tr>
<td>CEF configuration tasks</td>
<td>Cisco IOS Switching Services Configuration Guide</td>
</tr>
</tbody>
</table>

Table 3: Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature</td>
<td>--</td>
</tr>
</tbody>
</table>

Table 4: RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 1771</td>
<td>A Border Gateway Protocol 4 (BGP4)</td>
</tr>
</tbody>
</table>
Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

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

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

**Table 6: Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN**

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN</td>
<td>Cisco IOS 16.6.1</td>
<td>The BGP Multipath Load Sharing for eBGP and iBGP feature allows you to configure multipath load balancing with both external BGP (eBGP) and internal BGP (iBGP) paths in Border Gateway Protocol (BGP) networks that are configured to use Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). This feature provides improved load balancing deployment and service offering capabilities and is useful for multi-homed autonomous systems and Provider Edge (PE) routers that import both eBGP and iBGP paths from multihomed and stub networks.</td>
</tr>
</tbody>
</table>
Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN
CHAPTER 3

Configuring EIGRP MPLS VPN PE-CE Site of Origin

- EIGRP MPLS VPN PE-CE Site of Origin, on page 19
- Information About EIGRP MPLS VPN PE-CE Site of Origin, on page 20
- How to Configure EIGRP MPLS VPN PE-CE Site of Origin Support, on page 22
- Configuration Examples for EIGRP MPLS VPN PE-CE SoO, on page 25
- Additional References, on page 26
- Feature Information for EIGRP MPLS VPN PE-CE Site of Origin, on page 27

EIGRP MPLS VPN PE-CE Site of Origin

The EIGRP MPLS VPN PE-CE Site of Origin feature introduces the capability to filter Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) traffic on a per-site basis for Enhanced Interior Gateway Routing Protocol (EIGRP) networks. Site of Origin (SoO) filtering is configured at the interface level and is used to manage MPLS VPN traffic and to prevent transient routing loops from occurring in complex and mixed network topologies. This feature is designed to support the MPLS VPN Support for EIGRP Between Provider Edge (PE) and Customer Edge (CE) feature. Support for backdoor links is provided by this feature when installed on PE routers that support EIGRP MPLS VPNs.

Prerequisites for EIGRP MPLS VPN PE-CE Site of Origin

This document assumes that Border Gateway Protocol (BGP) is configured in the network core (or the service provider backbone). The following tasks will also need to be completed before you can configure this feature:

- This feature was introduced to support the MPLS VPN Support for EIGRP Between Provider Edge and Customer Edge feature and should be configured after the EIGRP MPLS VPN is created.

- All PE routers that are configured to support the EIGRP MPLS VPN must run Cisco IOS XE Release 2.1 or a later release, which provides support for the SoO extended community

Restrictions for EIGRP MPLS VPN PE-CE Site of Origin

- If a VPN site is partitioned and the SoO extended community attribute is configured on a backdoor router interface, the backdoor link cannot be used as an alternate path to reach prefixes originated in other partitions of the same site
A unique SoO value must be configured for each individual VPN site. The same value must be configured on all provider edge and customer edge interfaces (if SoO is configured on the CE routers) that support the same VPN site.

Information About EIGRP MPLS VPN PE-CE Site of Origin

EIGRP MPLS VPN PE-CE Site of Origin Support Overview

The EIGRP MPLS VPN PE-CE Site of Origin feature introduces SoO support for EIGRP-to-BGP and BGP-to-EIGRP redistribution. The SoO extended community is a BGP extended community attribute that is used to identify routes that have originated from a site so that the readvertisement of that prefix back to the source site can be prevented. The SoO extended community uniquely identifies the site from which a PE router has learned a route. SoO support provides the capability to filter MPLS VPN traffic on a per-EIGRP-site basis. SoO filtering is configured at the interface level and is used to manage MPLS VPN traffic and to prevent routing loops from occurring in complex and mixed network topologies, such as EIGRP VPN sites that contain both VPN and backdoor links.

The configuration of the SoO extended community allows MPLS VPN traffic to be filtered on a per-site basis. The SoO extended community is configured in an inbound BGP route map on the PE router and is applied to the interface. The SoO extended community can be applied to all exit points at the customer site for more specific filtering but must be configured on all interfaces of PE routers that provide VPN services to CE routers.

Site of Origin Support for Backdoor Links

The EIGRP MPLS VPN PE-CE Site of Origin (SoO) feature introduces support for backdoor links. A backdoor link or a route is a connection that is configured outside of the VPN between a remote and main site; for example, a WAN leased line that connects a remote site to the corporate network. Backdoor links are typically used as back up routes between EIGRP sites if the VPN link is down or not available. A metric is set on the backdoor link so that the route though the backdoor router is not selected unless there is a VPN link failure.

The SoO extended community is defined on the interface of the backdoor router. It identifies the local site ID, which should match the value that is used on the PE routers that support the same site. When the backdoor router receives an EIGRP update (or reply) from a neighbor across the backdoor link, the router checks the update for an SoO value. If the SoO value in the EIGRP update matches the SoO value on the local backdoor interface, the route is rejected and not added to the EIGRP topology table. This scenario typically occurs when the route with the local SoO valued in the received EIGRP update was learned by the other VPN site and then advertised through the backdoor link by the backdoor router in the other VPN site. SoO filtering on the backdoor link prevents transient routing loops from occurring by filtering out EIGRP updates that contain routes that carry the local site ID.

If this feature is enabled on the PE routers and the backdoor routers in the customer sites, and SoO values are defined on both the PE and backdoor routers, both the PE and backdoor routers will support convergence between the VPN sites. The other routers in the customer sites need only propagate the SoO values carried by the routes, as the routes are forwarded to neighbors. These routers do not otherwise affect or support convergence beyond normal Diffusing Update Algorithm (DUAL) computations.
Router Interoperation with the Site of Origin Extended Community

The configuration of an SoO extended community allows routers that support EIGRP MPLS VPN PE-CE Site of Origin feature to identify the site from which each route originated. When this feature is enabled, the EIGRP routing process on the PE or CE router checks each received route for the SoO extended community and filters based on the following conditions:

- A received route from BGP or a CE router contains an SoO value that matches the SoO value on the receiving interface: If a route is received with an associated SoO value that matches the SoO value that is configured on the receiving interface, the route is filtered because it was learned from another PE router or from a backdoor link. This behavior is designed to prevent routing loops.

- A received route from a CE router is configured with an SoO value that does not match: If a route is received with an associated SoO value that does not match the SoO value that is configured on the receiving interface, the route is added to the EIGRP topology table so that it can be redistributed into BGP. If the route is already installed to the EIGRP topology table but is associated with a different SoO value, the SoO value from the topology table will be used when the route is redistributed into BGP.

- A received route from a CE router does not contain an SoO value: If a route is received without a SoO value, the route is accepted into the EIGRP topology table, and the SoO value from the interface that is used to reach the next hop CE router is appended to the route before it is redistributed into BGP.

When BGP and EIGRP peers that support the SoO extended community receive these routes, they will also receive the associated SoO values and pass them to other BGP and EIGRP peers that support the SoO extended community. This filtering is designed to prevent transient routes from being relearned from the originating site, which prevents transient routing loops from occurring.

Redistribution of BGP VPN Routes That Carry the Site of Origin into EIGRP

When an EIGRP routing process on a PE router redistributes BGP VPN routes into an EIGRP topology table, EIGRP extracts the SoO value (if one is present) from the appended BGP extended community attributes and appends the SoO value to the route before adding it to the EIGRP topology table. EIGRP tests the SoO value for each route before sending updates to CE routers. Routes that are associated with SoO values that match the SoO value configured on the interface are filtered out before they are passed to the CE routers. When an EIGRP routing process receives routes that are associated with different SoO values, the SoO value is passed to the CE router and carried through the CE site.

Benefits of the EIGRP MPLS VPN PE-CE Site of Origin Support Feature

The configuration of the EIGRP MPLS VPN PE-CE Site of Origin Support feature introduces per-site VPN filtering, which improves support for complex topologies, such as MPLS VPNs with backdoor links, CE routers that are dual-homed to different PE routers, and PE routers that support CE routers from different sites within the same virtual routing and forwarding (VRF) instance.
How to Configure EIGRP MPLS VPN PE-CE Site of Origin Support

Configuring the Site of Origin Extended Community

The configuration of the SoO extended community allows MPLS VPN traffic to be filtered on a per-site basis. The SoO extended community is configured in an inbound BGP route map on the PE router and is applied to the interface. The SoO extended community can be applied to all exit points at the customer site for more specific filtering but must be configured on all interfaces of PE routers that provide VPN services to CE routers.

Before you begin

• Confirm that the Border Gateway Protocol (BGP) is configured in the network core (or the service provider backbone).
• Configure an EIGRP MPLS VPN before configuring this feature.
• All PE routers that are configured to support the EIGRP MPLS VPN must support the SoO extended community.
• A unique SoO value must be configured for each VPN site. The same value must be used on the interface of the PE router that connects to the CE router for each VPN site.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `route-map map-name {permit | deny} [sequence-number]`
4. `set extcommunity soo extended-community-value`
5. `exit`
6. `interface type number`
7. `no switchport`
8. `vrf forwarding vrf-name`
9. `ip vrf sitemap route-map-name`
10. `ip address ip-address subnet-mask`
11. `end`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><code>enable</code></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device&gt; enable</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td>Enters route-map configuration mode and creates a route map.</td>
</tr>
<tr>
<td><strong>Step 3</strong> route-map map-name {(\text{permit</td>
<td>deny}) [sequence-number]</td>
</tr>
<tr>
<td>Example: Device(config)# route-map Site-of-Origin permit 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> set extcommunity soo {extended-community-value</td>
<td>Sets BGP extended community attributes.</td>
</tr>
<tr>
<td>Example: Device(config-route-map)# set extcommunity soo 100:1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> exit</td>
<td>Exits route-map configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device(config-route-map)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> interface type number</td>
<td>Enters interface configuration mode to configure the specified interface.</td>
</tr>
<tr>
<td>Example: Device(config)# interface GigabitEthernet 1/0/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> no switchport</td>
<td>causes the interface to cease operating as a Layer 2 port and become a Cisco-routed (Layer 3) port:</td>
</tr>
<tr>
<td>Example: Device(config-if)# no switchport</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> vrf forwarding vrf-name</td>
<td>Associates the VRF with an interface or subinterface.</td>
</tr>
<tr>
<td>Example: Device(config-if)# ip vrf forwarding VRF1</td>
<td>• The VRF name configured in this step should match the VRF name created for the EIGRP MPLS VPN with the MPLS VPN Support for EIGRP Between Provider Edge and Customer Edge feature.</td>
</tr>
<tr>
<td><strong>Step 9</strong> ip vrf sitemap route-map-name</td>
<td>Associates the VRF with an interface or subinterface.</td>
</tr>
<tr>
<td>Example: Device(config-if)# ip vrf sitemap Site-of-Origin</td>
<td>• The route map name configured in this step should match the route map name created to apply the SoO extended community in Step 3.</td>
</tr>
</tbody>
</table>
Configuring the IP address for the interface.

**Command or Action**
```
ip address ip-address subnet-mask
```
**Example:**
```
Device(config-if)# ip address 10.0.0.1 255.255.255.255
```
**Purpose**
- Configures the IP address for the interface.
- The IP address needs to be reconfigured after enabling VRF forwarding.

**Step 10**

**Step 11**
```
end
```
**Example:**
```
Device(config-if)# end
```
**Purpose**
- Exits interface configuration mode and enters privileged EXEC mode.

### Verifying the Configuration of the SoO Extended Community

**SUMMARY STEPS**

1. `enable`
2. `show ip bgp vpnv4 { all | rd route-distinguisher | vrf vrf-name } [ip-prefix/length]`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; <code>enable</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> `show ip bgp vpnv4 { all</td>
<td>rd route-distinguisher</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# <code>ip bgp vpnv4 vrf SOO-1 20.2.1.1/32</code></td>
<td></td>
</tr>
</tbody>
</table>

**What to do next**

- For mixed EIGRP MPLS VPN network topologies that contain backdoor routes, the next task is to configure the “prebest path” cost community for backdoor routes.

---

**Summary**

- Configures the IP address for the interface.
- Exits interface configuration mode and enters privileged EXEC mode.
- Displays VPN address information from the BGP table.

**Detailed Steps**

1. **Enable**
   - `enable`
   - **Example:** Device> `enable`
2. **Show IP BGP VPNv4**
   - `show ip bgp vpnv4 { all | rd route-distinguisher | vrf vrf-name } [ip-prefix/length]`
   - **Example:** Device# `ip bgp vpnv4 vrf SOO-1 20.2.1.1/32`
Configuration Examples for EIGRP MPLS VPN PE-CE SoO

Example Configuring the Site of Origin Extended Community

The following example, beginning in global configuration mode, configures SoO extended community on an interface:

```plaintext
route-map Site-of-Origin permit 10
set extcommunity soo 100:1
exit
GigabitEthernet1/0/1
ip vrf forwarding RED
ip vrf sitemap Site-of-Origin
ip address 10.0.0.1 255.255.255.255
end
```

Example Verifying the Site of Origin Extended Community

The following example shows VPN address information from the BGP table and verifies the configuration of the SoO extended community:

```
switch# show ip bgp vpnv4 all 10.0.0.1
BGP routing table entry for 100:1:10.0.0.1/32, version 6
Paths: (1 available, best #1, no table)
Advertised to update-groups:
1
100 300
192.168.0.2 from 192.168.0.2 (172.16.13.13)
Origin incomplete, localpref 100, valid, external, best
Extended Community: SOO:100:1
```

```
CE1#show ip eigrp topo 20.2.1.1/32
EIGRP-IPv4 Topology Entry for AS(30)/ID(30.0.0.1) for 20.2.1.1/32
State is Passive, Query origin flag is 1, 2 Successor(s), FD is 131072
Descriptor Blocks:
31.1.1.2 (GigabitEthernet1/0/13), from 31.1.1.2, Send flag is 0x0
  Composite metric is (131072/130816), route is External
  Vector metric:
    Minimum bandwidth is 1000000 Kbit
    Total delay is 5020 microseconds
    Reliability is 255/255
    Load is 1/255
    Minimum MTU is 1500
    Hop count is 2
    Originating router is 30.0.0.2
    Extended Community: SoO:100:1
  External data:
    AS number of route is 0
    External protocol is Connected, external metric is 0
    Administrator tag is 0 (0x00000000)
```
Show command Provider Edge Device

PE2#show ip eigrp vrf SOO-1 topology 31.1.1.0/24
EIGRP-IPv4 VR(L3VPN) Topology Entry for AS(30)/ID(2.2.2.22)
    Topology(base) TID(0) VRF(SOO-1)
EIGRP-IPv4(30): Topology base(0) entry for 31.1.1.0/24
    State is Passive, Query origin flag is 1, 1 Successor(s), FD is 1310720
    Descriptor Blocks:
    1.1.1.1, from VPNv4 Sourced, Send flag is 0x0
        Composite metric is (1310720/0), route is Internal (VPNv4 Sourced)
    Vector metric:
        Minimum bandwidth is 1000000 Kbit
        Total delay is 10000000 picoseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 0
        Originating router is 1.1.1.11
    Extended Community: SoO:100:1

Additional References

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Command List, All Releases</td>
</tr>
<tr>
<td>BGP Cost Community feature and the “pre-bestpath” point of insertion</td>
<td></td>
</tr>
<tr>
<td>CEF commands</td>
<td>Cisco IOS IP Switching Command Reference</td>
</tr>
<tr>
<td>CEF configuration tasks</td>
<td>Cisco Express Forwarding Overview module of the Cisco IOS IP Switching Configuration Guide</td>
</tr>
<tr>
<td>EIGRP commands</td>
<td>Cisco IOS IP Routing: EIGRP Command Reference</td>
</tr>
<tr>
<td>EIGRP configuration tasks</td>
<td>Configuring EIGRP</td>
</tr>
<tr>
<td>MPLS VPNs</td>
<td>MPLS Layer 3 VPNs module of the Cisco IOS Multiprotocol Label Switching Configuration Guide</td>
</tr>
</tbody>
</table>

Table 7: Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>--</td>
</tr>
</tbody>
</table>
Table 8: MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://tools.cisco.com/ITDIT/MIBS/servlet/index">http://tools.cisco.com/ITDIT/MIBS/servlet/index</a></td>
</tr>
</tbody>
</table>

Table 9: RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>--</td>
</tr>
</tbody>
</table>

Table 10: Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/c/en/us/support/index.html">http://www.cisco.com/c/en/us/support/index.html</a></td>
</tr>
</tbody>
</table>

Feature Information for EIGRP MPLS VPN PE-CE Site of Origin

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](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.
The EIGRP MPLS VPN PE-CE Site of Origin feature introduces the capability to filter Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) traffic on a per-site basis for Enhanced Interior Gateway Routing Protocol (EIGRP) networks. Site of Origin (SoO) filtering is configured at the interface level and is used to manage MPLS VPN traffic and to prevent transient routing loops from occurring in complex and mixed network topologies.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>EIGRP MPLS VPN PE-CE Site of Origin</td>
<td>Cisco IOS 16.6.1</td>
<td>The EIGRP MPLS VPN PE-CE Site of Origin feature introduces the capability to filter MPLS traffic on a per-site basis for Enhanced Interior Gateway Routing Protocol (EIGRP) networks. Site of Origin (SoO) filtering is configured at the interface level and is used to manage MPLS VPN traffic and to prevent transient routing loops from occurring in complex and mixed network topologies.</td>
</tr>
</tbody>
</table>
CHAPTER 4

Configuring Ethernet-over-MPLS (EoMPLS) and Pseudowire Redundancy (PWR)

• Finding Feature Information, on page 29
• Configuring EoMPLS, on page 29
• Configuring Pseudowire Redundancy, on page 39

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.

Configuring EoMPLS

Information About EoMPLS

EoMPLS is one of the AToM transport types. EoMPLS works by encapsulating Ethernet PDUs in MPLS packets and forwarding them across the MPLS network. Each PDU is transported as a single packet.

Cisco IOS XE Everest 16.6.1 supports only the following mode:

• Port mode—Allows all traffic on a port to share a single VC across an MPLS network. Port mode uses VC type 5.
### Prerequisites for EoMPLS

Before you configure EoMPLS, ensure that the network is configured as follows:

- Configure IP routing in the core so that the PE routers can reach each other through IP.
- Configure MPLS in the core so that a label switched path (LSP) exists between the PE routers.
- Configure no switchport, no keepalive and no ip address before configuring xconnect on the attachment circuit.
- For load-balancing, port-channel load-balance command is mandatory to be configured.

### Restrictions for EoMPLS

- This feature is not supported on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.
- VLAN mode is not supported. Ethernet Flow Point is not supported.
- Port-channel as attachment circuit is not supported
- QoS : Customer DSCP Re-marking is not supported with VPWS and EoMPLS.
- VCCV Ping with explicit null is not supported.
- L2 VPN Interworking is not supported.
- L2 Protocol Tunneling CLI is not supported.
- Untagged, tagged and 802.1Q in 802.1Q are supported as incoming traffic.
- Flow Aware Transport Pseudowire Redundancy (FAT PW) is supported only in Protocol-CLI mode. Supported load balancing parameters are Source IP, Source MAC address, Destination IP and Destination MAC address.
- Enabling or disabling Control word is supported.
- MPLS QoS is supported in Pipe and Uniform Mode. Default mode is Pipe Mode.
- Both – the legacy xconnect and Protocol-CLI (interface pseudowire configuration) modes are supported.

By default, EoMPLS PW tunnels all protocols like CDP, STP. EoMPLS PW cannot perform selective protocol tunneling as part of L2 Protocol Tunneling CLI.
Configuring Port-Mode EoMPLS

Port-Mode EoMPLS can be configured in two modes:

- Xconnect Mode
- Protocol CLI Method

Xconnect Mode

To configure port-mode EoMPLS in xconnect mode, perform the following task:

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `interface interface-id`
4. `no switchport`
5. `no ip address`
6. `no keepalive`
7. `xconnect peer-device-id vc-id encapsulation mpls`
8. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td><em>Example:</em> <code>Device&gt; enable</code></td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><em>Example:</em> <code>Device# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>interface interface-id</code></td>
<td>Defines the interface to be configured as a trunk, and enters interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td><em>Example:</em> <code>Device(config)# interface TenGigabitEthernet1/0/36</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>no switchport</code></td>
<td>For physical ports only, enters Layer 3 mode.</td>
</tr>
<tr>
<td></td>
<td><em>Example:</em> <code>Device(config-if)# no switchport</code></td>
<td></td>
</tr>
</tbody>
</table>
### Protocol CLI Method

To configure port-mode EoMPLS in protocol-CLI mode, perform the following task:

#### SUMMARY STEPS

1. enable
2. configure terminal
3. port-channel load-balance dst-ip
4. interface interface-id
5. no switchport
6. no ip address
7. no keepalive
8. exit
9. interface pseudowire number
10. encapsulation mpls
11. neighbor peer-device-id vc-id
12. load-balance dst-ip
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 1 | `enable` | Enables privileged EXEC mode.  
Example:  
Device> `enable`  
- Enter your password if prompted. |
| Step 2 | `configure terminal` | Enters global configuration mode.  
Example:  
Device# `configure terminal` |
| Step 3 | `port-channel load-balance dst-ip` | Sets the load-distribution method to the destination IP address.  
Example:  
Device(config)# `port-channel load-balance`  
192.168.2.25  
- `dst-ip` — Destination IP address |
| Step 4 | `interface interface-id` | Defines the interface to be configured as a trunk, and enters interface configuration mode.  
Example:  
Device(config)# `interface TenGigabitEthernet1/0/21` |
| Step 5 | `no switchport` | For physical ports only, enters Layer 3 mode.  
Example:  
Device(config-if)# `no switchport` |
| Step 6 | `no ip address` | Ensures that there is no IP address assigned to the physical port.  
Example:  
Device(config-if)# `no ip address` |
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>no keepalive</td>
<td>Ensures that the device does not send keepalive messages.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-if)# no keepalive</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>exit</td>
<td>Exits interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-if)# exit</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>interface pseudowire number</td>
<td>Establishes an interface pseudowire with a value that you specify and enters pseudowire configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-if)# interface pseudowire 17</td>
<td></td>
</tr>
<tr>
<td></td>
<td>number — Specifies the number of the pseudowire to be configured.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>encapsulation mpls</td>
<td>Specifies the tunneling encapsulation.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-if)# encapsulation mpls</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>neighbor peer-device-id vc-id</td>
<td>Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-if)# neighbor 4.4.4.4 17</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>load-balance dst-ip</td>
<td>Enables edge load balancing of traffic across multiple core facing interfaces using equal cost multipaths (ECMP).</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-if)# load-balance 192.168.2.25</td>
<td></td>
</tr>
<tr>
<td></td>
<td>dst-ip — Destination IP address</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>load-balance flow-label both</td>
<td>Enables core load balancing based on flow-labels.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-if)# load-balance flow-label both</td>
<td></td>
</tr>
</tbody>
</table>
### Configuration Examples for EoMPLS

#### Figure 2: EoMPLS Topology

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 14</strong></td>
<td><code>l2vpn xconnect context context-name</code>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>Device(config-if)# l2vpn xconnect context vpws17</code></td>
<td>Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect context configuration mode.</td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td><code>member interface-id</code>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>Device(config-if)# member TenGigabitEthernet1/0/21</code></td>
<td>Specifies interface that forms a Layer 2 VPN (L2VPN) cross connect.</td>
</tr>
<tr>
<td><strong>Step 16</strong></td>
<td><code>member pseudowire number</code>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>Device(config-if)# member pseudowire 17</code></td>
<td>Specifies pseudowire interface that forms a Layer 2 VPN (L2VPN) cross connect.</td>
</tr>
<tr>
<td><strong>Step 17</strong></td>
<td><code>end</code>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>Device(config)# end</code></td>
<td>Returns to privileged EXEC mode.</td>
</tr>
</tbody>
</table>
### PE Configuration

```plaintext
mpls ip
mpls label protocol ldp
mpls ldp graceful-restart
cmpls ldp router-id loopback 1 force
interface Loopback1
ip address 1.1.1.1 255.255.255.255

ip ospf 100 area 0
router ospf 100
router-id 1.1.1.1
nsf
system mtu 9198
port-channel load-balance dst-ip
interface GigabitEthernet2/0/39
no switchport
no ip address
no keepalive

interface pseudowire101
encapsulation mpls
neighbor 4.4.4.4 101
load-balance flow ip dst-ip
load-balance flow-label both
l2vpn xconnect context pw101
member pseudowire101
member GigabitEthernet2/0/39

interface TenGigabitEthernet3/0/10
switchport trunk allowed vlan 142
switchport mode trunk
channel-group 42 mode active

interface Port-channel42
switchport trunk allowed vlan 142
switchport mode trunk

interface Vlan142
ip address 142.1.1.1 255.255.255.0

ip ospf 100 area 0
mpls ip
mpls label protocol ldp
```

### CE Configuration

```plaintext
interface GigabitEthernet1/0/33
switchport trunk allowed vlan 912
switchport mode trunk spanning-tree portfast trunk

interface Vlan912
ip address 10.91.2.3 255.255.255.0
```

The following is a sample output of `show mpls l2 vc vcid vc-id detail` command:

```
Local interface: Gi1/0/1 up, line protocol up, Ethernet up
  Destination address: 1.1.1.1, VC ID: 101, VC status: up
Output interface: Vl182, imposed label stack {17 16}
Preferred path: not configured
Default path: active
Next hop: 182.1.1.1
Load Balance: ECMP
flow classification: ip dst-ip
Create time: 06:22:11, last status change time: 05:58:42
```
Last label FSM state change time: 05:58:42  
Signaling protocol: LDP, peer 1.1.1.1:0 up  
Targeted Hello: 4.4.4.4(LDP Id) -> 1.1.1.1, LDP is UP  
Graceful restart: not configured and not enabled  
Non stop routing: not configured and not enabled  
Status TLV support (local/remote) : enabled/supported  
LDP route watch : enabled  
Label/status state machine : established, LruRru  
Last local dataplane status rcvd: No fault  
Last BFD dataplane status rcvd: Not sent  
Last BFD peer monitor status rcvd: No fault  
Last local AC circuit status rcvd: No fault  
Last local AC circuit status sent: No fault  
Last local PW i/f circ status rcvd: No fault  
Last local LDP TLV status sent: No fault  
Last remote LDP TLV status rcvd: No fault  
Last remote LDP ADJ status rcvd: No fault  
MPLS VC labels: local 512, remote 16  
Group ID: local n/a, remote 0  
MTU: local 9198, remote 9198  
Remote interface description: Sequencing: receive disabled, send disabled  
Control Word: On (configured: autosense)  
SSO Descriptor: 1.1.1.1/101, local label: 512  
Dataplane:  
SSM segment/switch IDs: 4096/4096 (used), PWID: 1  
VC statistics: transit packet totals: receive 172116845, send 172105364  
transit byte totals: receive 176837217071, send 172103349728  
transit packet drops: receive 0, seq error 0, send 0  

The following is a sample output of `show l2vpn atom vc vcid vc-id detail` command:

pseudo Wire101 is up, VC status is up PW type: Ethernet  
Create time: 06:30:41, last status change time: 06:07:12  
Last label FSM state change time: 06:07:12  
Destination address: 1.1.1.1 VC ID: 101  
Output interface: Vl182, imposed label stack {17 16}  
Preferred path: not configured  
Default path: active  
Next hop: 182.1.1.1  
Load Balance: ECMP  
Flow classification: ip dst-ip  
Member of xconnect service pw101  
Associated member G11/0/1 is up, status is up  
Interworking type is Like2Like  
Service id: 0xe5000001  
Signaling protocol: LDP, peer 1.1.1.1:0 up  
Targeted Hello: 4.4.4.4(LDP Id) -> 1.1.1.1, LDP is UP  
Graceful restart: not configured and not enabled  
Non stop routing: not configured and not enabled  
PWid FEC (128), VC ID: 101 Status TLV support (local/remote) : enabled/supported  
LDP route watch : enabled  
Label/status state machine : established, LruRru
Local dataplane status received: No fault
BFD dataplane status received: Not sent
BFD peer monitor status received: No fault
Status received from access circuit: No fault
Status sent to access circuit: No fault
Status received from pseudowire i/f: No fault
Status sent to network peer: No fault
Status received from network peer: No fault
Adjacency status of remote peer: No fault
Sequencing: receive disabled, send disabled

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Local</th>
<th>Remote</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label 512</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Group ID n/a</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Interface:

<table>
<thead>
<tr>
<th>MTU</th>
<th>9198</th>
<th>9198</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control word on</td>
<td>on</td>
<td>on</td>
</tr>
<tr>
<td>PW type</td>
<td>Ethernet</td>
<td>Ethernet</td>
</tr>
<tr>
<td>VCCV CV type 0x02</td>
<td>0x02</td>
<td></td>
</tr>
<tr>
<td>VCCV CC type 0x06</td>
<td>0x06</td>
<td></td>
</tr>
<tr>
<td>Status TLV</td>
<td>enabled</td>
<td>supported</td>
</tr>
<tr>
<td>Flow Label T=1, R=1</td>
<td>T=1, R=1</td>
<td></td>
</tr>
<tr>
<td>SSO Descriptor: 1.1.1.1/101, local label: 512</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Dataplane:

SSM segment/switch IDs: 4096/4096 (used), PWID: 1
Rx Counters: 176196691 input transit packets, 181028952597 bytes
0 drops, 0 seq err
Tx Counters: 176184928 output transit packets, 176182865992 bytes
0 drops

The following is a sample output of show mpls forwarding-table network mask command.

<table>
<thead>
<tr>
<th>Local</th>
<th>Outgoing Prefix</th>
<th>Bytes Label</th>
<th>Outgoing Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Label or Tunnel Id</td>
<td>Switched</td>
<td>interface</td>
</tr>
<tr>
<td>57</td>
<td>No Label</td>
<td>1.1.1.1/32</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>No Label</td>
<td>1.1.1.1/32</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>No Label</td>
<td>1.1.1.1/32</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>No Label</td>
<td>1.1.1.1/32</td>
<td>0</td>
</tr>
</tbody>
</table>
Configuring Pseudowire Redundancy

Information About Pseudowire Redundancy

The L2VPN Pseudowire Redundancy feature enables you to configure your network to detect a failure in the network and reroute the Layer 2 (L2) service to another endpoint that can continue to provide service. This feature provides the ability to recover from a failure either of the remote provider edge (PE) router or of the link between the PE and customer edge (CE) routers.

Pseudowire Redundancy (PWR) can be configured using both – the xconnect and the protocol-CLI method.

Prerequisites for Pseudowire Redundancy

- Configure no switchport, no keepalive and no ip address before configuring xconnect mode to connect the attachment circuit.
- For load-balancing, port-channel load-balance command is mandatory to be configured.

Restrictions for Pseudowire Redundancy

- This feature is not supported on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.
- VLAN mode, EFP (Ethernet Flow Point) and IGMP Snooping is not supported.
- PWR is supported with port mode EoMPLS only.
- Untagged, tagged and 802.1Q in 802.1Q are supported as incoming traffic.
- Flow Label for ECMP Load balancing in core network based on customer’s source IP, destination IP, source MAC and destination MAC.
- Enabling or disabling Control word is supported.
- MPLS QoS is supported in Pipe and Uniform Mode. Default mode is Pipe Mode.
- Port-channel as attachment circuit is not supported.
- QoS : Customer DSCP Re-marking is not supported with VPWS and EoMPLS.
- VCCV Ping with explicit null is not supported.
- L2 VPN Interworking is not supported.
- Not more than one backup pseudowire supported.
- PW redundancy group switchover is not supported.

Configuring Pseudowire Redundancy

Pseudowire Redundancy can be configured in two modes:
To configure pseudowire redundancy in xconnect mode, perform the following task:

- Xconnect Mode
- Protocol CLI Method

**Xconnect Mode**

To enable load balance, use the corresponding load-balance commands from Xconnect Mode, on page 31 section of Configuring Port-Mode EoMPLS.

---

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `interface interface-id`
4. `no switchport`
5. `no ip address`
6. `no keepalive`
7. `xconnect peer-device-id vc-id encapsulation mpls`
8. `backup peer peer-router-ip-addr vcid vc-id [ priority value ]`
9. `end`

---

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> interface interface-id</td>
<td>Defines the interface to be configured as a trunk, and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Example: Device(config)# interface GigabitEthernet1/0/44</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> no switchport</td>
<td>For physical ports only, enters Layer 3 mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>
### Protocol CLI Method

To configure pseudowire redundancy in protocol-CLI mode, perform the following task:

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Device(config-if)# no switchport</strong></td>
<td>Ensures that there is no IP address assigned to the physical port.</td>
</tr>
<tr>
<td><strong>Step 5</strong> no ip address <strong>Example:</strong> Device(config-if)# no ip address</td>
<td>Ensures that the device does not send keepalive messages.</td>
</tr>
<tr>
<td><strong>Step 6</strong> no keepalive <strong>Example:</strong> Device(config-if)# no keepalive</td>
<td>Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports.</td>
</tr>
<tr>
<td><strong>Step 7</strong> xconnect peer-device-id vc-id encapsulation mpls <strong>Example:</strong> Device(config-if)# xconnect 1.1.1.1 117 encapsulation mpls</td>
<td>Specifies a redundant peer for a pseudowire virtual circuit (VC).</td>
</tr>
<tr>
<td><strong>Step 8</strong> backup peer peer-router-ip-addr vc-id vc-id [ priority value ] <strong>Example:</strong> Device(config-if)# backup peer 6.6.6.6 118 priority 9</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Step 9</strong> end <strong>Example:</strong> Device(config-if)# end</td>
<td></td>
</tr>
</tbody>
</table>
3. `interface interface-id`
4. `no switchport`
5. `no ip address`
6. `no keepalive`
7. `exit`
8. `interface pseudowire number`
9. `encapsulation mpls`
10. `neighbor peer-device-id vc-id`
11. `exit`
12. `interface pseudowire number`
13. `encapsulation mpls`

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
  • Enter your password if prompted. |
<p>| Example:          |         |
| Device&gt; enable    |         |
| <strong>Step 2</strong> configure terminal | Enters global configuration mode.  |
| Example:          |         |
| Device# configure terminal |         |
| <strong>Step 3</strong> interface interface-id | Defines the interface to be configured as a trunk, and enters interface configuration mode. |
| Example:          |         |
| Device(config)# interface GigabitEthernet2/0/39 |         |
| <strong>Step 4</strong> no switchport | For physical ports only, enters Layer 3 mode. |
| Example:          |         |
| Device(config-if)# no switchport |         |
| <strong>Step 5</strong> no ip address | Ensures that there is no IP address assigned to the physical port. |
| Example:          |         |
| Device(config-if)# no ip address |         |</p>
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>no keepalive</td>
<td>Ensures that the device does not send keepalive messages.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-if)# no keepalive</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>exit</td>
<td>Exits interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-if)# exit</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>interface pseudowire <em>number</em></td>
<td>Establishes an interface pseudowire with a value that you specify and</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td>enters pseudowire configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Device(config)# interface pseudowire 101</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>encapsulation mpls</td>
<td>Specifies the tunneling encapsulation.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-if)# encapsulation mpls</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>neighbor <em>peer-device-id</em> <em>vc-id</em></td>
<td>Specifies the peer IP address and virtual circuit (VC) ID value of a</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td>Layer 2 VPN (L2VPN) pseudowire.</td>
</tr>
<tr>
<td></td>
<td>Device(config-if)# neighbor 4.4.4.4 101</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>exit</td>
<td>Exits interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-if)# exit</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>interface pseudowire <em>number</em></td>
<td>Establishes an interface pseudowire with a value that you specify and</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td>enters pseudowire configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Device(config)# interface pseudowire 102</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>encapsulation mpls</td>
<td></td>
</tr>
</tbody>
</table>
Configuration Examples for Pseudowire Redundancy

### PE Configuration

```
mpls ip
mpls label protocol ldp
mpls ldp graceful-restart
mpls ldp router-id loopback1 force !
interface Loopback1
  ip address 1.1.1.1 255.255.255.255
  ip ospf 100 area 0
  router ospf 100
  router-id 1.1.1.1
  nsf !
interface GigabitEthernet2/0/39
  no switchport
  no ip address
  no keepalive !
interface pseudowire101
  encapsulation mpls
  neighbor 4.4.4.4 101 !
interface pseudowire102
  encapsulation mpls
  neighbor 3.3.3.3 101
  l2vpn xconnect context pw101
  member pseudowire101 group pwgrp1 priority 1
  member pseudowire102 group pwgrp1 priority 15
  member GigabitEthernet2/0/39 !
interface TenGigabitEthernet3/0/10
  switchport trunk allowed vlan 142
  switchport mode trunk
  channel-group 42 mode active !
interface Port-channel42
  switchport trunk allowed vlan 142
  switchport mode trunk !
interface Vlan142
  ip address 142.1.1.1 255.255.255.0
  ip ospf 100 area 0
mpls ip
mpls label protocol ldp !
```

### CE Configuration

```
interface GigabitEthernet1/0/33
  switchport trunk allowed vlan 912
  switchport mode trunk spanning-tree portfast trunk !
interface Vlan912
  ip address 10.91.2.3 255.255.255.0 !
```

The following is sample output of the `show mpls l2transport vc ve-id` command:

```
Device# show mpls l2transport vc 101
  Local intf  Local circuit  Dest address  VC ID  Status
  -----------------  -------------  -------------  --------  -------
```

---

### Configuration Examples for Pseudowire Redundancy

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>

Multiprotocol Label Switching (MPLS) Configuration Guide, Cisco IOS XE Fuji 16.8.x (Catalyst 9300 Switches)
<table>
<thead>
<tr>
<th>Local intf</th>
<th>Local circuit</th>
<th>Dest address</th>
<th>VC ID</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gi2/0/39</td>
<td>Ethernet</td>
<td>3.3.3.3</td>
<td>102</td>
<td>STANDBY</td>
</tr>
</tbody>
</table>

Device# `show mpls l2transport vc 102`
Configuration Examples for Pseudowire Redundancy
CHAPTER 5

Configuring IPv6 Provider Edge over MPLS (6PE)

• Finding Feature Information, on page 47
• Configuring 6PE, on page 47

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.

Configuring 6PE

Information About 6PE

6PE is a technique that provides global IPv6 reachability over IPv4 MPLS. It allows one shared routing table for all other devices. 6PE allows IPv6 domains to communicate with one another over the IPv4 without an explicit tunnel setup, requiring only one IPv4 address per IPv6 domain.

While implementing 6PE, the provider edge routers are upgraded to support 6PE, while the rest of the core network is not touched (IPv6 unaware). This implementation requires no reconfiguration of core routers because forwarding is based on labels rather than on the IP header itself. This provides a cost-effective strategy for deploying IPv6. The IPv6 reachability information is exchanged by PE routers using multiprotocol Border Gateway Protocol (mp-iBGP) extensions.

6PE relies on mp-iBGP extensions in the IPv4 network configuration on the PE router to exchange IPv6 reachability information in addition to an MPLS label for each IPv6 address prefix to be advertised. PE routers are configured as dual stacks, running both IPv4 and IPv6, and use the IPv4 mapped IPv6 address for IPv6 prefix reachability exchange. The next hop advertised by the PE router for 6PE and 6VPE prefixes is still the IPv4 address that is used for IPv4 L3 VPN routes. A value of ::FFFF: is prepended to the IPv4 next hop, which is an IPv4-mapped IPv6 address.

The following figure illustrates the 6PE topology.
### Scale Numbers

<table>
<thead>
<tr>
<th>Platform</th>
<th>3650</th>
<th>3850</th>
<th>9300</th>
<th>9500</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPLS L3VPN VRF</td>
<td>127</td>
<td>127</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>MPLS L3VPN Routes VRF</td>
<td>7k</td>
<td>7k</td>
<td>7k</td>
<td>32k</td>
</tr>
<tr>
<td>(All provider edge should be in Per-VRF label allocation mode)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPLS L3VPN Routes Prefix</td>
<td>3k</td>
<td>3k</td>
<td>3k</td>
<td>4k</td>
</tr>
</tbody>
</table>

**The prefix numbers listed in the table are for IPv4. IPv6 numbers will be half of IPv4.**

### Prerequisites for 6PE

Redistribute PE-CE IGP IPv6 routes into core BGP and vice-versa

### Restrictions for 6PE

eBGP as CE-PE is not supported. Static Routes, OSPFv3, ISIS, RIPv2 are supported as CE-PE.

### Configuring 6PE

Ensure that you configure 6PE on PE routers participating in both the IPv4 cloud and IPv6 clouds.

BGP running on a PE router should establish (IPv4) neighborhood with BGP running on other PEs. Subsequently, it should advertise the IPv6 prefixes learnt from the IPv6 table to the neighbors. The IPv6 prefixes advertised by BGP would automatically have IPv4-encoded-IPv6 addresses as the nexthop-address in the advertisement.

To configure 6PE, complete the following steps:
### SUMMARY STEPS

1. enable
2. configure terminal
3. ipv6 unicast-routing
4. router bgp as-number
5. bgp router-id interface interface-id
6. bgp log-neighbor-changes
7. bgp graceful-restart
8. neighbor \{ ip-address \ ipv6-address \ peer-group-name \} remote-as as-number
9. neighbor \{ ip-address \ ipv6-address \ peer-group-name \} update-source interface-type interface-number
10. address-family ipv6
11. redistribute protocol as-number match \{ internal \ external 1 \ external 2 \}
12. neighbor \{ ip-address \ ipv6-address \ peer-group-name \} activate
13. neighbor \{ ip-address \ ipv6-address \ peer-group-name \} send-label
14. exit-address-family
15. end

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ipv6 unicast-routing</td>
<td>Enables the forwarding of IPv6 unicast datagrams.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# ipv6 unicast-routing</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> router bgp as-number</td>
<td>Enters the number that identifies the autonomous system (AS) in which the router resides.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# router bgp 65001</td>
<td>as-number—Autonomous system number. Range for 2-byte numbers is 1 to 65535. Range for 4-byte numbers is 1.0 to 65535.65535.</td>
</tr>
<tr>
<td><strong>Step 5</strong> bgp router-id interface interface-id</td>
<td>Configures a fixed router ID for the local Border Gateway Protocol (BGP) routing process.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Device(config-router)# bgp router-id interface Loopback1</td>
<td>Enables logging of BGP neighbor resets.</td>
</tr>
<tr>
<td><strong>Step 6</strong> bgp log-neighbor-changes</td>
<td>Enables the Border Gateway Protocol (BGP) graceful restart capability globally for all BGP neighbors.</td>
</tr>
<tr>
<td>Example: Device(config-router)# bgp log-neighbor-changes</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> bgp graceful-restart</td>
<td></td>
</tr>
<tr>
<td>Example: Device(config-router)# bgp graceful-restart</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> neighbor { ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td>Example: Device(config-router)# neighbor 33.33.33.33 remote-as 65001</td>
<td>• ip-address—IP address of a peer router with which routing information will be exchanged.</td>
</tr>
<tr>
<td></td>
<td>• ipv6-address—IPv6 address of a peer router with which routing information will be exchanged.</td>
</tr>
<tr>
<td></td>
<td>• peer-group-name—Name of the BGP peer group.</td>
</tr>
<tr>
<td></td>
<td>• remote-as—Specifies a remote autonomous system.</td>
</tr>
<tr>
<td></td>
<td>• as-number—Number of an autonomous system to which the neighbor belongs, ranging from 1 to 65535.</td>
</tr>
<tr>
<td><strong>Step 9</strong> neighbor { ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td>Example: Device(config-router)# neighbor 33.33.33.33 update-source Loopback1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> address-family ipv6</td>
<td>Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard IPv6 address prefixes.</td>
</tr>
<tr>
<td>Example: Device(config-router)# address-family ipv6</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> redistribute protocol as-number match { internal</td>
<td>external 1</td>
</tr>
<tr>
<td>Example: Device(config-router-af)# redistribute ospf 11 match internal external 1</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td></td>
</tr>
<tr>
<td>neighbor { ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config-router-af)# neighbor 33.33.33 activate</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td></td>
</tr>
<tr>
<td>neighbor { ip-address</td>
<td>ipv6-address</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config-router-af)# neighbor 33.33.33 send-label</td>
<td></td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td></td>
</tr>
<tr>
<td>exit-address-family</td>
<td>Exits BGP address-family submode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config-router-af)# exit-address-family</td>
<td></td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td></td>
</tr>
<tr>
<td>end</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# end</td>
<td></td>
</tr>
</tbody>
</table>

### Configuration Examples for 6PE

*Figure 4: 6PE Topology*
**PE Configuration**

<table>
<thead>
<tr>
<th>Command</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>address-family ipv6 unicast</td>
<td>redistribute bgp 65001</td>
</tr>
<tr>
<td>redistribute bgp 65001</td>
<td>exit-address-family</td>
</tr>
<tr>
<td>!</td>
<td>router bgp 65001</td>
</tr>
<tr>
<td>bgp router-id interface Loopback1</td>
<td>bgp graceful-restart</td>
</tr>
<tr>
<td>bgp log-neighbor-changes</td>
<td>neighbor 33.33.33.33 remote-as 650001</td>
</tr>
<tr>
<td>neighbor 33.33.33.33 remote-as 65001</td>
<td>neighbor 33.33.33.33 update-source Loopback1</td>
</tr>
<tr>
<td>address-family ipv4</td>
<td>neighbor 33.33.33 activate</td>
</tr>
<tr>
<td>neighbor 33.33.33 activate</td>
<td>address-family ipv6</td>
</tr>
<tr>
<td>redistribute ospf 11 match internal external 1</td>
<td>neighbor 33.33.33 activate</td>
</tr>
<tr>
<td>1 external 2 include-connected</td>
<td>neighbor 33.33.33 activate</td>
</tr>
<tr>
<td>neighbor 33.33.33 activate</td>
<td>neighbor 33.33.33 activate</td>
</tr>
<tr>
<td>neighbor 33.33.33 activate</td>
<td>neighbor 33.33.33 activate</td>
</tr>
<tr>
<td>neighbor 33.33.33 activate</td>
<td>neighbor 33.33.33 activate</td>
</tr>
</tbody>
</table>

**CE Configuration**

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipv6 unicast-routing</td>
</tr>
<tr>
<td>!</td>
</tr>
<tr>
<td>interface vlan4</td>
</tr>
<tr>
<td>no ip address</td>
</tr>
<tr>
<td>ipv6 address 10:1:1:2::2/64</td>
</tr>
<tr>
<td>ipv6 enable</td>
</tr>
<tr>
<td>ospfV3 11 ipv6 area 0</td>
</tr>
<tr>
<td>!</td>
</tr>
<tr>
<td>router ospfV3 11</td>
</tr>
<tr>
<td>address-family ipv6 unicast</td>
</tr>
<tr>
<td>exit-address-family</td>
</tr>
</tbody>
</table>

The following is a sample output of `show bgp ipv6 unicast summary`:

BGP router identifier 1.1.1.1, local AS number 100
BGP table version is 34, main routing table version 34
4 network entries using 1088 bytes of memory
4 path entries using 608 bytes of memory
4/4 BGP path/bestpath attribute entries using 1120 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2816 total bytes of memory
BGP activity 6/2 prefixes, 16/12 paths, scan interval 60 secs

```
Neighbor  V  AS  MsgRcvd  MsgSent  TblVer  InQ  OutQ  Up/Down
2.2.2.2  4 100  21  21   34   0   0   00:04:57
```

**sh ipv route**

IPv6 Routing Table - default - 7 entries

Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
       IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP
       external
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
       RL - RPL, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       1a - LISP alt, 1r - LISP site-registrations, 1d - LISP dyn-eid 1A
       - LISP away
       C 10:1:1:2::/64 [0/0]
       via Vlan4, directly connected
L 10:1:1:2::1/128 [0/0]  
via Vlan4, receive  
LC 11:11:11:11::1/128 [0/0]  
via Loopback1, receive  
B 30:1:1:2::/64 [200/0]  
via 33.33.33.33%default, indirectly connected  
B 40:1:1:2::/64 [200/0]  
via 44.44.44.44%default, indirectly connected

The following is a sample output of `show bgp ipv6 unicast` command:

BGP table version is 112, local router ID is 11.11.11.11  
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,  
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,  
x best-external, a additional-path, c RIB-compressed,  
t secondary path,  
Origin codes: i - IGP, e - EGP, ? - incomplete  
RPKI validation codes: V valid, I invalid, N Not found

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>*&gt; 10:1:1:2::/64</td>
<td>::</td>
<td>0</td>
<td>32768</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>*&gt;i 30:1:1:2::/64</td>
<td>::FFFF:33.33.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*&gt;i 40:1:1:2::/64</td>
<td>::FFFF:44.44.44</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*&gt;i 173:1:1:2::/64</td>
<td>::FFFF:33.33.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following is a sample output of `show ipv6 cef 40:1:1:2::0/64 detail` command:

40:1:1:2::/64, epoch 6, flags [rib defined all labels]  
recursive via 44.44.44.44 label 67  
nexthop 1.20.4.2 Port-channel103 label 99-(local:147)
CHAPTER 6

Configuring IPv6 VPN Provider Edge over MPLS (6VPE)

- Finding Feature Information, on page 55
- Configuring 6VPE, on page 55

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.

Configuring 6VPE

Information About 6VPE

6VPE is a mechanism to use the IPv4 backbone to provide VPN IPv6 services. It takes advantage of operational IPv4 MPLS backbones, eliminating the need for dual-stacking within the MPLS core. This translates to savings in operational costs and addresses the security limitations of the 6PE approach. 6VPE is more like a regular IPv4 MPLS-VPN provider edge, with an addition of IPv6 support within VRF. It provides logically separate routing table entries for VPN member devices.

Components of MPLS-based 6VPE Network

- VPN route target communities – A list of all other members of a VPN community.

- Multiprotocol BGP (MP-BGP) peering of VPN community PE routers – Propagates VRF reachability information to all members of a VPN community.

- MPLS forwarding – Transports all traffic between all VPN community members across a VPN service-provider network.
In the MPLS-VPN model a VPN is defined as a collection of sites sharing a common routing table. A customer site is connected to the service provider network by one or more interfaces, where the service provider associates each interface with a VPN routing table—known as the VRF table.

### Scale Numbers

<table>
<thead>
<tr>
<th>Platform</th>
<th>3650</th>
<th>3850</th>
<th>9300</th>
<th>9500</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPLS L3VPN VRF</td>
<td>127</td>
<td>127</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>MPLS L3VPN Routes VRF</td>
<td>7k</td>
<td>7k</td>
<td>7k</td>
<td>32k</td>
</tr>
<tr>
<td>(All provider edge should be in Per-VRF label allocation mode)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPLS L3VPN Routes Prefix</td>
<td>3k</td>
<td>3k</td>
<td>3k</td>
<td>4k</td>
</tr>
</tbody>
</table>

**The prefix numbers listed in the table are for IPv4. IPv6 numbers will be half of IPv4.**

### Restrictions for 6VPE

- Inter-AS and carrier supporting carrier (CSC) is not supported.
- VRF Route-Leaking is not supported.
- EIGRP and eBGP as CE-PE is not supported.
- OSPFv3, RIP, ISIS, Static Routes are supported as CE-PE.
- MPLS Label Allocation modes supported are Per-VRF and Per-Prefix. Per-Prefix is the default mode.

### Information About 6VPE

6VPE is a mechanism to use the IPv4 backbone to provide VPN IPv6 services. It takes advantage of operational IPv4 MPLS backbones, eliminating the need for dual-stacking within the MPLS core. This translates to savings in operational costs and addresses the security limitations of the 6PE approach. 6VPE is more like a regular IPv4 MPLS-VPN provider edge, with an addition of IPv6 support within VRF. It provides logically separate routing table entries for VPN member devices.

### Components of MPLS-based 6VPE Network

- VPN route target communities – A list of all other members of a VPN community.
- Multiprotocol BGP (MP-BGP) peering of VPN community PE routers – Propagates VRF reachability information to all members of a VPN community.
- MPLS forwarding – Transports all traffic between all VPN community members across a VPN service-provider network.

In the MPLS-VPN model a VPN is defined as a collection of sites sharing a common routing table. A customer site is connected to the service provider network by one or more interfaces, where the service provider associates each interface with a VPN routing table—known as the VRF table.
Configuration Examples for 6VPE

Figure 5: 6VPE Topology
<table>
<thead>
<tr>
<th>PE Configuration</th>
<th>CE Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface TenGigabitEthernet1/0/38</td>
<td>interface TenGigabitEthernet1/0/38</td>
</tr>
<tr>
<td>no switchport</td>
<td>no switchport</td>
</tr>
<tr>
<td>ip address 10.3.1.2 255.255.255.0</td>
<td>ip address 10.3.1.2 255.255.255.0</td>
</tr>
<tr>
<td>ip ospf 2 area 0</td>
<td>ip ospf 2 area 0</td>
</tr>
<tr>
<td>ipv6 address 10:111:111:111::2/64</td>
<td>ipv6 address 10:111:111:111::2/64</td>
</tr>
<tr>
<td>ipv6 enable</td>
<td>ipv6 enable</td>
</tr>
<tr>
<td>ipv6 ospf 1 area 0</td>
<td>ipv6 ospf 1 area 0</td>
</tr>
<tr>
<td>router ospfv3 1</td>
<td>router ospfv3 1</td>
</tr>
<tr>
<td>nsr</td>
<td>nsr</td>
</tr>
<tr>
<td>graceful-restart</td>
<td>graceful-restart</td>
</tr>
<tr>
<td>address-family ipv6 unicast</td>
<td>address-family ipv6 unicast</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>PE Configuration</td>
<td>CE Configuration</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------</td>
</tr>
</tbody>
</table>
| vrf definition 6VPE-1  
  rd 65001:11  
  route-target export 1:1  
  route-target import 1:1  
  !  
  address-family ipv4  
  exit-address-family  
  !  
  address-family ipv6  
  exit-address-family  
  !  
  interface TenGigabitEthernet1/0/38  
  no switchport  
  vrf forwarding 6VPE-1  
  ip address 10.3.1.1 255.255.255.0  
  ip ospf 2 area 0  
  ipv6 address 10:111:111:111::1/64  
  ipv6 enable  
  ospfv3 1 ipv6 area 0  
  !  
  router ospf 2 vrf 6VPE-1  
  router-id 1.1.11.11  
  redistribute bgp 65001 subnets  
  !  
  router ospfv3 1  
  nhr  
  graceful-restart  
  !  
  address-family ipv6 unicast vrf 6VPE-1  
  redistribute bgp 65001  
  exit-address-family  
  !  
  router bgp 65001  
  bgp router-id interface Loopback1  
  bgp log-neighbor-changes  
  bgp graceful-restart  
  neighbor 33.33.33.33 remote-as 65001  
  neighbor 33.33.33.33 update-source Loopback1  
  !  
  address-family ipv4 vrf 6VPE-1  
  redistribute ospf 2 match internal external  
  1 external 2  
  exit-address-family  
  address-family ipv6 vrf 6VPE-1  
  redistribute ospf 1 match internal external  
  1 external 2 include-connected  
  exit-address-family  
  !  
  address-family vpnv4  
  neighbor 33.33.33.33 activate  
  neighbor 33.33.33.33 send-community both  
  neighbor 44.44.44.44 activate  
  neighbor 44.44.44.44 send-community both  
  neighbor 55.55.55.55 activate  
  neighbor 55.55.55.55 send-community both  
  exit-address-family  
  !  
  address-family vpnv6  
  neighbor 33.33.33.33 activate  
  neighbor 33.33.33.33 send-community both  
  neighbor 44.44.44.44 activate  |
**PE Configuration**

<table>
<thead>
<tr>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>neighbor 44.44.44.44 send-community both</td>
</tr>
<tr>
<td>neighbor 55.55.55.55 activate</td>
</tr>
<tr>
<td>neighbor 55.55.55.55 send-community both</td>
</tr>
<tr>
<td>exit-address-family !</td>
</tr>
</tbody>
</table>

**CE Configuration**

<table>
<thead>
<tr>
<th>Configuration</th>
</tr>
</thead>
</table>

The following is a sample output of `show mpls forwarding-table vrf`:

<table>
<thead>
<tr>
<th>Local Outgoing Prefix Bytes Label Outgoing Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label Label or Tunnel Id Switched interface</td>
</tr>
<tr>
<td>29 No Label A:A::565::/64[V] \ 0 aggregate/VRF601</td>
</tr>
</tbody>
</table>

The following is a sample output of `show vrf counter` command:

 Maximum number of VRFs supported: 256
 Maximum number of IPv4 VRFs supported: 256
 Maximum number of IPv6 VRFs supported: 256
 Maximum number of platform iVRFs supported: 10
 Current number of VRFs: 127
 Current number of IPv4 VRFs: 6
 Current number of IPv6 VRFs: 127
 Current number of VRFs in delete state: 0
 Current number of platform iVRFs: 1

The following is a sample output of `show ipv6 route vrf` command:

 IPv6 Routing Table - VRF1 - 8 entries Codes: C - Connected, L - Local, S - Static, U - Per-user Static route B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2 IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP external ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect RL - RPL, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1 OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2 la - LISP alt, lr - LISP site-registrations, ld - LISP dyn-eid 1A - LISP away

 B 1:1:1:1::1/128 [200/1] via 1.1.1.11%default, indirectly connected
 O 2:2:2:2::2/128 [110/1] via FE80::A2E0:AF0:FE30:3E40,
 TenGigabitEthernet1/0/7
 B 3:3:3:3:3:3/128 [200/1] via 3.3.3.33%default, indirectly connected
 B 10:1:1:1::/64 [200/0] via 1.1.1.11%default, indirectly connected
 C 10:2:2:2::/64 [0/0] via TenGigabitEthernet1/0/7, directly connected
 L 10:2:2:2::1/128 [0/0] via TenGigabitEthernet1/0/7, receive
 B 10:3:3:3:3:3/64 [200/0] via 3.3.3.33%default, indirectly connected
 L FF00::/8 [0/0] via Null10, receive

---

**Multiprotocol Label Switching (MPLS) Configuration Guide, Cisco IOS XE Fuji 16.8.x (Catalyst 9300 Switches)**
Configuring IP-aware Netflow for VRF Ingress

Restrictions for IP-aware Netflow for VRF Ingress

- Supported only on the following SKUs:
  - C9300-24T
  - C9300-24P
  - C9300-24U
  - C9300-48T
  - C9300-48P
  - C9300-48U
  - C9300-24UX
  - C9300-48UXM

- IP-aware VRF ingress Netflow is supported with IPv4, IPv6 and MVPNv4 as CE facing interface
- Supported only on layer 3 interface
- Supported only for ingress traffic on the VRF interface
- Supported only for MPLS L3 VPN VRF interface
- IP aware VRF ingress Netflow on MVPNv6 as CE facing interface is not supported
- Not supported on portchannel, SVI as CE facing interface
- Not supported for egress traffic on the VRF interface
- Not supported on MPLS L2VPN Attachment circuit interface
Information About IP-aware Netflow for VRF Ingress

This feature enables collecting the virtual routing and forwarding (VRF) ID from incoming packets on a router by applying an input flow monitor having a flow record that collects the VRF ID as a key or a non-key field.

### Table 13: Scale Numbers

<table>
<thead>
<tr>
<th>Platform</th>
<th>SDM Template</th>
<th>Max IPv4 Flows</th>
<th>Max IPv6 Flows</th>
</tr>
</thead>
<tbody>
<tr>
<td>3650</td>
<td>Advanced</td>
<td>8k</td>
<td>4k</td>
</tr>
<tr>
<td>3850</td>
<td>Advanced</td>
<td>8k</td>
<td>4k</td>
</tr>
<tr>
<td>9300</td>
<td>Access</td>
<td>16k</td>
<td>8k</td>
</tr>
<tr>
<td>9400</td>
<td>Distribution</td>
<td>32k</td>
<td>16k</td>
</tr>
<tr>
<td>9500</td>
<td>Access</td>
<td>32k</td>
<td>16k</td>
</tr>
</tbody>
</table>

How to Configure IP-aware Netflow for VRF Ingress

### Creating a Flow Record

Perform the following task to create a flow record.

Step 1

**SUMMARY STEPS**

1. configure terminal
2. flow record flow_record_name
3. description description
4. match ipv4 version
5. match ipv4 {source | destination} address
6. match ipv4 protocol
7. match transport {source-port | destination-port}
8. match ipv4 tos
9. match ipv4 ttl
10. match flow direction
11. collect counter packets long
12. collect counter bytes long
13. end
14. show flow record
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters flow record configuration mode.</td>
</tr>
<tr>
<td><code>flow record flow_record_name</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config)# flow record flow-record-1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>(Optional) Creates a description for the flow record.</td>
</tr>
<tr>
<td><code>description description</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config-flow-record)# description flow-record-1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Specifies a match to the IP version from the IPv4 header.</td>
</tr>
<tr>
<td><code>match ipv4 version</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config-flow-record)# match ipv4 version</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Specifies a match to the IPv4 source and destination address.</td>
</tr>
<tr>
<td>`match ipv4 {source</td>
<td>destination} address`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config-flow-record)# match ipv4 protocol</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Specifies a match to the IPv4 protocol.</td>
</tr>
<tr>
<td><code>match ipv4 protocol</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config-flow-record)# match ipv4 protocol</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Configures source-port or destination port as a key field for the flow record.</td>
</tr>
<tr>
<td>`match transport {source-port</td>
<td>destination-port}`</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Configures IPv4 ToS as a key field for the flow record.</td>
</tr>
<tr>
<td><code>match ipv4 tos</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config-flow-record)# match ipv4 tos</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>Configures IPv4 TTL as a key field for the flow record.</td>
</tr>
<tr>
<td><code>match ipv4 ttl</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config-flow-record)# match ipv4 ttl</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>Specifies a match to the flow identifying fields.</td>
</tr>
<tr>
<td><code>match flow direction</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config-flow-record)# match flow direction</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>Configures the number of packets seen in a flow as a non-key field and enables collecting the total number of packets from the flow.</td>
</tr>
<tr>
<td><code>collect counter packets long</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config-flow-record)# collect flow direction</code></td>
<td></td>
</tr>
</tbody>
</table>
### Creating a Flow Exporter

You can create a flow exporter to define the export parameters for a flow.

#### SUMMARY STEPS

1. `configure terminal`
2. `flow exporter flow_exporter_name`
3. `description description`
4. `destination { hostname | ipv4-address | ipv6-address }
5. `source interface-type interface-name`
6. `end`
7. `show flow exporter`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>flow exporter flow_exporter_name</code></td>
<td>Enters flow exporter configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config)# flow exporter flow-exporter-1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>description description</code></td>
<td>(Optional) Creates a description for the flow exporter.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-flow-exporter)# description</td>
<td></td>
</tr>
<tr>
<td>flow-exporter-1</td>
<td></td>
</tr>
</tbody>
</table>
Creating a Flow Monitor

You can create a flow monitor and associate it with a flow record.

**SUMMARY STEPS**

1. configure terminal
2. flow monitor monitor-name
3. description description
4. record record-name
5. exporter exporter-name
6. cache type normal {timeout | active | inactive} | type normal
7. end
8. show flow monitor

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> flow monitor monitor-name</td>
<td>Creates a flow monitor and enters flow monitor configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device (config)# flow monitor flow-monitor-1</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>(Optional) Creates a description for the flow monitor.</td>
</tr>
<tr>
<td>description <em>description</em></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device (config-flow-monitor)# description flow-monitor-1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Specifies the name of a record that was created previously.</td>
</tr>
<tr>
<td>record <em>record-name</em></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device (config-flow-monitor)# record flow-record-1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Specifies the name of an exporter that was created previously.</td>
</tr>
<tr>
<td>exporter <em>exporter-name</em></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device (config-flow-monitor)# exporter flow-exporter-1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>(Optional) Specifies to configure flow cache parameters.</td>
</tr>
<tr>
<td>cache type normal</td>
<td></td>
</tr>
<tr>
<td>{timeout</td>
<td>active</td>
</tr>
<tr>
<td></td>
<td>type normal</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device (config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode.</td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Displays information about all the flow monitors.</td>
</tr>
<tr>
<td>show flow monitor</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device # show flow monitor</td>
<td></td>
</tr>
</tbody>
</table>

### Applying Flow Monitor to an Interface

**SUMMARY STEPS**

1. configure terminal
2. interface *interface-type interface-name*
3. no switchport
4. vrf forwarding *vrf-name*
5. {ip | ipv6} flow-monitor *monitor-name input*
6. end
7. show flow interface

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
</tbody>
</table>
## Configuration Examples for IP-aware Netflow for VRF Ingress

The `show flow interface` command displays information about Netflow on the specified interface:

```
Interface TenGigabitEthernet1/0/36
FNF: monitor: v4vrfingress
direction: Input
traffic(ip): on
FNF: monitor: v6vrfingress
direction: Input
traffic(ipv6): on
```

The `show flow monitor flow-monitor-name cache` command displays the contents of the cache for the flow monitor.

```
Cache type: Normal (Platform cache)
Cache size: 10000
Current entries: 100

Flows added: 100
Flows aged: 0

IPV4 SOURCE ADDRESS: 108.3.20.100
IPV4 DESTINATION ADDRESS: 108.2.20.100
TRNS SOURCE PORT: 0
TRNS DESTINATION PORT: 0
FLOW DIRECTION: Input
IP VERSION: 4
IP TOS: 0x20
```
The show flow exporter command displays information about all the flow exporters.

Flow Exporter v4vrfingress:
- Description: User defined
- Export protocol: NetFlow Version 9
- Transport Configuration:
  - Destination type: IP
  - Destination IP address: 15.15.15.16
  - Source IP address: 15.15.15.15
  - Source Interface: TenGigabitEthernet1/0/1
  - Transport Protocol: UDP
  - Destination Port: 9995
  - Source Port: 52319
  - DSCP: 0x0
  - TTL: 255
  - Output Features: Used

Flow Exporter v6vrfingress:
- Description: User defined
- Export protocol: NetFlow Version 9
- Transport Configuration:
  - Destination type: IP
  - Destination IP address: 15.15.15.16
  - Source IP address: 15.15.15.15
  - Source Interface: TenGigabitEthernet1/0/1
  - Transport Protocol: UDP
  - Destination Port: 9995
  - Source Port: 50881
  - DSCP: 0x0
  - TTL: 255
  - Output Features: Used

The show platform software fed switch active fnf monitors-dump displays Netflow monitors dump.

FNF Monitors
============
Monitor (0x7f4afc031748):
  profile_id(c461d4fe) ref_ct(1) wdavc_monitor(0)
  wdavc_monitor_create_requested(False)
  wdavc_remote_monitoring_remote_caching(0) flags(0x0000) is_wireless(No)
  is_etta_over_fnf No ettaOrBaseProfile(00000000) etta_refcnt(0)
  field(113) size(16) param(0) flags(1) offset(0)
  field(114) size(16) param(0) flags(1) offset(16)
  field(118) size(2) param(0) flags(1) offset(32)
  field(119) size(2) param(0) flags(1) offset(34)
  field(156) size(1) param(0) flags(1) offset(36)
  field(181) size(8) param(0) flags(0) offset(37)
  field(42) size(1) param(0) flags(1) offset(45)
  field(46) size(1) param(0) flags(1) offset(46)
Feature History and Information for IP-aware Netflow for VRF Ingress

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>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Fuji 16.8.1a</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>
Feature History and Information for IP-aware Netflow for VRF Ingress
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, on page 71

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` and `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 interface 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 is 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 next-hop1`
- `ip route destination-prefix mask interface2 next-hop2`
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.
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.

SUMMARY STEPS

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

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Identify the size of the network.</td>
</tr>
<tr>
<td></td>
<td>Identify the following to determine the number of devices and ports that you need:</td>
</tr>
<tr>
<td></td>
<td>• How many customers do you need to support?</td>
</tr>
<tr>
<td></td>
<td>• How many VPNs are needed per customer?</td>
</tr>
<tr>
<td></td>
<td>• How many virtual routing and forwarding instances are there for each VPN?</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Identify the routing protocols in the core.</td>
</tr>
<tr>
<td></td>
<td>Determine which routing protocols you need in the core network.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Determine if you need MPLS VPN High Availability support.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Determine if you need Border Gateway Protocol (BGP) load sharing and redundant paths in the MPLS VPN core.</td>
</tr>
<tr>
<td></td>
<td>For configuration steps, see the “Load Sharing MPLS VPN Traffic” feature module in the MPLS Layer 3 VPNs Inter-AS and CSC Configuration Guide.</td>
</tr>
</tbody>
</table>

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

Connecting the MPLS Virtual Private Network Customers

Defining VRFs on the PE Devices to Enable Customer Connectivity

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

SUMMARY STEPS

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

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td></td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Defines the virtual private network (VPN) routing instance by assigning a virtual routing and forwarding (VRF) name and enters VRF configuration mode.</td>
</tr>
<tr>
<td>ip vrf vrf-name</td>
<td>• The vrf-name argument is the name assigned to a VRF.</td>
</tr>
<tr>
<td>Example: Device(config)# ip vrf vpn1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Creates routing and forwarding tables.</td>
</tr>
<tr>
<td>rd route-distinguisher</td>
<td>• 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:</td>
</tr>
<tr>
<td>Example: Device(config-vrf)# rd 100:1</td>
<td>• 16-bit AS number:your 32-bit number, for example, 101:3</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Step 5 route-target {import</td>
<td>Creates a route-target extended community for a VRF.\n</td>
</tr>
<tr>
<td>Step 6 exit</td>
<td>(Optional) Exits to global configuration mode.</td>
</tr>
<tr>
<td>Example: \nDevice(config-vrf)# exit</td>
<td></td>
</tr>
</tbody>
</table>

**Configuring VRF Interfaces on PE Devices for Each VPN Customer**

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

**SUMMARY STEPS**

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

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: \nDevice&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
Configuring Routing Protocols Between the PE and CE Devices

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

Verifying the Virtual Private Network Configuration

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

SUMMARY STEPS

1. `show ip vrf`

DETAILED STEPS

`show ip vrf`

Displays the set of defined VRF instances and associated interfaces. The output also maps the VRF instances to the configured route distinguisher.
Verifying Connectivity Between MPLS Virtual Private Network Sites

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

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

SUMMARY STEPS

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

DETAILED STEPS

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

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

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

**Step 1**  
enable  
Enables privileged EXEC mode.
Step 2  \texttt{show ip route vrf} \texttt{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  \texttt{show ip cef vrf} \texttt{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

<table>
<thead>
<tr>
<th>PE Configuration</th>
<th>CE Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip vrf vpn1</td>
<td>ip cef</td>
</tr>
<tr>
<td>rd 100:1</td>
<td>mpls ldp router-id Loopback0 force</td>
</tr>
<tr>
<td>route-target export 100:1</td>
<td>mpls label protocol ldp</td>
</tr>
<tr>
<td>route-target import 100:1</td>
<td>!</td>
</tr>
<tr>
<td>ip cef</td>
<td>interface Loopback0</td>
</tr>
<tr>
<td>mpls ldp router-id Loopback0 force</td>
<td>ip address 10.0.0.9 255.255.255.255</td>
</tr>
<tr>
<td>mpls label protocol ldp</td>
<td>!</td>
</tr>
<tr>
<td>! interface Loopback0</td>
<td>interface</td>
</tr>
<tr>
<td>ip address 10.0.0.1 255.255.255.255</td>
<td>ip address 192.0.2.1 255.255.255.0</td>
</tr>
<tr>
<td>! interface</td>
<td>no cdp enable</td>
</tr>
<tr>
<td>ip vrf forwarding vpn1</td>
<td>router rip</td>
</tr>
<tr>
<td>ip address 192.0.2.3 255.255.255.0</td>
<td>version 2</td>
</tr>
<tr>
<td>no cdp enable</td>
<td>timers basic 30 60 60 120</td>
</tr>
<tr>
<td>interface</td>
<td>redistribute connected</td>
</tr>
<tr>
<td>ip address 192.0.2.2 255.255.255.0</td>
<td>network 10.0.0.0</td>
</tr>
<tr>
<td>mpls label protocol ldp</td>
<td>network 192.0.2.0</td>
</tr>
<tr>
<td>mpls ip</td>
<td>no auto-summary</td>
</tr>
<tr>
<td>! router rip</td>
<td>! address-family ipv4 vrf vpn1</td>
</tr>
<tr>
<td>version 2</td>
<td>version 2</td>
</tr>
<tr>
<td>timers basic 30 60 60 120</td>
<td>redistribute bgp 100 metric transparent</td>
</tr>
<tr>
<td>! address-family ipv4 vrf vpn1</td>
<td>network 192.0.2.0</td>
</tr>
<tr>
<td>version 2</td>
<td>distribute-list 20 in</td>
</tr>
<tr>
<td>redistribute bgp 100 metric transparent</td>
<td>no auto-summary</td>
</tr>
<tr>
<td>network 192.0.2.0</td>
<td>exit-address-family</td>
</tr>
<tr>
<td>distribute-list 20 in</td>
<td>! router bgp 100</td>
</tr>
<tr>
<td>no auto-summary</td>
<td>no synchronization</td>
</tr>
<tr>
<td>exit-address-family</td>
<td>bgp log-neighbor changes</td>
</tr>
<tr>
<td>! router bgp 100</td>
<td>neighbor 10.0.0.3 remote-as 100</td>
</tr>
<tr>
<td>no synchronization</td>
<td>neighbor 10.0.0.3 update-source Loopback0</td>
</tr>
<tr>
<td>bgp log-neighbor changes</td>
<td>no auto-summary</td>
</tr>
<tr>
<td>neighbor 10.0.0.3 activate</td>
<td>! address-family vpn4</td>
</tr>
<tr>
<td>neighbor 10.0.0.3 send-community extended</td>
<td>neighbor 10.0.0.3 activate</td>
</tr>
<tr>
<td>bgp scan-time import 5</td>
<td>neighbor 10.0.0.3 activate</td>
</tr>
<tr>
<td>exit-address-family</td>
<td>neighbor 10.0.0.3 send-community extended</td>
</tr>
<tr>
<td>! address-family ipv4 vrf vpn1</td>
<td>redistribute connected</td>
</tr>
<tr>
<td>redistribute rip</td>
<td>redistribute rip</td>
</tr>
<tr>
<td>no auto-summary</td>
<td>no auto-summary</td>
</tr>
<tr>
<td>no synchronization</td>
<td>no synchronization</td>
</tr>
<tr>
<td>exit-address-family</td>
<td>exit-address-family</td>
</tr>
</tbody>
</table>
### Example: Configuring an MPLS Virtual Private Network Using Static Routes

<table>
<thead>
<tr>
<th>PE Configuration</th>
<th>CE Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip vrf vpn1</td>
<td>ip cef</td>
</tr>
<tr>
<td>rd 100:1</td>
<td></td>
</tr>
<tr>
<td>route-target export 100:1</td>
<td>interface Loopback0</td>
</tr>
<tr>
<td>route-target import 100:1</td>
<td>ip address 10.0.0.9 255.255.255.255</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ip cef</td>
<td></td>
</tr>
<tr>
<td>mpls ldp router-id Loopback0 force</td>
<td></td>
</tr>
<tr>
<td>mpls label protocol ldp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>interface Loopback0</td>
<td></td>
</tr>
<tr>
<td>ip address 10.0.0.1 255.255.255.255</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ip vrf forwarding vpn1</td>
<td></td>
</tr>
<tr>
<td>ip address 192.0.2.3 255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>no cdp enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ip address 192.168.0.1 255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>mpls label protocol ldp</td>
<td></td>
</tr>
<tr>
<td>mpls ip</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>router ospf 100</td>
<td></td>
</tr>
<tr>
<td>network 10.0.0.0 0.0.0.0 area 100</td>
<td></td>
</tr>
<tr>
<td>network 192.168.0.0 255.255.255.0 area 100</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>router bgp 100</td>
<td></td>
</tr>
<tr>
<td>no synchronization</td>
<td></td>
</tr>
<tr>
<td>bgp log-neighbor changes</td>
<td></td>
</tr>
<tr>
<td>neighbor 10.0.0.3 remote-as 100</td>
<td></td>
</tr>
<tr>
<td>neighbor 10.0.0.3 update-source Loopback0</td>
<td></td>
</tr>
<tr>
<td>no auto-summary</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>address-family vpnv4</td>
<td></td>
</tr>
<tr>
<td>neighbor 10.0.0.3 activate</td>
<td></td>
</tr>
<tr>
<td>neighbor 10.0.0.3 send-community extended</td>
<td></td>
</tr>
<tr>
<td>bgp scan-time import 5</td>
<td></td>
</tr>
<tr>
<td>exit-address-family</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>address-family ipv4 vrf vpn1</td>
<td></td>
</tr>
<tr>
<td>redistribute connected</td>
<td></td>
</tr>
<tr>
<td>redistribute static</td>
<td></td>
</tr>
<tr>
<td>no auto-summary</td>
<td></td>
</tr>
<tr>
<td>no synchronization</td>
<td></td>
</tr>
<tr>
<td>exit-address-family</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ip route vrf vpn1 10.0.0.9 255.255.255.255</td>
<td></td>
</tr>
<tr>
<td>192.0.2.2</td>
<td></td>
</tr>
<tr>
<td>ip route vrf vpn1 192.0.2.0 255.255.0.0</td>
<td></td>
</tr>
<tr>
<td>192.0.2.2</td>
<td></td>
</tr>
</tbody>
</table>
Additional References

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Command List, All Releases</td>
</tr>
<tr>
<td>For complete syntax and usage information for the commands used in this chapter.</td>
<td>See the MPLS Commands section of the Command Reference (Catalyst 9300 Series Switches)</td>
</tr>
<tr>
<td>Configuring Cisco Express Forwarding</td>
<td>“Configuring Basic Cisco Express Forwarding” module in the Cisco Express Forwarding Configuration Guide</td>
</tr>
</tbody>
</table>

Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>

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 14: Feature Information for MPLS Virtual Private Networks

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.5.1a</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>
CHAPTER 9

MPLS QoS: Classifying and Marking EXP

• Classifying and Marking MPLS EXP, on page 87

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.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `class-map [match-all | match-any] class-map-name`
4. `match mpls experimental topmost mpls-exp-value`
5. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Switch&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> class-map `[match-all</td>
<td>match-any] class-map-name`</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter the class map name.</td>
</tr>
<tr>
<td>Switch(config)# class-map exp3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>match mpls experimental topmost mpls-exp-value</code></td>
<td>Specifies the match criteria.</td>
</tr>
<tr>
<td>Example:</td>
<td>The <code>match mpls experimental topmost</code> command classifies traffic on the basis of the EXP value in the topmost label header.</td>
</tr>
<tr>
<td>Switch(config-cmap)# match mpls experimental topmost 3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>end</code></td>
<td>(Optional) Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Switch(config-cmap)# end</td>
<td></td>
</tr>
</tbody>
</table>

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.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `policy-map policy-map-name`
4. `class class-map-name`
5. `set mpls experimental imposition mpls-exp-value`
6. `end`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: <strong>Switch&gt; enable</strong></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: <strong>Switch# configure terminal</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><code>policy-map policy-map-name</code></td>
<td>Specifies the name of the policy map to be created and</td>
</tr>
<tr>
<td>Example: <strong>Switch(config)# policy-map mark-up-exp-2</strong></td>
<td>enters policy-map configuration mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter the policy map name.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><code>class class-map-name</code></td>
<td>Creates a class map to be used for matching traffic to a</td>
</tr>
<tr>
<td>Example: <strong>Switch(config-pmap)# class prec012</strong></td>
<td>specified class, and enters class-map configuration mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter the class map name.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><code>set mpls experimental imposition mpls-exp-value</code></td>
<td>Sets the value of the MPLS EXP field on all imposed label entries.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
### Marking MPLS EXP on Label Switched Packets

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

**Before you begin**

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.

#### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `policy-map policy-map-name`
4. `class class-map-name`
5. `set mpls experimental topmost mpls-exp-value`
6. `end`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | `enable` | Enables privileged EXEC mode.  
**Example:**  
Switch> enable | |
| **Step 2** | `configure terminal` | Enters global configuration mode.  
**Example:**  
Switch# configure terminal | |
| **Step 3** | `policy-map policy-map-name` | Specifies the name of the policy map to be created and enters policy-map configuration mode.  
**Example:**  
Switch(config)# policy-map mark-up-exp-2 |  
**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.
### Command or Action

**Step 4**  
`class class-map-name`  
Example:  
`Switch(config-pmap)# class-map exp012`

- **Purpose**: 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`

- **Purpose**: Sets the MPLS EXP field value in the topmost label on the output interface.

**Step 6**  
`end`  
Example:  
`Switch(config-pmap-c)# end`

- **Purpose**: (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.

### SUMMARY STEPS

1. `enable`  
2. `configure terminal`  
3. `policy-map policy-map-name`  
4. `class class-map-name`  
5. `police cir bps bc pir bps bc`  
6. `conform-action transmit`  
7. `exceed-action set-mpls-exp-topmost-transmit dscp table dscp-table-value`  
8. `violate-action drop`  
9. `end`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: <code>Switch&gt; enable</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
</tr>
</tbody>
</table>
| 2    | **configure terminal**  
*Example:*  
Switch# configure terminal | Enters global configuration mode. |
| 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. |
| 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 name. |
| 5    | **police cir bps bc pir bps be**  
*Example:*  
Switch(config-pmap-c)# police cir 1000000 pir 200000 | Defines a policer for classified traffic and enters policy-map class police configuration mode. |
| 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. |
| 7    | **exceed-action set-mpls-exp-topmost-transmit dscp table dscp-table-value**  
*Example:*  
Switch(config-pmap-c-police)# exceed-action set-mpls-exp-topmost-transmit dscp table dscp2exp | Defines the action to take on packets that exceed the values specified by the policer. |
| 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. |
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 9</strong> end</td>
<td>(Optional) Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Switch(config-pmap-c-police)# end</td>
<td></td>
</tr>
</tbody>
</table>

### 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-c)# 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 1000000
Switch(config-pmap-c)# 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.

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>For IP imposition marking, the IP precedence value is copied to the MPLS EXP value by default.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>The <code>set mpls experimental imposition</code> command works only on packets that have new or additional MPLS labels added to them.</td>
</tr>
</tbody>
</table>

### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `policy-map policy-map-name`
4. `class class-map-name`
5. `set mpls experimental imposition mpls-exp-value`
6. `end`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **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:** | Sets the value of the MPLS EXP field on all imposed label entries. |
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:

```plaintext
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-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:

```plaintext
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.

```plaintext
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-c)# exit
Switch(config-pmap-c)# exit
```
Switch(config)# interface GigabitEthernet 0/0/1
Switch(config-if)# service-policy input ip2tag

## Additional References

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Commands List, All Releases</td>
</tr>
<tr>
<td>QoS commands</td>
<td>Cisco IOS Quality of Service Solutions Command Reference</td>
</tr>
</tbody>
</table>

### Standards and RFCs

<table>
<thead>
<tr>
<th>Standard/RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No new or modified standards are supported, and support for existing standards has not been modified.</td>
</tr>
</tbody>
</table>

### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

## 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](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

### Table 15: Feature Information for QoS MPLS EXP

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.5.1a</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>
CHAPTER 10

Configuring Virtual Private LAN Service (VPLS) and VPLS BGP-Based Autodiscovery

- Finding Feature Information, on page 99
- Configuring VPLS, on page 99
- Configuring VPLS BGP-based Autodiscovery, on page 114

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.

Configuring VPLS

Information About VPLS

VPLS Overview

VPLS (Virtual Private LAN Service) enables enterprises to link together their Ethernet-based LANs from multiple sites via the infrastructure provided by their service provider. From the enterprise perspective, the service provider's public network looks like one giant Ethernet LAN. For the service provider, VPLS provides an opportunity to deploy another revenue-generating service on top of their existing network without major capital expenditures. Operators can extend the operational life of equipment in their network.

Virtual Private LAN Services (VPLS) uses the provider core to join multiple attachment circuits together to simulate a virtual bridge that connects the multiple attachment circuits together. From a customer point of view, there is no topology for VPLS. All of the CE devices appear to connect to a logical bridge emulated by the provider core.
Full-Mesh Configuration

The full-mesh configuration requires a full mesh of tunnel label switched paths (LSPs) between all the PEs that participate in the VPLS. With full-mesh, signaling overhead and packet replication requirements for each provisioned VC on a PE can be high.

You set up a VPLS by first creating a virtual forwarding instance (VFI) on each participating PE router. The VFI specifies the VPN ID of a VPLS domain, the addresses of other PE devices in the domain, and the type of tunnel signaling and encapsulation mechanism for each peer PE router.

The set of VFIs formed by the interconnection of the emulated VCs is called a VPLS instance; it is the VPLS instance that forms the logic bridge over a packet switched network. The VPLS instance is assigned a unique VPN ID.

The PE devices use the VFI to establish a full-mesh LSP of emulated VCs to all the other PE devices in the VPLS instance. PE devices obtain the membership of a VPLS instance through static configuration using the Cisco IOS CLI.

The full-mesh configuration allows the PE router to maintain a single broadcast domain. Thus, when the PE router receives a broadcast, multicast, or unknown unicast packet on an attachment circuit, it sends the packet out on all other attachment circuits and emulated circuits to all other CE devices participating in that VPLS instance. The CE devices see the VPLS instance as an emulated LAN.

To avoid the problem of a packet looping in the provider core, the PE devices enforce a "split-horizon" principle for the emulated VCs. That means if a packet is received on an emulated VC, it is not forwarded on any other emulated VC.

After the VFI has been defined, it needs to be bound to an attachment circuit to the CE device.

The packet forwarding decision is made by looking up the Layer 2 virtual forwarding instance (VFI) of a particular VPLS domain.

A VPLS instance on a particular PE router receives Ethernet frames that enter on specific physical or logical ports and populates a MAC table similarly to how an Ethernet switch works. The PE router can use the MAC address to switch those frames into the appropriate LSP for delivery to the another PE router at a remote site.

If the MAC address is not in the MAC address table, the PE router replicates the Ethernet frame and floods it to all logical ports associated with that VPLS instance, except the ingress port where it just entered. The PE router updates the MAC table as it receives packets on specific ports and removes addresses not used for specific periods.
**VPLS BGP Based Autodiscovery**

VPLS Autodiscovery enables each Virtual Private LAN Service (VPLS) provider edge (PE) device to discover other PE devices that are part of the same VPLS domain. VPLS Autodiscovery also tracks PE devices when they are added to or removed from a VPLS domain. As a result, with VPLS Autodiscovery enabled, you no longer need to manually configure a VPLS domain and maintain the configuration when a PE device is added or deleted. VPLS Autodiscovery uses the Border Gateway Protocol (BGP) to discover VPLS members and set up and tear down pseudowires in a VPLS domain.

BGP uses the Layer 2 VPN (L2VPN) Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 virtual forwarding instance (VFI) is configured. The prefix and path information is stored in the L2VPN database, which allows BGP to make decisions about the best path. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, this endpoint information is used to configure a pseudowire mesh to support L2VPN-based services.

The BGP autodiscovery mechanism facilitates the configuration of L2VPN services, which are an integral part of the VPLS feature. VPLS enables flexibility in deploying services by connecting geographically dispersed sites as a large LAN over high-speed Ethernet in a robust and scalable IP Multiprotocol Label Switching (MPLS) network.

### Scale Numbers

**Table 16: VPLS Scale**

<table>
<thead>
<tr>
<th>Platform</th>
<th>Scale numbers as per SDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>3650</td>
<td>32 VFI, 32 VLAN, 8 neighbour per VFI, 256 VC/PWs</td>
</tr>
<tr>
<td>3850</td>
<td>32 VFI, 32 VLAN, 8 neighbour per VFI, 256 VC/PWs</td>
</tr>
<tr>
<td>9300</td>
<td>128 VFI, 128 VLAN, 32 neighbour per VFI, 1024 VC/PWs</td>
</tr>
<tr>
<td>9500</td>
<td>128 VFI, 128 VLAN, 32 neighbour per VFI, 4096 VC/PWs</td>
</tr>
</tbody>
</table>

### Configuration Examples for VPLS

*Figure 8: VPLS Topology*
### PE1 Configuration

<table>
<thead>
<tr>
<th>Pseudowire-class</th>
<th>Encapsulation mpls</th>
<th>12 Vfi 2129 Manual</th>
<th>Vpn id 2129</th>
<th>Neighbor 44.254.44.44 Pw-class vpls2129</th>
<th>Neighbor 188.98.89.98 Pw-class vpls2129</th>
</tr>
</thead>
<tbody>
<tr>
<td>vpls2129</td>
<td>mpls</td>
<td>manual</td>
<td>2129</td>
<td>44.254.44.44</td>
<td>188.98.89.98</td>
</tr>
<tr>
<td>! Interface TenGigabitEthernet1/0/24 Switchport trunk allowed vlan 2129 Switchport mode trunk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>! Interface Vlans2129 No ip address Xconnect vfi 2129</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### PE2 Configuration

<table>
<thead>
<tr>
<th>Pseudowire-class</th>
<th>Encapsulation mpls</th>
<th>12 Vfi 2129 Manual</th>
<th>Vpn id 2129</th>
<th>Neighbor 1.1.1.72 Pw-class vpls2129</th>
<th>Neighbor 188.98.89.98 Pw-class vpls2129</th>
</tr>
</thead>
<tbody>
<tr>
<td>vpls2129</td>
<td>mpls</td>
<td>manual</td>
<td>2129</td>
<td>1.1.1.72</td>
<td>188.98.89.98</td>
</tr>
<tr>
<td>! Interface TenGigabitEthernet1/0/47 Switchport trunk allowed vlan 2129 Switchport mode trunk End</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>! Interface Vlans2129 No ip address Xconnect vfi 2129</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The `show mpls 12transport vc` command provides information about the virtual circuits.

Local interface: VFI 2129 VFI up

- Interworking type is Ethernet
- Destination address: 44.254.44.44, VC ID: 2129, VC status: up
- Output interface: Gi1/0/9, imposed label stack (18 17)
- Preferred path: not configured
- Default path: active
- Next hop: 177.77.177.2
- Create time: 19:09:33, last status change time: 09:24:14
- Last label FSM state change time: 09:24:14
- Signaling protocol: LDP, peer 44.254.44.44:0 up
  - Targeted Hello: 1.1.1.72 (LDP Id) -> 44.254.44.44, LDP is UP
- Graceful restart: configured and enabled
- Non stop routing: not configured and not enabled
- Status TLV support (local/remote): enabled/supported
- LDP route watch: enabled
- Label/status state machine: established, LruRru
- Last local dataplane status rcvd: No fault
- Last BFD dataplane status rcvd: Not sent
- Last BFD peer monitor status rcvd: No fault
- Last local AC circuit status rcvd: No fault
- Last local AC circuit status sent: No fault
- Last local PW i/f circ status rcvd: No fault
- Last local LDP TLV status sent: No fault
- Last remote LDP TLV status rcvd: No fault
- Last remote LDP ADJ status rcvd: No fault
- MPLS VC labels: local 512, remote 17
  - Group ID: local n/a, remote 0
  - MTU: local 1500, remote 1500
- Remote interface description:
  - Sequencing: receive disabled, send disabled
  - Control Word: Off
SSO Descriptor: 44.254.44.44/2129, local label: 512
Dataplane:
  SSM segment/switch IDs: 20498/20492 (used), PWID: 2
VC statistics:
  transit packet totals: receive 0, send 0
  transit byte totals: receive 0, send 0
  transit packet drops: receive 0, seq error 0, send 0

The `show l2vpn atm vc` shows that ATM over MPLS is configured on a VC.

 pseudowire100005 is up, VC status is up PW type: Ethernet
 Create time: 19:25:56, last status change time: 09:40:37
 Last label FSM state change time: 09:40:37
 Destination address: 44.254.44.44 VC ID: 2129
 Output interface: Gi1/0/9, imposed label stack {18 17}
 Preferred path: not configured
 Default path: active
 Next hop: 177.77.177.2
 Member of vfi service 2129
 Bridge-Domain id: 2129
 Service id: 0x32000003
 Signaling protocol: LDP, peer 44.254.44.44:0 up
 Targeted Hello: 1.1.1.72(LDP Id) -> 44.254.44.44, LDP is UP
 Graceful restart: configured and enabled
 Non stop routing: not configured and not enabled
 PWid FEC (128), VC ID: 2129
 Status TLV support (local/remote) : enabled/supported
 LDP route watch : enabled
 Label/status state machine : established, LruRru
 Local dataplane state status received : No fault
 BFD dataplane status received : Not sent
 BFD peer monitor status received : No fault
 Status received from access circuit : No fault
 Status sent to access circuit : No fault
 Status received from pseudowire i/f : No fault
 Status sent to network peer : No fault
 Status received from network peer : No fault
 Adjacency status of remote peer : No fault
 Sequencing: receive disabled, send disabled
 Bindings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Local</th>
<th>Remote</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>512</td>
<td>17</td>
</tr>
<tr>
<td>Group ID</td>
<td>n/a</td>
<td>0</td>
</tr>
<tr>
<td>Interface</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTU</td>
<td>1500</td>
<td>1500</td>
</tr>
<tr>
<td>Control word</td>
<td>off</td>
<td>off</td>
</tr>
<tr>
<td>PW type</td>
<td>Ethernet</td>
<td>Ethernet</td>
</tr>
<tr>
<td>VCCV CV type</td>
<td>0x02</td>
<td>0x02</td>
</tr>
</tbody>
</table>
Restrictions for VPLS

- This feature is not supported on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.
- Protocol-based CLI Method (interface pseudowire configuration) is not supported. Only VFI and Xconnect mode are supported.
- Flow-Aware Transport Pseudowire (FAT PW) is not supported.
- IGMP Snooping is not Supported. Multicast traffic floods with IGMP Snooping disabled.
- L2 Protocol Tunneling is not supported.
- Integrated Routing and Bridging (IRB) not supported.
- Virtual Circuit Connectivity Verification (VCCV) ping with explicit null is not supported.
- Pseudowire Redundancy with VPLS not supported.
- The switch is supported only as spoke in H-VPLS but not as hub.
- MAC Address Withdrawal is not supported.
- L2 VPN Interworking is not supported.
- VC statistics are not displayed for flood traffic in the output of show mpls l2 vc vcid detail command.
- Q-in-Q traffic is not supported.
- dot1q tunnel is not supported in the attachment circuit.

Configuring PE Layer 2 Interfaces to CEs

Configuring 802.1Q Trunks for Tagged Traffic from a CE

SUMMARY STEPS

1. enable
2. `configure terminal`
3. `interface interface-id`
4. `no ip address ip_address mask [secondary ]`
5. `switchport`
6. `switchport trunk encapsulation dot1q`
7. `switchport trunk allow vlan vlan_ID`
8. `switchport mode trunk`
9. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><code>enable</code></td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device&gt; enable</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Defines the interface to be configured as a trunk, and enters interface configuration mode.</td>
</tr>
<tr>
<td><code>interface interface-id</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config)# interface TenGigabitEthernet1/0/24</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Disables IP processing and enters interface configuration mode.</td>
</tr>
<tr>
<td><code>no ip address ip_address mask [secondary ]</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config-if)# no ip address</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Modifies the switching characteristics of the Layer 2-switched interface.</td>
</tr>
<tr>
<td><code>switchport</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config-if)# switchport</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Sets the switch port encapsulation format to 802.1Q.</td>
</tr>
<tr>
<td><code>switchport trunk encapsulation dot1q</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config-if)# switchport trunk encapsulation dot1q</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Sets the list of allowed VLANs.</td>
</tr>
<tr>
<td><code>switchport trunk allow vlan vlan_ID</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>
### Configuring 802.1Q Access Ports for Untagged Traffic from a CE

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `interface interface-id`
4. `no ip address ip_address mask [secondary ]`
5. `switchport`
6. `switchport mode access`
7. `switchport access vlan vlan_ID`
8. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>Device&gt; enable</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>Device# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>interface interface-id</code></td>
<td>Defines the interface to be configured as a trunk, and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

| Device(config)# interface TenGigabitEthernet1/0/24 |

### Purpose

Disables IP processing and enters interface configuration mode.

### Step 4

**no ip address ip_address mask [secondary]**

**Example:**

Device(config-if)# no ip address

### Step 5

**switchport**

**Example:**

Device(config-if)# switchport

### Step 6

**switchport mode access**

**Example:**

Device(config-if)# switchport mode access

### Step 7

**switchport access vlan vlan_ID**

**Example:**

Device(config-if)# switchport access vlan 2129

### Step 8

**end**

**Example:**

Device(config)# end

---

### Configuring Layer 2 VLAN Instances on a PE

Configuring the Layer 2 VLAN interface on the PE enables the Layer 2 VLAN instance on the PE router to the VLAN database to set up the mapping between the VPLS and VLANs.

#### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vlan vlan-id**
4. **interface vlan vlan-id**
5. **end**

---

**Multiprotocol Label Switching (MPLS) Configuration Guide, Cisco IOS XE Fuji 16.8.x (Catalyst 9300 Switches)**
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **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**
  vlan vlan-id
  **Example:**
  Device(config)# vlan 2129 | Configures a specific virtual LAN (VLAN). |
| **Step 4**
  interface vlan vlan-id
  **Example:**
  Device(config-vlan)# interface vlan 2129 | Configures an interface on the VLAN. |
| **Step 5**
  end
  **Example:**
  Device(config)# end | Returns to privileged EXEC mode. |

## Configuring MPLS in the PE

To configure MPLS in the PE, you must provide the required MPLS parameters.

## SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `mpls ip`
4. `mpls label protocol ldp`
5. `mpls label protocol ldp`
6. `end`
7. `mpls ldp logging neighbor-changes`
DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>enable</td>
<td>Enters privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Configures MPLS hop-by-hop forwarding.</td>
</tr>
<tr>
<td>mpls ip</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# mpls ip</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Specifies the default Label Distribution Protocol for a platform.</td>
</tr>
<tr>
<td>mpls label protocol ldp</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config-vlan)# mpls label protocol ldp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Specifies the default Label Distribution Protocol for a platform.</td>
</tr>
<tr>
<td>mpls label protocol ldp</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config-vlan)# interface vlan 2129</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>(Optional) Determines logging neighbor changes.</td>
</tr>
<tr>
<td>mpls ldp logging neighbor-changes</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# mpls ldp logging neighbor-changes</td>
<td></td>
</tr>
</tbody>
</table>

**Configuring VFI in the PE**

The virtual switch instance (VFI) specifies the VPN ID of a VPLS domain, the addresses of other PE devices in this domain, and the type of tunnel signaling and encapsulation mechanism for each peer (This is where you create the VFI and associated VCs.). Configure a VFI as follows:
### SUMMARY STEPS

1. enable
2. configure terminal
3. l2 vfi vfi-name manual
4. vpn id vpn-id
5. neighbor remote-router-id {encapsulation mpls}
6. end

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
Example:  
Device> enable |
| **Step 2** configure terminal | Enters global configuration mode.  
Example:  
Device# configure terminal |
| **Step 3** l2 vfi vfi-name manual | Enables the Layer 2 VFI manual configuration mode.  
Example:  
Device(config)# l2 vfi 2129 manual |
| **Step 4** vpn id vpn-id | Configures a VPN ID for a VPLS domain. The emulated VCs bound to this Layer 2 VRF use this VPN ID for signaling.  
**Note**  
vpn-id is the same as vlan-id.  
Example:  
Device(config-vfi)# vpn id 2129 |
| **Step 5** neighbor remote-router-id {encapsulation mpls} | Specifies the remote peering router ID and the tunnel encapsulation type or the pseudo-wire property to be used to set up the emulated VC.  
Example:  
Device(config-vfi)# neighbor remote-router-id {encapsulation mpls} |
| **Step 6** end | Returns to privileged EXEC mode.  
Example:  
Device(config)# end |
Associating the Attachment Circuit with the VFI at the PE

After defining the VFI, you must bind it to one or more attachment circuits.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `interface vlan vlan-id`
4. `no ip address`
5. `xconnect vfi vfi-name`
6. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>enable</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Device&gt; enable</code></td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td><strong>Step 2</strong></td>
</tr>
<tr>
<td></td>
<td><code>configure terminal</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Device# configure terminal</code></td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Step 3</strong></td>
</tr>
<tr>
<td></td>
<td><code>interface vlan vlan-id</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Device(config)# interface vlan 2129</code></td>
</tr>
<tr>
<td></td>
<td>Creates or accesses a dynamic switched virtual interface (SVI).</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> <code>vlan-id</code> is the same as <code>vpn-id</code>.</td>
</tr>
<tr>
<td></td>
<td><strong>Step 4</strong></td>
</tr>
<tr>
<td></td>
<td><code>no ip address</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Device(config-vlan)# no ip address</code></td>
</tr>
<tr>
<td></td>
<td>Disables IP processing. (You configure a Layer 3 interface for the VLAN if you configure an IP address.)</td>
</tr>
<tr>
<td></td>
<td><strong>Step 5</strong></td>
</tr>
<tr>
<td></td>
<td><code>xconnect vfi vfi-name</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Device(config-vlan)# xconnect vfi 2129</code></td>
</tr>
<tr>
<td></td>
<td>Specifies the Layer 2 VFI that you are binding to the VLAN port.</td>
</tr>
<tr>
<td></td>
<td><strong>Step 6</strong></td>
</tr>
<tr>
<td></td>
<td><code>end</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Device(config)# end</code></td>
</tr>
<tr>
<td></td>
<td>Returns to privileged EXEC mode.</td>
</tr>
</tbody>
</table>
## Configuration Examples for VPLS

The **show mpls 12transport vc** command provides information the virtual circuits.

**Local interface:** VFI 2129 vfi up  
**Interworking type is Ethernet**  
**Destination address:** 44.254.44.44, **VC ID:** 2129, **VC status:** up  
**Output interface:** Gi1/0/9, **imposed label stack {18 17}**  
**Preferred path:** not configured  
**Default path:** active  
**Next hop:** 177.77.177.2  
**Create time:** 19:09:33, **last status change time:** 09:24:14  
**Last label FSM state change time:** 09:24:14  
**Signaling protocol:** LDP, peer 44.254.44.44:0 up  
**Targeted Hello:** 1.1.1.72 (LDP Id) -> 44.254.44.44, LDP is UP  
**Graceful restart:** configured and enabled  
**Non stop routing:** not configured and not enabled  
**Status TLV support (local/remote) :** enabled/supported  
**LDP route watch :** enabled  
**Label/status state machine :** established, LruRru

### PE1 Configuration

- pseudowire-class vpls2129  
- encapsulation mpls  
- 12 vfi 2129 manual  
- vpn id 2129  
- neighbor 44.254.44.44 pw-class vpls2129  
- neighbor 188.98.89.98 pw-class vpls2129  

> !

- interface TenGigabitEthernet1/0/24  
- switchport trunk allowed vlan 2129  
- switchport mode trunk  

> !

- interface Vlan2129  
- no ip address  
- xconnect vfi 2129  

> !

### PE2 Configuration

- pseudowire-class vpls2129  
- encapsulation mpls  
- no control-word  
- 12 vfi 2129 manual  
- vpn id 2129  
- neighbor 1.1.1.72 pw-class vpls2129  
- neighbor 188.98.89.98 pw-class vpls2129  

> !

- interface TenGigabitEthernet1/0/47  
- switchport trunk allowed vlan 2129  
- switchport mode trunk  

> end  

- !

- interface Vlan2129  
- no ip address  
- xconnect vfi 2129  

> !
Last local dataplane status rcvd: No fault
Last BFD dataplane status rcvd: Not sent
Last BFD peer monitor status rcvd: No fault
Last local AC circuit status rcvd: No fault
Last local AC circuit status sent: No fault
Last local PW i/f circ status rcvd: No fault
Last local LDP TLV status sent: No fault
Last remote LDP TLV status rcvd: No fault
Last remote LDP ADJ status rcvd: No fault

MPLS VC labels: local 512, remote 17
Group ID: local n/a, remote 0
MTU: local 1500, remote 1500
Remote interface description:
Sequencing: receive disabled, send disabled
Control Word: Off
SSO Descriptor: 44.254.44.44/2129, local label: 512
Dataplane:
SSM segment/switch IDs: 20498/20492 (used), PWID: 2
VC statistics:
transit packet totals: receive 0, send 0
transit byte totals: receive 0, send 0
transit packet drops: receive 0, seq error 0, send 0

The show l2vpn atm vc shows that ATM over MPLS is configured on a VC.

pseudowire100005 is up, VC status is up PW type: Ethernet
Create time: 19:25:56, last status change time: 09:40:37
Last label FSM state change time: 09:40:37
Destination address: 44.254.44.44 VC ID: 2129
Output interface: Gi1/0/9, imposed label stack {18 17}
Preferred path: not configured
Default path: active
Next hop: 177.77.177.2
Member of vfi service 2129
Bridge-Domain id: 2129
Service id: 0x32000003
Signaling protocol: LDP, peer 44.254.44.44:0 up
Targeted Hello: 1.1.1.72 (LDP Id) -> 44.254.44.44, LDP is UP
Graceful restart: configured and enabled
Non stop routing: not configured and not enabled
PWid FEC (128), VC ID: 2129
Status TLV support (local/remote) : enabled/supported
LDP route watch : enabled
Label/status state machine : established, LruRru
Local dataplane status received : No fault
BFD dataplane status received : Not sent
BFD peer monitor status received : No fault
Status received from access circuit : No fault
Status sent to access circuit : No fault
Status received from pseudowire i/f : No fault
Status sent to network peer : No fault
Information About VPLS BGP-Based Autodiscovery

VPLS BGP Based Autodiscovery

VPLS Autodiscovery enables each Virtual Private LAN Service (VPLS) provider edge (PE) device to discover other PE devices that are part of the same VPLS domain. VPLS Autodiscovery also tracks PE devices when they are added to or removed from a VPLS domain. As a result, with VPLS Autodiscovery enabled, you no longer need to manually configure a VPLS domain and maintain the configuration when a PE device is added or deleted. VPLS Autodiscovery uses the Border Gateway Protocol (BGP) to discover VPLS members and set up and tear down pseudowires in a VPLS domain.

BGP uses the Layer 2 VPN (L2VPN) Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 virtual forwarding instance (VFI) is configured. The prefix and path information is stored in the L2VPN database, which allows BGP to make decisions about the best path. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, this endpoint information is used to configure a pseudowire mesh to support L2VPN-based services.
The BGP autodiscovery mechanism facilitates the configuration of L2VPN services, which are an integral part of the VPLS feature. VPLS enables flexibility in deploying services by connecting geographically dispersed sites as a large LAN over high-speed Ethernet in a robust and scalable IP Multiprotocol Label Switching (MPLS) network.

Scale Numbers

Table 17: BGP - AD Scale

<table>
<thead>
<tr>
<th>Platform</th>
<th>Scale numbers as per SDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>3650</td>
<td>32 VFI, 32 VLAN, 8 neighbour per VFI, 256 VC/PWs</td>
</tr>
<tr>
<td>3850</td>
<td>32 VFI, 32 VLAN, 8 neighbour per VFI, 256 VC/PWs</td>
</tr>
<tr>
<td>9300</td>
<td>128 VFI, 128 VLAN, 32 neighbour per VFI, 1024 VC/PWs</td>
</tr>
<tr>
<td>9500</td>
<td>128 VFI, 128 VLAN, 32 neighbour per VFI, 4096 VC/PWs</td>
</tr>
</tbody>
</table>

Enabling VPLS BGP-based Autodiscovery

Perform this task to enable Virtual Private LAN Service (VPLS) PE devices to discover other PE devices that are part of the same VPLS domain.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `l2 vfi vfi-name  autodiscovery`
4. `vpn id vpn-id`
5. `end`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device&gt; enable</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device# configure terminal</code></td>
<td></td>
</tr>
</tbody>
</table>
### Configuring BGP to Enable VPLS Autodiscovery

The Border Gateway Protocol (BGP) Layer 2 VPN (L2VPN) address family supports a separate L2VPN Routing Information Base (RIB) that contains endpoint provisioning information for Virtual Private LAN Service (VPLS) Autodiscovery. BGP learns the endpoint provisioning information from the L2VPN database, which is updated each time a Layer 2 virtual forwarding instance (VFI) is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to configure a pseudowire mesh to support L2VPN-based services.

#### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `router bgp autonomous-system-number`
4. `no bgp default ipv4-unicast`
5. `bgp log-neighbor-changes`
6. `neighbor remote-as { ip-address | peer-group-name } remote-as autonomous-system-number`
7. `neighbor { ip-address | peer-group-name } update-source interface-type interface-number`
8. Repeat Steps 6 and 7 to configure other BGP neighbors.
9. `address-family l2vpn vpls number`
10. `neighbor { ip-address | peer-group-name } activate`
11. `neighbor { ip-address | peer-group-name } send-community { both | standard | extended }`
12. Repeat Steps 10 and 11 to activate other BGP neighbors under an L2VPN address family.
13. `exit-address-family`
14. `end`

### Command or Action | Purpose
---|---
**Step 3** | Enables VPLS Autodiscovery on a PE device and enters L2 VFI configuration mode.
| `l2 vfi vfi-name autodiscovery` |  |
| **Example:** |  |
| ` Device(config)# l2 vfi 2128 autodiscovery` |  |

**Step 4** | Configures a VPN ID for the VPLS domain.
| `vpn id vpn-id` |  |
| **Example:** |  |
| ` Device(config-vfi)# vpn id 2128` |  |

**Step 5** | Returns to privileged EXEC mode.
| `end` |  |
| **Example:** |  |
| ` Device(config)# end` |  |
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
<th>Example</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode. | **Example:**  
Device> enable  
- Enter your password if prompted. |
| **Step 2** configure terminal | Enters global configuration mode. | **Example:**  
Device# configure terminal |
| **Step 3** router bgp autonomous-system-number | Enters router configuration mode for the specified routing process. | **Example:**  
Device(config)# router bgp 1000 |
| **Step 4** no bgp default ipv4-unicast | Disables the IPv4 unicast address family for the BGP routing process. | **Example:**  
Device(config-router)# no bgp default ipv4-unicast  
**Note:** Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured using the neighbor remote-as router configuration command unless you configure the no bgp default ipv4-unicast router configuration command before configuring the neighbor remote-as command. Existing neighbor configurations are not affected. |
| **Step 5** bgp log-neighbor-changes | Enables logging of BGP neighbor resets. | **Example:**  
Device(config-router)# bgp log-neighbor-changes |
| **Step 6** neighbor remote-as {ip-address | peer-group-name} remote-as autonomous-system-number | Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.  
- If the autonomous-system-number argument matches the autonomous system number specified in the router bgp command, the neighbor is an internal neighbor.  
- If the autonomous-system-number argument does not match the autonomous system number specified in the router bgp command, the neighbor is an external neighbor. | **Example:**  
Device(config-router)# neighbor 44.254.44.44 remote-as 1000 |
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td><strong>neighbor</strong> { ip-address</td>
<td>peer-group-name } <strong>update-source</strong> interface-type interface-number</td>
</tr>
</tbody>
</table>
|      | **Example:**  
|      | Device(config-router)# neighbor 44.254.44.44 update-source Loopback300 | |
| 8    | Repeat Steps 6 and 7 to configure other BGP neighbors. | Exits interface configuration mode. |
| 9    | **address-family l2vpn vpls** number | Specifies the L2VPN address family and enters address family configuration mode. |
|      | **Example:**  
|      | Device(config-router)# address-family l2vpn vpls | The optional **vpls** keyword specifies that the VPLS endpoint provisioning information is to be distributed to BGP peers. |
| 10   | **neighbor** { ip-address | peer-group-name } **activate** | Enables the exchange of information with a BGP neighbor. |
|      | **Example:**  
|      | Device(config-router-af)# neighbor 44.254.44.44 activate | |
| 11   | **neighbor** { ip-address | peer-group-name } **send-community** { both | standard | extended } | Specifies that a communities attribute should be sent to a BGP neighbor. |
|      | **Example:**  
|      | Device(config-router-af)# neighbor 44.254.44.44 send-community both | |
| 12   | Repeat Steps 10 and 11 to activate other BGP neighbors under an L2VPN address family. | |
| 13   | **exit-address-family** | Exits address family configuration mode and returns to router configuration mode. |
|      | **Example:**  
|      | Device(config-router-af)# exit-address-family | |
| 14   | **end** | Exits router configuration mode and returns to privileged EXEC mode. |
|      | **Example:**  
|      | Device(config-router-af)# end | |
Configuration Examples for VPLS BGP-AD

PE Configuration

```
router bgp 1000
bgp log-neighbor-changes
bgp graceful-restart
neighbor 44.254.44.44 remote-as 1000
neighbor 44.254.44.44 update-source Loopback300
! address-family 12vpn vpls
neighbor 44.254.44.44 activate
neighbor 44.254.44.44 send-community both
exit-address-family
! l2 vfi 2128 autodiscovery
vpn id 2128
interface Vlan2128
no ip address
xconnect vfi 2128
!
```

The following is a sample output of `show platform software fed sw 1 matm macTable vlan 2000` command:

```
VLAN  MAC         Type        Seq#   macHandle    siHandle
      diHandle  *a_time *e_time  ports
2000  2852.6134.05c8 0x8002  0     0xffbb31c8    0xffbb9ef938
       0x5154       0      0           Vlan2000
2000  0000.0078.9012 0x1     32627 0xffbb665ec8    0xffbb60b198
       0xffbb653f98 300    278448   Port-channel11
2000  2852.6134.0000 0x1     32651 0xffba15e1a8    0xff454c2328
       0xffbb653f98 300    63       Port-channel11
2000  0000.0012.3456 0x2000001 32655 0xffba15c508    0xff44f9ec98
       0x0         300    1           Vlan2000:33.33.33.33
Total Mac number of addresses:: 4
*a_time=aging_time(secs)  *e_time=total_elapsed_time(secs)
```

Type:

```
MAT_DYNAMIC_ADDR 0x1  MAT_STATIC_ADDR 0x2
MAT_CPU_ADDR 0x4  MAT_DISCARD_ADDR 0x8
MAT_ALL_VLANS 0x10  MAT_NO_FORWARD 0x20
MAT_IPMULT_ADDR 0x40  MAT_RESYNC 0x80
MAT_DO NOT AGE 0x100  MAT_SECURE_ADDR 0x200
MAT_NO PORT 0x400  MAT_DROP_ADDR 0x800
MAT_DUP_ADDR 0x1000  MAT_NULL_DESTINATION 0x2000
MAT_DOT1X_ADDR 0x4000  MAT_ROUTER_ADDR 0x8000
MAT_WIRELESS_ADDR 0x10000  MAT_SECURE_CFG_ADDR 0x20000
MAT_OPQ_DATA_PRESENT 0x40000  MAT_WIRED_TUNNEL_ADDR 0x80000
MAT_DLR_ADDR 0x100000  MAT_MRP_ADDR 0x200000
MAT_MSRP_ADDR 0x400000  MAT_LISP_LOCAL_ADDR 0x800000
MAT_LISP_REMOTE_ADDR 0x1000000  MAT_VPLS_ADDR 0x2000000
```

The following is a sample output of `show bgp 12vpn vpls all` command:
BGP table version is 6, local router ID is 222.5.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,
t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000:2128:1.1.1.72/96</td>
<td>0.0.0.0</td>
<td>32768</td>
<td>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000:2128:44.254.44.44/96</td>
<td>44.254.44.44</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>?</td>
</tr>
</tbody>
</table>
Configuring MPLS VPN Route Target Rewrite

- Finding Feature Information, on page 121
- Prerequisites for MPLS VPN Route Target Rewrite, on page 121
- Restrictions for MPLS VPN Route Target Rewrite, on page 121
- Information About MPLS VPN Route Target Rewrite, on page 122
- How to Configure MPLS VPN Route Target Rewrite, on page 123
- Configuration Examples for MPLS VPN Route Target Rewrite, on page 129

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 VPN Route Target Rewrite

- You should know how to configure Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs).
- You need to identify the RT replacement policy and target device for the autonomous system (AS).

Restrictions for MPLS VPN Route Target Rewrite

Route Target Rewrite can only be implemented in a single AS topology.
Information About MPLS VPN Route Target Rewrite

Route Target Replacement Policy

Routing policies for a peer include all configurations that may impact inbound or outbound routing table updates. The MPLS VPN Route Target Rewrite feature can influence routing table updates by allowing the replacement of route targets on inbound and outbound Border Gateway Protocol (BGP) updates. Route targets are carried as extended community attributes in BGP Virtual Private Network IP Version 4 (VPNv4) updates. Route target extended community attributes are used to identify a set of sites and VPN routing and forwarding (VRF) instances that can receive routes with a configured route target.

You can configure the MPLS VPN Route Target Rewrite feature on provider edge (PE) devices.

The figure below shows an example of route target replacement on PE devices in an Multiprotocol Label Switching (MPLS) VPN single autonomous system topology. This example includes the following configurations:

- PE1 is configured to import and export RT 65000:1 for VRF Customer A and to rewrite all inbound VPNv4 prefixes with RT 65000:1 to RT 65000:2.
- PE2 is configured to import and export RT 65000:2 for VRF Customer B and to rewrite all inbound VPNv4 prefixes with RT 65000:2 to RT 65000:1.

Figure 10: Route Target Replacement on Provide Edge(PE) devices in a single MPLS VPN Autonomous System Topology

Route Maps and Route Target Replacement

The MPLS VPN Route Target Rewrite feature extends the Border Gateway Protocol (BGP) inbound/outbound route map functionality to enable route target replacement. The set extcomm-list delete command entered in route-map configuration mode allows the deletion of a route target extended community attribute based on an extended community list.
### How to Configure MPLS VPN Route Target Rewrite

#### Configuring a Route Target Replacement Policy

Perform this task to configure a route target (RT) replacement policy for your internetwork.

If you configure a provider edge (PE) device to rewrite RT $x$ to RT $y$ and the PE has a virtual routing and forwarding (VRF) instance that imports RT $x$, you need to configure the VRF to import RT $y$ in addition to RT $x$.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. `ip extcommunity-list {standard-list-number | expanded-list-number} {permit | deny} [regular-expression] [rt | soo extended-community-value]`
4. `route-map map-name {permit | deny} [sequence-number]`
5. `match extcommunity {standard-list-number | expanded-list-number}`
6. `set extcomm-list extended-community-list-number delete`
7. `set extcommunity {rt extended-community-value [additive] | soo extended-community-value}`
8. `end`
9. `show route-map map-name`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>enable</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Device&gt; enable</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>configure terminal</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Device# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>**ip extcommunity-list {standard-list-number</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>• The permit keyword permits access for a matching condition.</td>
<td></td>
</tr>
<tr>
<td>• The deny keyword denies access for a matching condition.</td>
<td></td>
</tr>
<tr>
<td>• The regular-expression argument specifies an input string pattern to match against. When you use an expanded extended community list to match route targets, include the pattern RT: in the regular expression.</td>
<td></td>
</tr>
<tr>
<td>• The rt keyword specifies the route target extended community attribute. The rt keyword can be configured only with standard extended community lists and not expanded community lists.</td>
<td></td>
</tr>
<tr>
<td>• The soo keyword specifies the site of origin (SOO) extended community attribute. The soo keyword can be configured only with standard extended community lists and not expanded community lists.</td>
<td></td>
</tr>
<tr>
<td>• The extended-community-value argument specifies the route target or site of origin. The value can be one of the following combinations:</td>
<td></td>
</tr>
<tr>
<td>• autonomous-system-number:network-number</td>
<td></td>
</tr>
<tr>
<td>• ip-address:network-number</td>
<td></td>
</tr>
</tbody>
</table>

The colon is used to separate the autonomous system number and network number or IP address and network number.

**Step 4**

**route-map map-name [permit | deny] [sequence-number]**

**Example:**

Device(config)# route-map rtrewrite permit 10

Defines the conditions for redistributing routes from one routing protocol into another or enables policy routing and enables route-map configuration mode.

• The map-name argument defines a meaningful name for the route map. The redistribute router configuration command uses this name to reference this route map. Multiple route maps can share the same map name.

• If the match criteria are met for this route map, and the permit keyword is specified, the route is redistributed as controlled by the set actions. In the case of policy routing, the packet is policy routed.

If the match criteria are not met, and the permit keyword is specified, the next route map with the same map tag is tested. If a route passes none of the match criteria for the set of route maps sharing the same name, it is not redistributed by that set.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **The permit** keyword is the default.  
• If the match criteria are met for the route map and the **deny** keyword is specified, the route is not redistributed. In the case of policy routing, the packet is not policy routed, and no further route maps sharing the same map tag name will be examined. If the packet is not policy routed, the normal forwarding algorithm is used.  
• The **sequence-number** argument is a number that indicates the position a new route map will have in the list of route maps already configured with the same name. If given with the **no** form of this command, the position of the route map should be deleted. |
| **Step 5**  
**match extcommunity** `{standard-list-number | expanded-list-number}`  
**Example:**  
Device(config-route-map)# match extcommunity 1  
**Example:**  
Device(config-route-map)# match extcommunity 101 |
| **Matches the Border Gateway Protocol (BGP) extended community list attributes.**  
• The **standard-list-number** argument is a number from 1 to 99 that identifies one or more permit or deny groups of extended community attributes.  
• The **expanded-list-number** argument is a number from 100 to 500 that identifies one or more permit or deny groups of extended community attributes. |
| **Step 6**  
**set extcomm-list** `extended-community-list-number delete`  
**Example:**  
Device(config-route-map)# set extcomm-list 1 delete |
| **Removes a route target from an extended community attribute of an inbound or outbound BGP Virtual Private Network Version 4 (VPNv4) update.**  
• The **extended-community-list-number** argument specifies the extended community list number. |
| **Step 7**  
**set extcommunity** `{rt extended-community-value [additive] | soo extended-community-value}`  
**Example:**  
Device(config-route-map)# set extcommunity rt 65000:1 additive |
| **Sets BGP extended community attributes.**  
• The **rt** keyword specifies the route target extended community attribute.  
• The **soo** keyword specifies the site of origin extended community attribute.  
• The **extended-community-value** argument specifies the value to be set. The value can be one of the following combinations:  
  • autonomous-system-number : network-number  
  • ip-address : network-number  
The colon is used to separate the autonomous system number and network number or IP address and network number. |
### Applying the Route Target Replacement Policy

Perform the following tasks to apply the route target replacement policy to your network:

#### Associating Route Maps with Specific BGP Neighbors

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `router bgp as-number`
4. `neighbor {ip-address | peer-group-name} remote-as as-number`
5. `address-family vpnv4 [unicast]`
6. `neighbor {ip-address | peer-group-name} activate`
7. `neighbor {ip-address | peer-group-name} send-community [both | extended | standard]`
8. `neighbor {ip-address | peer-group-name} route-map map-name {in | out}`
9. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **Step 3** router bgp *as-number* | Configures a Border Gateway Protocol (BGP) routing process and places the device in router configuration mode.  
- The *as-number* argument indicates the number of an autonomous system that identifies the device to other BGP devices and tags the routing information passed along.  
- The range is 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535. |
| **Example:**  
Device(config)# router bgp 100 | |
| **Step 4** neighbor {ip-address | peer-group-name} remote-as *as-number* | Adds an entry to the BGP or multiprotocol BGP neighbor table.  
- The *ip-address* argument specifies the IP address of the neighbor.  
- The *peer-group-name* argument specifies the name of a BGP peer group.  
- The *as-number* argument specifies the autonomous system to which the neighbor belongs. |
| **Example:**  
Device(config-router)# neighbor 172.10.0.2 remote-as 200 | |
| **Step 5** address-family vpnv4 [unicast] | Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard Virtual Private Network Version 4 (VPNv4) address prefixes.  
- The optional *unicast* keyword specifies VPNv4 unicast address prefixes. |
| **Example:**  
Device(config-router)# address-family vpnv4 | |
| **Step 6** neighbor {ip-address | peer-group-name} activate | Enables the exchange of information with a neighboring BGP device.  
- The *ip-address* argument specifies the IP address of the neighbor.  
- The *peer-group-name* argument specifies the name of a BGP peer group. |
| **Example:**  
Device(config-router-af)# neighbor 172.16.0.2 activate | |
| **Step 7** neighbor {ip-address | peer-group-name} send-community {both | extended | standard} | Specifies that a communities attribute should be sent to a BGP neighbor.  
- The *ip-address* argument specifies the IP address of the BGP-speaking neighbor.  
- The *peer-group-name* argument specifies the name of a BGP peer group.  
- The *both* keyword sends standard and extended community attributes.  
- The *extended* keyword sends an extended community attribute. |
| **Example:**  
Device(config-router-af)# neighbor 172.16.0.2 send-community extended | |
Verifying the Route Target Replacement Policy

**SUMMARY STEPS**

1. `enable`
2. `show ip bgp vpnv4 vrf vrf-name`
3. `exit`

**DETAILED STEPS**

**Step 1**

`enable`

Enables privileged EXEC mode. Enter your password if prompted.

**Example:**

```
Device> enable
Device# 
```

**Step 2**

`show ip bgp vpnv4 vrf vrf-name`

Verifies that Virtual Private Network Version 4 (VPNV4) prefixes with a specified route target (RT) extended community attribute are replaced with the proper RT extended community attribute to verify that the provider edge (PE) devices receive the rewritten RT extended community attributes.

Verify route target replacement on PE1:

**Example:**
Device# show ip bgp vpnv4 vrf Customer_A 192.168.1.1/32 internal
BGP routing table entry for 65000:1:192.168.1.1/32, version 6901
Paths: (1 available, best #1, table Customer_A)
    Advertised to update-groups: 5
    Refresh Epoch 1
    650002
    3.3.3.3 (metric 3) (via default) from 3.3.3.3 (55.5.4.1)
    Origin IGP, metric 0, localpref 100, valid, internal, best
    Extended Community: RT:65000:1
    mpls labels in/out nolabel/3025
    rx pathid: 0, tx pathid: 0x0
    net: 0xFFB0A72E38, path: 0xFFB0E6A370, pathext: 0xFFB0E5D970
    flags: net: 0x0, path: 0x7, pathext: 0x181

Step 3    exit
Returns to user EXEC mode:
Example:

Device# exit
Device>

Configuration Examples for MPLS VPN Route Target Rewrite

Examples: Configuring Route Target Replacement Policies

This example shows the route target (RT) replacement configuration of a Provider Edge (PE) device that exchanges Virtual Private Network Version 4 (VPNv4) prefixes with another Provider Edge (PE) device. The route map extmap is configured to replace RTs on inbound updates. Any incoming update with RT 65000:2 is replaced with RT 65000:1.

!  ip extcommunity-list 1 permit rt 65000:2
!  route-map rtrewrite permit 10
    match extcommunity 1
    set extcomm-list 1 delete
    set extcommunity rt 65000:1 additive
!

This example shows the use of the route-map configuration continue command when you need to apply more than one replacement rule on an update. In this example, an incoming update with RT 7777:222222222 is replaced with RT 65000:2. Without the continue 20 command, route-map evaluation would stop when a match on sequence 10 is made. With the continue 20 command, route-map evaluation continues into sequence 20 even if a match occurs in sequence 10.

!  ip extcommunity-list 2 permit rt 7777:222222222
  ip extcommunity-list 3 permit rt 2:2
  ip extcommunity-list 4 permit rt 200000:111
Examples: Applying Route Target Replacement Policies

Examples: Associating Route Maps with Specific BGP Neighbor

This example shows the association of route map extmap with a Border Gateway Protocol (BGP) neighbor. The BGP inbound route map is configured to replace route targets (RTs) on incoming updates.

```
router bgp 1
address-family vpnv4
neighbor 2.2.2.2 route-map rtrewrite in
```

This example shows the association of the same route map with the outbound BGP neighbor. The route map is configured to replace RTs on outgoing updates.

```
router bgp 1
address-family vpnv4
neighbor 2.2.2.2 route-map rtrewrite out
```
CHAPTER 12

Configuring Multicast Virtual Private Network

• Configuring Multicast VPN, on page 131

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 11: 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).
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.

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.

SUMMARY STEPS

1. enable
2. configure terminal
3. vrf definition vrf-name
4. rd route-distinguisher
5. route-target both ASN:nn or IP-address:nn
6. address family ipv4 unicast value
7. mdt default group-address
8. mdt data group number
9. mdt data threshold kbps
### 10. mdt log-reuse

### 11. end

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device&gt; enable</td>
<td><strong>Enter your password if prompted.</strong></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>vrf definition vrf-name</td>
<td>Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config)# vrf definition vrf1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>rd route-distinguisher</td>
<td>Creates routing and forwarding tables for a VRF.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-vrf)# rd 1:1</td>
<td><strong>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:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 16-bit autonomous system number (ASN): your 32-bit number. For example, 101:3.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 32-bit IP address: your 16-bit number. For example, 192.168.122.15:1.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>route-target both ASN:nn or IP-address:nn</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-vrf)# route-target both 1:1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>address family ipv4 unicast value</td>
<td>Enters VRF address family configuration mode to specify an address family for a VRF.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-vrf)# address family ipv4 unicast</td>
<td><strong>The ipv4 keyword specifies an IPv4 address family for a VRF.</strong></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>mdt default group-address</td>
<td>Configures the multicast group address range for data MDT groups for a VRF.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-vrf-af)# mdt default 226.10.10.10</td>
<td><strong>A tunnel interface is created as a result of this command.</strong></td>
</tr>
</tbody>
</table>
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.

SUMMARY STEPS

1. enable
2. configure terminal
3. ip multicast-routing
4. ip multicast-routing vrf vrf-name
5. vrf definition vrf-name
6. rd route-distinguisher
7. route-target both ASN:nn or IP-address:nn
8. address family ipv4 unicast value
9. mdt default group-address
10. end
11. configure terminal
12. ip pim vrf vrf-name rp-address value
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 1 | enable | Enables privileged EXEC mode.  
**Example:**  
Device> enable |  
• Enter your password if prompted. |
| Step 2 | configure terminal | Enters global configuration mode.  
**Example:**  
Device# configure terminal | |
| Step 3 | ip multicast-routing | Enables multicast routing.  
**Example:**  
Device(config)# ip multicast-routing | |
| Step 4 | ip multicast-routing vrf vrf-name | Supports the MVPN VRF instance.  
**Example:**  
Device(config)# ip multicast-routing vrf vrf1 | |
| Step 5 | vrf definition vrf-name | Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name.  
**Example:**  
Device(config)# vrf definition vrf1 | |
| Step 6 | rd route-distinguisher | Creates routing and forwarding tables for a VRF.  
**Example:**  
Device(config-vrf)# rd 1:1 |  
• 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 | 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.  
**Example:**  
Device(config-vrf)# route-target both 1:1 | |
| Step 8 | address family ipv4 unicast value | Enters VRF address family configuration mode to specify an address family for a VRF.  
**Example:**  
Device(config-vrf)# address family ipv4 unicast |  
• The `ipv4` keyword specifies an IPv4 address family for a VRF.
### 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)

### SUMMARY STEPS

1. enable
2. configure terminal

### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 9</strong> mdt default group-address</td>
<td>Configures the multicast group address range for data MDT groups for a VRF.</td>
</tr>
<tr>
<td>Example: Device(config-vrf-af)# mdt default 226.10.10.10</td>
<td>• 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.</td>
</tr>
<tr>
<td><strong>Step 10</strong> end</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device(config-vrf-af)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> ip pim vrf vrf-name rp-address value</td>
<td>Enters the RP configuration mode.</td>
</tr>
<tr>
<td>Example: Device(config-vrf-af)# ip pim vrf vrf1 rp-address 1.1.1.1</td>
<td></td>
</tr>
</tbody>
</table>
3. `router bgp as-number`
4. `address-family ipv4 mdt`
5. `neighbor neighbor-address activate`
6. `neighbor neighbor-address send-community [both | extended | standard]`
7. `exit`
8. `address-family vpnv4`
9. `neighbor neighbor-address activate`
10. `neighbor neighbor-address send-community [both | extended | standard]`
11. `end`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> router bgp as-number</td>
<td>Enters router configuration mode and creates a BGP routing process.</td>
</tr>
<tr>
<td>Example: Device(config)# router bgp 65535</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> address-family ipv4 mdt</td>
<td>Enters address family configuration mode to create an IP MDT address family session.</td>
</tr>
<tr>
<td>Example: Device(config-router)# address-family ipv4 mdt</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> neighbor neighbor-address activate</td>
<td>Enables the MDT address family for this neighbor.</td>
</tr>
<tr>
<td>Example: Device(config-router-af)# neighbor 192.168.1.1 activate</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> neighbor neighbor-address send-community [both</td>
<td>extended</td>
</tr>
<tr>
<td>Example: Device(config-router-af)# neighbor 192.168.1.1 send-community extended</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> exit</td>
<td>Exits address family configuration mode and returns to router configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><code>exit</code></td>
<td>Enters address family configuration mode to create a VPNv4 address family session.</td>
</tr>
<tr>
<td><code>address-family vpnv4</code></td>
<td>Enables the VPNv4 address family for this neighbor.</td>
</tr>
<tr>
<td><code>neighbor neighbor-address activate</code></td>
<td>Enables community and (or) extended community exchange with the specified neighbor.</td>
</tr>
<tr>
<td><code>end</code></td>
<td>Exits address family configuration mode and enters privileged EXEC mode.</td>
</tr>
</tbody>
</table>

Verifying Information for the MDT Default Group

**SUMMARY STEPS**

1. `enable`
2. `show ip pim [vrf vrf-name] mdt bgp`
3. `show ip pim [vrf vrf-name] mdt send`
4. `show ip pim vrf vrf-name mdt history interval minutes`

**DETAILED STEPS**

**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 including 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 ss 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

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Command List, All Releases</td>
</tr>
<tr>
<td>For complete syntax and usage information for the commands used in this chapter.</td>
<td>See the Multicast VPN Commands section of the Command Reference (Catalyst 9300 Series Switches)</td>
</tr>
</tbody>
</table>
Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>

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.

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

Table 18: Feature Information for Multicast VPN

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Everest 16.5.1a</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>