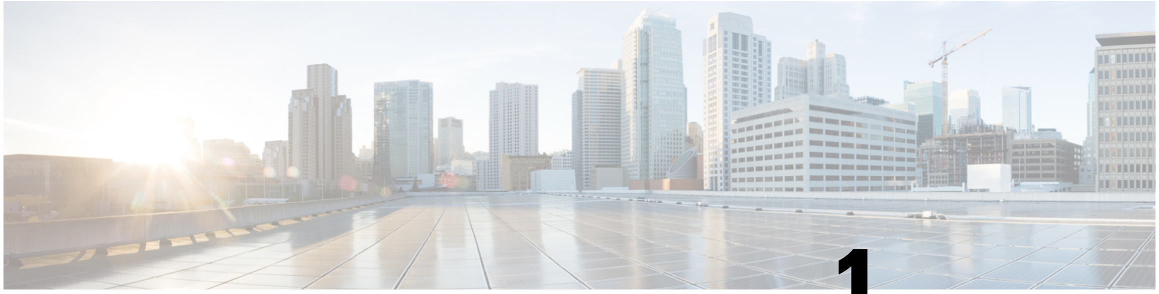
# Bridging, Routing, Segmentation, and QoS Configuration Guide for vEdge Routers, Cisco SD-WAN Releases 19.1, 19.2, and 19.3

**First Published:** 2019-04-25

**Americas Headquarters**
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

# CONTENTS



CHAPTER 1

# What's New for Cisco SD-WAN

**Note** The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

This chapter describes what's new in Cisco SD-WAN for each release.

## What's New for Cisco SD-WAN Release 19.2.x

This section applies to Cisco vEdge devices.

Cisco is constantly enhancing the SD-WAN solution with every release and we try and keep the content in line with the latest enhancements. The following table lists new and modified features we documented in the Configuration, Command Reference, and Hardware Installation guides. For information on additional features and fixes that were committed to the SD-WAN solution, see the *Resolved and Open Bugs* section in the Release Notes.

***Table 1: What's New for Cisco vEdge Device***

| Feature | Description |
|---|---|
| **Getting Started** | |
| API Cross-Site Request Forgery Prevention | This feature adds protection against Cross-Site Request Forgery (CSRF) that occurs when using Cisco SD-WAN REST APIs. This protection is provided by including a CSRF token with API requests. You can put requests on an allowed list so that they do not require protection if needed. See Cross-Site Request Forgery Prevention. |
| **Systems and Interfaces** | |

| Feature | Description |
|---|---|
| Secure Shell Authentication Using RSA Keys | This feature helps configure RSA keys by securing communication between a client and a Cisco SD-WAN server. See SSH Authentication using vManage on Cisco XE SD-WAN Devices. See Configure SSH Authentication. |
| **Policies** | |
| Packet Duplication for Noisy Channels | This feature helps mitigate packet loss over noisy channels, thereby maintaining high application QoE for voice and video. See Configure and Monitor Packet Duplication. |
| Control Traffic Flow Using Class of Service Values | This feature lets you control the flow of traffic into and out of a Cisco device's interface based on the conditions defined in the quality of service (QoS) map. A priority field and a layer 2 class of service (CoS) were added for configuring the re-write rule. See Configure Localized Data Policy for IPv4 Using Cisco vManage. |
| **Security** | |
| Secure Communication Using Pairwise IPsec Keys | This feature allows private pairwise IPSec session keys to be created and installed for secure communication between IPSec devices and its peers. For related information, see IPSec Pairwise Keys Overview. |
| **Network Optimization and High Availability** | |
| Disaster Recovery for vManage | This feature helps you configure Cisco vManage in an active or standby mode to counteract hardware or software failures that may occur due to unforeseen circumstances. See Configure Disaster Recovery. |
| Share VNF Devices Across Service Chains | This feature lets you share Virtual Network Function (VNF) devices across service chains to improve resource utilisation and reduce resource fragmentation. See Share VNF Devices Across Service Chains. |
| Monitor Service Chain Health | This feature lets you configure periodic checks on the service chain data path and reports the overall status. To enable service chain health monitoring, NFVIS version 3.12.1 or later should be installed on all CSP devices in a cluster. See Monitor Service Chain Health. |
| Manage PNF Devices in Service Chains | This feature lets you add Physical Network Function (PNF) devices to a network, in addition to the Virtual Network function (VNF) devices. These PNF devices can be added to service chains and shared across service chains, service groups, and a cluster. Inclusion of PNF devices in the service chain can overcome the performance and scaling issues caused by using only VNF devices in a service chain. See Manage PNF Devices in Service Chains. |

CHAPTER 2

# Bridging

This chapter contains these topics:

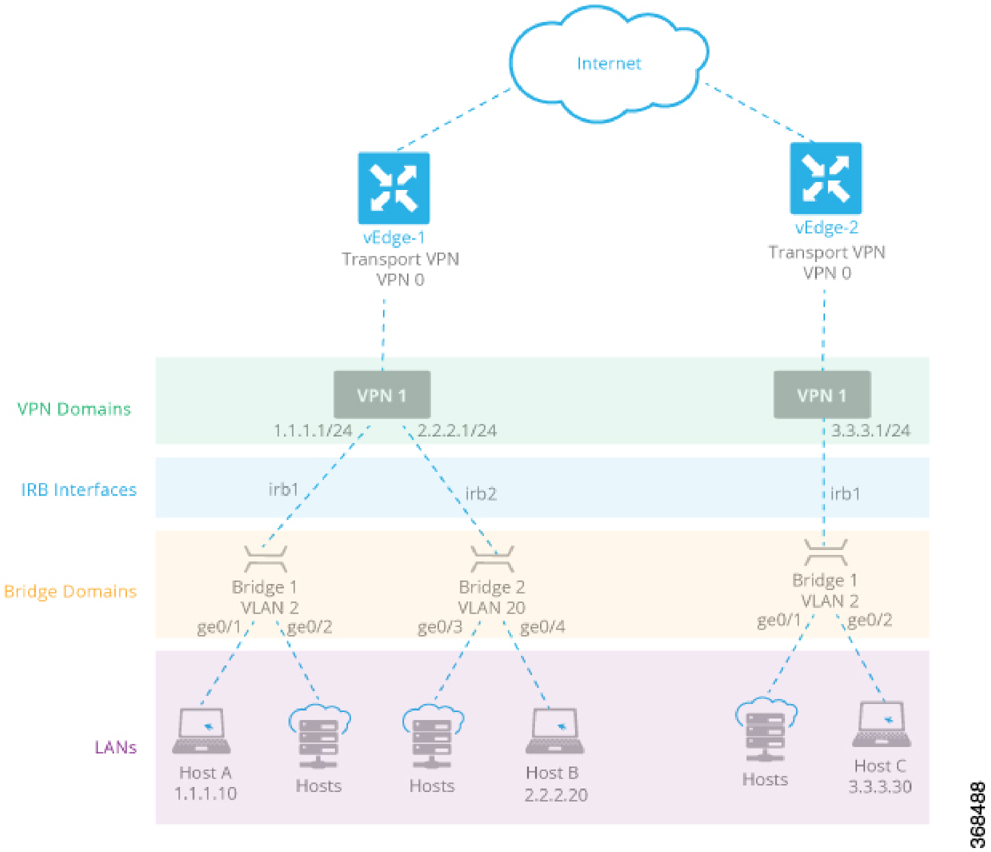
## Bridging Overview

A Cisco vEdge device can act as a transparent bridge, switching traffic between LANs that are part of a VLAN at the local device's site. To implement bridging, the Cisco SD-WAN architecture defines the concept of a *bridge domain*. Each bridge domain corresponds to a single VLAN. From a switching point of view, each bridge domain is a separate broadcast domain, and each has its own Ethernet switching table (or MAC table) to use for switching traffic within the broadcast domain. Multiple bridge domains, and hence multiple VLANs, can coexist on a single Cisco vEdge device.

To allow hosts in different bridge domains to communicate with each other, Cisco vEdge devices support *integrated routing and bridging* (IRB). IRB is implemented using *logical IRB interfaces*, which connect a bridge domain to a VPN, or what might better be called a *VPN domain*. The VPN domain provides the Layer 3 routing services necessary so that traffic can be exchanged between different VLANs. Each bridge domain can have a single IRB interface and can connect to a single VPN domain, and a single VPN domain can connect to multiple bridge domains on a vEdge router. The route table in the VPN domain provides reachability between all bridge domains which participate in that VPN domain, whether the bridge domain is located on the local router or on a remote router.

## Components of Bridging

The following figure illustrates the components of the Cisco SD-WAN bridging solution.



## Bridge Domains

In standard transparent bridging, virtual LANs, or VLANs, segregate LANs into logical LANs, and each VLAN is an isolated broadcast domain. All VLAN traffic remains in the VLAN, and it is directed to its destination by means of Ethernet switching tables. The Cisco SD-WAN implementation of bridging overlays the concept of a *bridge domain* on top of the standard VLAN: A bridge domain comprises a single VLAN, and all the ports within a VLAN are part of a single broadcast domain. Within each broadcast domain, the standard bridging operations of learning, forwarding, flooding, filtering, and aging are performed on VLAN traffic to create and maintain the Ethernet switching table (or MAC table) for that VLAN, and hence for that bridge domain.

Each bridge domain is identified by a number. The VLAN within a bridge domain is identified by an 802.1Q identifier, which is called a VLAN tag or VLAN ID. Frames within a bridge domain can remain untagged, or you can configure a VLAN ID to tag the frames. In the Cisco SD-WAN design, the VLAN and the VLAN ID are the property of the bridge domain. They are not the property of an interface or a switching port.

Ports that connect to the WAN segments are associated with a bridge domain. In the Cisco SD-WAN overlay network, these ports are the physical Gigabit Ethernet interfaces on Cisco vEdge devices. Specifically, they are the base interfaces, for example, **ge-0/0**. You cannot use subinterfaces for bridge domain ports.

Each broadcast domain in the Cisco SD-WAN overlay network is uniquely identified by the combination of bridge domain number and VLAN ID (if configured). This design means that The same VLAN ID can be used in different bridge domains on a single Cisco vEdge device. For example, the VLAN ID 2 can exist in bridge domain 1 and bridge domain 50. In a situation where the VLAN IDs are different, two bridge domains can include the same port interfaces. For example, both (bridge 2, VLAN 2) and (bridge 10, VLAN 23) can include interfaces ge0/0 and ge0/1. Here, these two interfaces effectively become trunk ports. However, because of how interface names are tracked internally, two bridge domains that use the same VLAN ID can have no overlap between the interfaces in the two domains. For example, if (bridge 1, VLAN 2) includes interfaces ge0/0 and ge0/1, these interfaces cannot be in (bridge 50, VLAN 2).

As mentioned above, all member interfaces within a VLAN are part of a single broadcast domain. Within each broadcast domain, the standard transparent bridging operations of learning, forwarding, flooding, filtering, and aging are performed on VLAN traffic to create and maintain the Ethernet switching table, also called the MAC table, for that VLAN.

The Cisco SD-WAN bridging domain architecture lacks the concepts of access ports and trunk ports. However, the Cisco SD-WAN architecure emulates these functions. For a Cisco vEdge device that has a single bridge domain, the interfaces in the bridge emulate access ports and so the router is similar to a single switch device. For a Cisco vEdge device with multiple bridge domains that are tagged with VLAN IDs, the interfaces in the bridges emulate trunk ports, and you can think of each domain as corresponding to a separate switching device.

### Native VLAN

Cisco SD-WAN bridge domains support 802.1Q native VLAN. All traffic sent and received on an interface configured for native VLAN do not have a VLAN tag in its Ethernet frame. That is, they are not tagged with a VLAN ID. If a host is connected on an interface enabled for native VLAN, the bridge domain receives no tagged frames. If the bridge domain connects to a switch that support trunk ports or connects to a hub, the bridge domain might receive both untagged and tagged frames.

Native VLAN is used primarily on trunk ports. VLAN provides backwards compatibility for devices that do not support VLAN tagging. For example, native VLAN allows trunk ports to accept all traffic regardless of what devices are connected to the port. Without native VLAN, the trunk ports would accept traffic only from devices that support VLAN tagging.

### Integrated Routing and Bridging (IRB)

Bridge domains and VLANs provide a means to divide a LAN into smaller broadcast domains. Each VLAN is a separate broadcast domain, and switching within that domain directs traffic to destinations within the VLAN. The result is that hosts within a single bridge domain can communicate among themselves, but cannot communicate with hosts in other VLANs. So, for example, if a business places its departments in separate VLANs, people within the finance department would be able to communicate only with others in that department, but would not be able to communicate with the manufacturing or engineering department.

The only way for traffic to cross Layer 2 VLAN boundaries to allow communicatation between bridge domains is via Layer 3 routing. This process of marrying switching and routing is done by *integrated routing and bridging*, or IRB. With IRB, a single Cisco vEdge device can pass traffic among different bridge domains on the same router and among bridge domains on remote vEdge routers. The only restriction is that all the bridge domains must reside in the same VPN domain in the overlay network.

The Cisco SD-WAN implementation of IRB connects a Layer 2 bridge domain to a Layer 3 VPN domain via an IRB interface. An IRB interface is a logical interface that inherits all the properties of a regular interface, but it is not associated with a port or with a physical interface. Each IRB interface is named with the stem “irb” and a number that matches the number of a bridge domain. For example, the interface **irb2** is the logical interface that connects to bridge domain 2. IRB interfaces cannot have subinterfaces.

You create IRB interfaces within a VPN. A VPN domain supports multiple IRB interfaces.

There is a one-to-one association between an IRB logical interface and a bridge domain: an IRB interface can be associated only with one bridge domain, and a bridge domain can be associated with only one IRB interface. As a result, a bridge domain can be part of only one VPN in the overlay network.

The IP address of an IRB interface is the subnet of the VLAN that resides in the bridge domain. From a switching perspective, the IP address of the IRB interface is part of the bridge domain.

# Configure Bridging Using Cisco vManage

To have a Cisco vEdge device act as a transparent bridge, configure bridging domains on the router. A router can have up to 16 bridging domains.

## Configure Bridging and Bridge Domains

1. In Cisco vManage, select **Configuration** > **Templates**.
2. Click the **Feature** tab to view your existing feature templates or create a new one.
3. Click **Add Template**. Select a device from the list of devices. The templates available for the selected device display in the right pane.
4. Choose the **Bridge** template.
5. Enter a name and description for the template.
6. Configure bridging domains under the **Basic Configuration** tab.

| Parameter Name | Description |
|---|---|
| Bridge Name | Enter a text description of the bridging domain. It can be up to 32 characters. |
| VLAN ID | Enter the VLAN identifier to associate with the bridging domain.<br>*Range:* 0 through 4095 |
| Maximum MAC Addresses | Specify the maximum number of MAC addresses that the bridging domain can learn.<br>*Range:* 0 through 4096<br>*Default:* 1024 |
| Age-Out Time (Seconds) | Specify how long to store an entry in the MAC table before it ages out.<br>*Range:* 10 through 4096 seconds<br>*Default:* 300 seconds (5 minutes) |

7. Click **Save**.

### Associate Interfaces with the Bridge Domain

To associate an interface with the bridge domain, click the Interface tab and click the **New Interface** button.

| Parameter Name | Description |
|---|---|
| Interface Name | Enter the name of the interface to associate with the bridging domain, in the format **ge***slot*/*port*. |
| Description | Enter a text description of the interface. |
| Native VLAN Support | Click Enabled to configure the interface to carry untagged traffic. By default, native VLAN is disabled. |
| Shutdown | Click No to enable the interface. By default, an interface in a bridge domain is disabled. |
| Static MAC Address | Click **Add Static MAC Address**, and in the MAC Static Address field that appears, enter a static MAC address entry for the interface in the bridge domain. Click Add MAC Address to add another static MAC address entry for the interface. Click Save to save the MAC address or addresses. |

# Configure Interface Bridge

Integrated routing and bridging (IRB) allows Cisco vEdge devices in different bridge domains to communicate with each other. To enable IRB, create logical IRB interfaces to connect a bridge domain to a VPN. The VPN provides the Layer 3 routing services necessary so that traffic can be exchanged between different VLANs. Each bridge domain can have a single IRB interface and can connect to a single VPN, and a single VPN can connect to multiple bridge domains on a Cisco vEdge device.

1. In Cisco vManage, navigate to **Configuration** > **Templates**.
2. Click the **Feature** tab to view your existing feature templates or create a new one.
3. Click **Add Template**. Select a device from the list of devices. The templates available for the selected device display in the right pane.
4. Choose the **VPN Interface Bridge** template.
5. Enter a name and description for the template and enter the parameter. Enter other parameters described in the subsequent sections.

### Create a Bridging Interface

To configure an interface to use for bridging servers, select the **Basic Configuration** tab and enter the following details.

| Parameter Name | Description |
|---|---|
| Shutdown | Click No to enable the interface. |

| Parameter Name | Description |
|---|---|
| Interface Name | Enter the name of the interface, in the format **irb***number*. The IRB interface number can be from 1 through 63, and must be the same as the VPN identifier configured in the Bridge feature template for the bridging domain that the IRB is connected to. |
| Description | Enter a description for the interface. |
| IPv4 Address | Enter the IPv4 address of the router. |
| DHCP Helper | Enter up to eight IP addresses for DHCP servers in the network, separated by commas, to make the interface a DHCP helper. A DHCP helper interface forwards BOOTP (Broadcast) DHCP requests that it receives from the specified DHCP servers. |
| Block Non Source IP | Click Yes to have the interface forward traffic only if the source IP address of the traffic matches the interface's IP prefix range. |
| Secondary IP Address (Maximum: 4) | Click Add to configure up to four secondary IPv4 addresses for a service-side interface. |

To save the template, click **Save**.

### Apply Access Lists

To apply access lists to IRB interfaces, select the ACL tab and configure the following parameters:

| Parameter Name | Description |
|---|---|
| Ingress ACL - IPv4 | Click On, and specify the name of an IPv4 access list to packets being received on the interface. |
| Egress ACL - IPv4 | Click On, and specify the name of an IPv4 access list to packets being transmitted on the interface. |

To save the template, click **Save**.

### Configure VRRP

To have an interface run the Virtual Router Redundancy Protocol (VRRP), which allows multiple devices to share a common virtual IP address for default gateway redundancy, select the VRRP tab. Then click the **New VRRP** button and configure the following parameters:

| Parameter Name | Description |
|---|---|
| Group ID | Enter the virtual router ID, which is a numeric identifier of the virtual router. You can configure a maximum of 24 groups.*Range:* 1 through 255 |

| Parameter Name | Description |
|---|---|
| Priority | Enter the priority level of the router. There router with the highest priority is elected as the primary device. If two Cisco vEdge devices have the same priority, the one with the higher IP address is elected as the primary one.*Range:* 1 through 254*Default:* 100 |
| Timer | Specify how often the primary VRRP router sends VRRP advertisement messages. If the subordinate routers miss three consecutive VRRP advertisements, they elect a new primary router.*Range:* 1 through 3600 seconds *Default:* 1 second |
| Track OMP<br>Track Prefix List | By default, VRRP uses of the state of the service (LAN) interface on which it is running to determine which Cisco vEdge device is the primary virtual router. If a Cisco vEdge device loses all its WAN control connections, the LAN interface still indicates that it is up even though the router is functionally unable to participate in VRRP. To take WAN side connectivity into account for VRRP, configure one of the following:<br>Track OMP—Click **On** for VRRP to track the Overlay Management Protocol (OMP) session running on the WAN connection. If the primary VRRP router loses all its OMP sessions, VRRP elects a new default gateway from those that have at least one active OMP session.<br>Track Prefix List—Track both the OMP session and a list of remote prefixes, which is defined in a prefix list configured on the local router. If the primary VRRP router loses all its OMP sessions, VRRP failover occurs as described for the Track OMP option. In addition, if reachability to one of the prefixes in the list is lost, VRRP failover occurs immediately, without waiting for the OMP hold timer to expire, thus minimizing the amount of overlay traffic is dropped while the Cisco IOS XE SD-WAN device determines the primary VRRP router. |
| IP Address | Enter the IP address of the virtual router. This address must be different from the configured interface IP addresses of both the local Cisco IOS XE SD-WAN device and the peer running VRRP. |

### Add ARP Table Entries

| Parameter Name | Description |
|---|---|
| IP Address | Enter the IP address for the ARP entry in dotted decimal notation or as a fully qualified host name. |
| MAC Address | Enter the MAC address in colon-separated hexadecimal notation. |

To save the ARP configuration, click **Add**.

To save the template, click **Save**.

### Advanced Interface Properties

To configure other interface properties, select the Advanced tab and configure the following parameters:

| Parameter Name | Description |
|---|---|
| MAC Address | Specify a MAC address to associate with the interface, in colon-separated hexadecimal notation. |
| IP MTU | Specify the maximum MTU size of packets on the interface.<br>*Range:* 576 through 1804<br>*Default:* 1500 bytes |
| TCP MSS | Specify the maximum segment size (MSS) of TPC SYN packets passing through the vEdge router. By default, the MSS is dynamically adjusted based on the interface or tunnel MTU such that TCP SYN packets are never fragmented.<br>*Range:* 552 to 1460 bytes<br>*Default:* None |
| Clear-Dont-Fragment | Click On to clear the Don't Fragment (DF) bit in the IPv4 packet header for packets being transmitted out the interface. When the DF bit is cleared, packets larger than that interface's MTU are fragmented before being sent. |
| ARP Timeout | Specify how long it takes for a dynamically learned ARP entry to time out.<br>*Range:* 0 through 2678400 seconds (744 hours)<br>*Default:* 1200 seconds (20 minutes) |
| ICMP Redirect | Click Disable to disable ICMP redirect messages on the interface. By default, an interface allows ICMP redirect messages. |

To save the template, click **Save**.

# Configure Bridging Using CLI

## Configure Bridging and Bridge Domains Using CLI

Bridge domains can be marked with a VLAN tag, or they can remain untagged.

### Create a Bridge Domain That Uses VLAN Tagging

For a bridge domain that uses VLAN tagging, a tag, called a VLAN ID, is inserted into all frame headers sent by the domain This tag identifies which VLAN the frames belong to, and it is used to determine which interfaces the Cisco vEdge device should send broadcast packets to.

To configure a bridge domain that uses VLAN tagging, create a bridging domain, assign a VLAN tag to that domain, and associate an interface with the domain:

1. Create a bridging domain:

   ```
   vEdge(config)#  bridge bridge-id
   ```

   Each domain is identified by a unique integer, in the range 1 through 63. Each Cisco vEdge device can have up to 16 bridging domains.

2. Tag the bridging domain with a VLAN ID:

   ```
   vEdge(config-bridge)# vlan number
   ```

   The VLAN identifier can be a value from 1 through 4095.

3. Associate an interface with the bridging domain, and enable that interface:

   ```
   vEdge(config-bridge)# interface  ge slot/port
   vEdge(config-interface)# no shutdown
   ```

   The interface must be a physical interface. You cannot use subinterfaces.

After you have added physical interfaces to a VLAN, if you want to change the VLAN identifier, you must first delete all the interfaces from the VLAN. Then configure a new VLAN identifier, and re-add the interfaces to the VLAN.

You can also configure these optional parameters:

1. Configure a description for the VLAN interface, to help identify the interface in operational command output:

   ```
   vEdge(config-bridge)# interface ge slot
   /
   port
   vEdge(config-interface)# description "
   text description "
   ```

2. Configure a static MAC address for the VLAN interface:

   ```
   vEdge(config-interface)# static-mac-address aa
   :
   bb
   :
   cc
   ```

```
:
dd
:
ee
:ff
```

3. Configure a name for the VLAN, to help identify the VLAN in operational command output:

   ```
   vEdge(config-bridge)# name "text description"
   ```

4. By default, a bridging domain can learn up to 1024 MAC addresses. You can modify this to a value from 0 through 4096:

   ```
   vEdge(config-bridge)#  max-macs number
   ```

5. By default, MAC table entries age out after 300 seconds (5 minutes). You can modify this to a value from 10 through 4096 seconds:

   ```
   vEdge(config-bridge)#  age-time seconds
   ```

Here is an example configuration:

```
vEdge# config
vEdge(config)# bridge 2
vEdge(bridge-2)# vlan 27
vEdge(bridge-2)# interface ge0/4
vEdge(interface-ge0-4)# no shutdown
vEdge(interface-ge0-4)# description "VLAN tag = 27"
vEdge(interface-ge0/4)# commit and-quit
vEdge# show running-config bridge
bridge 2
 vlan 27
 interface ge0/4
  description "VLAN tag = 27"
  no native-vlan
  no shutdown
 !
!
vEdge#
```

After your have configured an interface in a bridge domain, you add or change a VLAN identifier for that domain only by first deleting the bridge domain from the configuration (with a **no bridge** *bridge-id* command) and then reconfiguing the domain with the desired interface name and VLAN tag identifier.

To see which interfaces bridging is running on, use the **show bridge interface** command:

```
vEdge# show bridge interface
                         ADMIN   OPER    ENCAP                  RX     RX      TX     TX

BRIDGE   INTERFACE  VLAN  STATUS  STATUS  TYPE   IFINDEX  MTU   PKTS   OCTETS  PKTS   OCTETS

------------------------------------------------------------------------------------------
2        ge0/4      27    Up      Up      vlan   41       1500  4      364     0      0
```

"Up" in the Admin Status column indicates that the interface has been configured, and "Up" in the Oper Status column indicates that bridging is running on the interface.

### Create a Bridge Domain with an Untagged VLAN

All frames in an untagged VLAN are sent with no VLAN tag, or VLAN ID, in the frame header. For frames that already contain a tag, the tag is removed before it is sent.

In the minimal configuration for a tagged VLAN, you simply create a bridging domain that contains an interface:

1. Create a bridging domain. This domain is identified by a unique integer.

   ```
   vEdge(config)# bridge number
   ```

   On each vEdge router, you can configure up to 16 bridging domains.

2. Associate an interface with the bridging domain, and enable that interface:

   ```
   vEdge(config-bridge)# interface  interface-name
   vEdge(config-interface)# no shutdown
   ```

You can also configure the optional parameters described in the previous section.

### Configure a Native VLAN

In the minimal configuration for a native VLAN, you create a bridging domain that contains an interface, and you mark that interface as a native VLAN interface:

1. Create a bridging domain. This domain is identified by a unique integer.

   ```
   vEdge(config)# bridge number
   ```

   On each vEdge router, you can configure up to 16 bridging domains.

2. Associate an interface with the bridging domain, and enable that interface:

   ```
   vEdge(config-bridge)# interface  interface-name
   vEdge(config-interface)# no shutdown
   ```

3. Enabled native VLAN on the interface:

   ```
   vEdge(config-interface)# native-vlan
   ```

You can also configure the optional parameters described in the section about creating a tagged VLAN.

# Configure IRB

With bridging, all frame traffic remains within its VLAN. To allow frames to be passed among different VLANs, you enable integrated routing and bridging (IRB). To do this, you create a logical IRB interface in a VPN domain that connects to the bridge domain. Frames with destinations in other VLANs travel over the IRB interface to the VPN domain, and the Layer 3 route table is used to forward the frames toward their destination. The route table learns the routes to other IRB interfaces. With IRB, communication can be established between VLANs that are connected to the same VPN. The VLANs can be both on the local vEdge router and on a remote router.

In a minimal configuration to configure IRB, you create an IRB interface and assign it an IP address:

1. In the desired VPN, create an IRB interface:

   ```
   vEdge(config)# vpn number
   vEdge(config-vpn)# interface  irb number
   ```

   The VPN number can be any number from 1 through 65530, which correspond to service VPNs, except for 512 (which is the management VPN). You cannot place IRB interfaces in either the transport VPN (VPN 0) or the management VPN (VPN 512). The IRB interface type is **irb**. The IRB interface number is a number from 1 through 63, and it must be the same number as the the identifier of the bridging domain that the IRB is connected to. For example, if you configure a bridging domain with an identifier of 2 (with the command **bridge 2**), the IRB interface number must be 2, and so you must configure **interface irb2**.

2. Configure an IP address for the IRB interface. This address is the subnet for the VLAN in the connected bridge domain:

   ```
   vEdge(config-irb)# ip address prefix/length
   ```

3. Enable the interface:

   ```
   vEdge(config-irb)# no shutdown
   ```

In all respects, the logical IRB interfaces is just another interface. This means, for instance, that you can configure additional interfaces properties as desired. (Note, however, that you cannot configure autonegotiation on IRB interfaces.) It also means that you can ping a logical IRB interface from another device in the same VPN, and you can ping the interface regardless of whether a corresponding bridge exists for that IRB interface. That is, if you configure interface **irb4**, but there is no corresponding **bridge 4**, you are still able to ping **irb4**.

Here is an example IRB configuration:

```
vEdge# show running-config vpn 1
vpn 1
 interface ge0/4
  ip address 10.20.24.15/24
  no shutdown
 !
 interface irb1
  ip address 1.1.1.15/24
  no shutdown
  access-list IRB_ICMP in
  access-list IRB_ICMP out
 !
 interface irb50
  ip address 3.3.3.15/24
  no shutdown
 !
!
vEdge# show running-config vpn 2
vpn 2
 interface irb2
  ip address 2.2.2.15/24
  no shutdown
 !
!
```

To display information about the IRB interfaces, use the **show interface** command. The IRB interfaces are listed in the Interface column, and the Encapsulation Type columns marks these interfaces as "vlan".

```
vEdge# show interface

                                IF      IF                                                                 TCP
                                ADMIN   OPER    ENCAP                                        SPEED          MSS                       RX      TX
VPN  INTERFACE  IP ADDRESS        STATUS  STATUS  TYPE   PORT TYPE  MTU   HWADDR             MBPS   DUPLEX  ADJUST  UPTIME      PACKETS PACKETS
-----------------------------------------------------------------------------------------------------------------------------------------------
0    ge0/0      10.1.15.15/24     Up      Up      null   transport  1500  00:0c:29:cb:4f:9c  10     full    0       0:02:48:12  1467    1460
0    ge0/1      -                 Up      Up      null   service    1500  00:0c:29:cb:4f:a6  10     full    0       0:02:48:12  0       0
0    ge0/2      -                 Up      Up      null   service    1500  00:0c:29:cb:4f:b0  10     full    0       0:02:48:03  0       0
0    ge0/3      10.0.20.15/24     Up      Up      null   service    1500  00:0c:29:cb:4f:ba  10     full    0       0:02:48:12  0       0
0    ge0/5      -                 Up      Up      null   service    1500  00:0c:29:cb:4f:ce  10     full    0       0:02:48:03  0       0
0    ge0/6      -                 Up      Up      null   service    1500  00:0c:29:cb:4f:d8  10     full    0       0:02:48:03  0       0
0    ge0/7      10.0.100.15/24    Up      Up      null   service    1500  00:0c:29:cb:4f:e2  10     full    0       0:02:48:12  0       0
0    system     172.16.255.15/32  Up      Up      null   loopback   1500  00:00:00:00:00:00  10     full    0       0:02:48:12  0       0
1    ge0/4      10.20.24.15/24    Up      Up      null   service    1500  00:0c:29:cb:4f:c4  10     full    0       0:02:48:00  92      14
1    irb1       1.1.1.15/24       Up      Up      vlan   service    1500  00:0c:00:00:aa:00  10     full    0       0:02:48:00  1178    0
1    irb50      3.3.3.15/24       Up      Up      vlan   service    1500  00:0c:00:00:aa:00  10     full    0       0:02:48:00  0       0
2    irb2       2.2.2.15/24       Up      Up      vlan   service    1500  00:0c:00:00:aa:00  10     full    0       0:02:48:01  0       0
512  eth0       10.0.1.15/24      Up      Up      null   service    1500  00:50:56:00:01:05  1000   full    0       0:02:48:01  210     148
```

# Configuration and Monitoring Commands

CLI commands for configuring and monitoring Layer 2 bridging and Layer 3 integrated routing and bridging (IRB) on Cisco vEdge routers.

### Bridging Configuration Commands

Use the following commands to configure bridging on a vEdge router.

```
bridge bridge-id
  age-time seconds
  interface interface-name
    description "text description"
    native-vlan
    [no] shutdown
    static-mac-address mac-address
  max-macs number
  name text
  vlan number
```

### Bridging Monitoring Commands

Use the following commands to monitor Layer 2 bridging on a vEdge router:

- **clear bridge mac** — Clear the MAC addresses that the vEdge router has learned.
- **clear bridge statistics** —Clear the bridging statistics.
- **show bridge interface** —List information about the interfaces on which bridging is configured.
- **show bridge mac** —List the MAC addresses that the vEdge router has learned.
- **show bridge table** —List the information in the bridge forwarding table.

### IRB Configuration Commands

Use the following commands to configure IRB within a VPN on a vEdge router:

```
vpn vpn-id
  interface irbnumber
    access-list acl-list
    arp
      ip address ip-address mac mac-address
    arp-timeout seconds
    autonegotiate
    clear-dont-fragment
    description "text description"
    dhcp-server (on vEdge routers only)
      address-pool prefix/length
      exclude ip-address
      lease-time minutes
      max-leases number
      offer-time minutes
      options
        default-gateway ip-address
        dns-servers ip-address
        domain-name domain-name
        interface-mtu mtu
        tftp-servers ip-address
      static-lease mac-address
    ip address address/subnet
    mac-address mac-address
    mtu bytes
    [no] shutdown
    tcp-mss-adjust bytes
```

### IRB Monitoring Commands

Use the following commands to monitor IRB:

- **show interface** —List information about the interfaces on which IRB is enabled.

CHAPTER 3

# Unicast Overlay Routing

The overlay network is controlled by the Cisco SD-WAN Overlay Management Protocol (OMP), which is at the heart of Cisco SD-WAN overlay routing. This solution allows the building of scalable, dynamic, on-demand, and secure VPNs. The Cisco SD-WAN solution uses a centralized controller for easy orchestration, with full policy control that includes granular access control and a scalable secure data plane between all edge nodes.

The Cisco SD-WAN solution allows edge nodes to communicate directly over any type of transport network, whether public WAN, internet, metro Ethernet, MPLS, or anything else.

# Supported Protocols

## OMP Routing Protocol

The Cisco SD-WAN Overlay Management Protocol (OMP) is the protocol responsible for establishing and maintaining the Cisco SD-WAN control plane. It provides the following services:

- Orchestration of overlay network communication, including connectivity among network sites, service chaining, and VPN or VRF topologies
- Distribution of service-level routing information and related location mappings
- Distribution of data plane security parameters
- Central control and distribution of routing policy

OMP is the control protocol that is used to exchange routing, policy, and management information between Cisco vSmart Controllers and  Cisco vEdge devices in the overlay network. These devices automatically initiate OMP peering sessions between themselves, and the two IP end points of the OMP session are the system IP addresses of the two devices.

OMP is an all-encompassing information management and distribution protocol that enables the overlay network by separating services from transport. Services provided in a typical VPN setting are usually located within a VPN domain, and they are protected so that they are not visible outside the VPN. In such a traditional architecture, it is a challenge to extend VPN domains and service connectivity.

OMP addresses these scalability challenges by providing an efficient way to manage service traffic based on the location of logical transport end points. This method extends the data plane and control plane separation concept from within routers to across the network. OMP distributes control plane information along with related policies. A central Cisco vSmart Controller makes all decisions related to routing and access policies for the overlay routing domain. OMP is then used to propagate routing, security, services, and policies that are used by edge devices for data plane connectivity and transport.

## OMP Route Advertisements

On Cisco vSmart Controllers and Cisco vEdge devices, OMP advertises to its peers the routes and services that it has learned from its local site, along with their corresponding transport location mappings, which are called TLOCs. These routes are called OMP routes or vRoutes to distinguish them from standard IP routes. The routes advertised are actually a tuple consisting of the route and the TLOC associated with that route. It is through OMP routes that the Cisco vSmart Controllers learn the topology of the overlay network and the services available in the network.

OMP interacts with traditional routing at local sites in the overlay network. It imports information from traditional routing protocols, such as OSPF and BGP, and this routing information provides reachability within the local site. The importing of routing information from traditional routing protocols is subject to user-defined policies.

Because OMP operates in an overlay networking environment, the notion of routing peers is different from a traditional network environment. From a logical point of view, the overlay environment consists of a centralized controller and a number of edge devices. Each edge device advertises its imported routes to the centralized controller and based on policy decisions, this controller distributes the overlay routing information to other edge devices in the network. Edge devices never advertise routing information to each other, either using OMP or any other method. The OMP peering sessions between the centralized controller and the edge devices are used exclusively to exchange control plane traffic; they are never, in any situation, used for data traffic.

Registered edge devices automatically collect routes from directly connected networks as well as static routes and routes learned from IGP protocols. The edge devices can also be configured to collect routes learned from BGP.

OMP performs path selection, loop avoidance, and policy implementation on each local device to decide which routes are installed in the local routing table of any edge device.

OMP advertises the following types of routes:

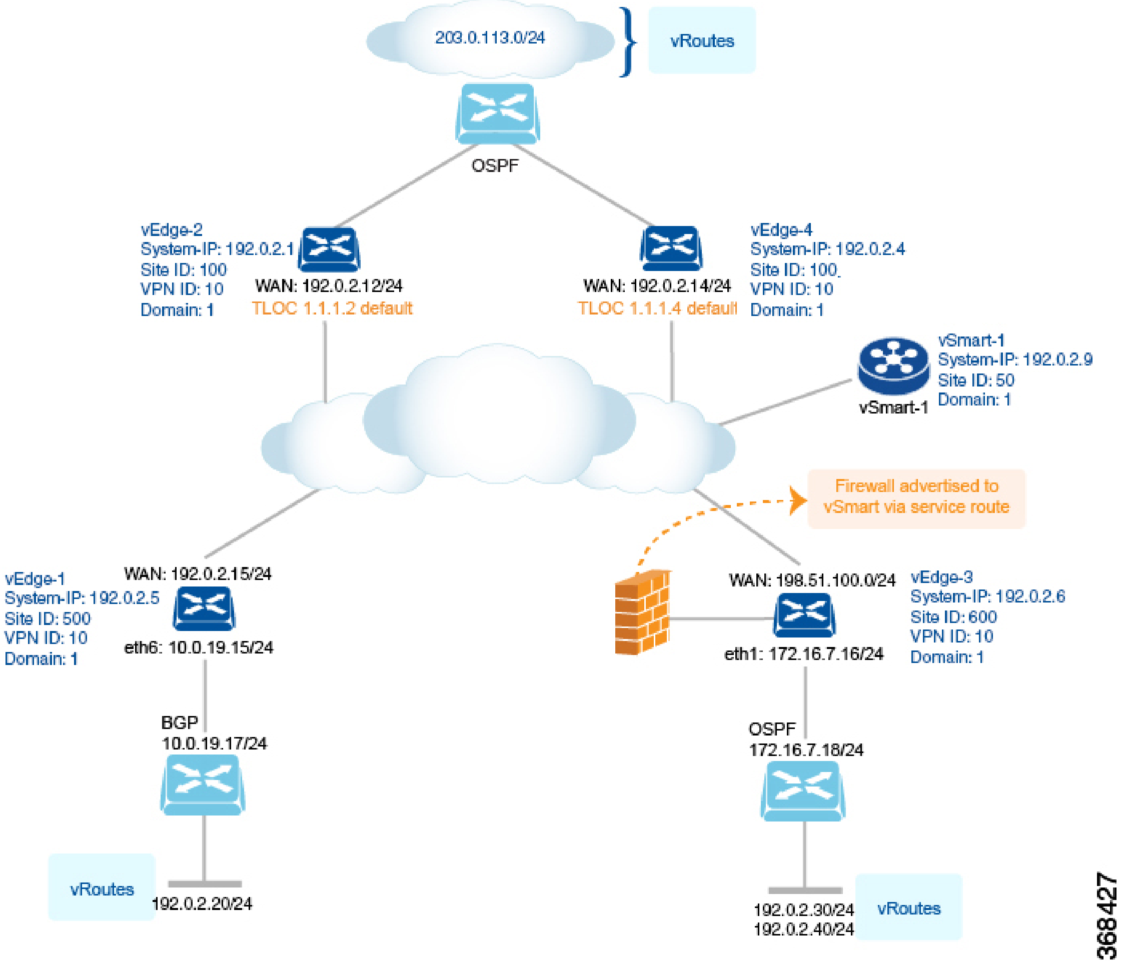
- OMP routes (also called vRoutes)—Prefixes that establish reachability between end points that use the OMP-orchestrated transport network. OMP routes can represent services in a central data center, services at a branch office, or collections of hosts and other end points in any location of the overlay network. OMP routes require and resolve into TLOCs for functional forwarding. In comparison with BGP, an OMP route is the equivalent of a prefix carried in any of the BGP AFI/SAFI fields.

- Transport locations (TLOCs)—Identifiers that tie an OMP route to a physical location. The TLOC is the only entity of the OMP routing domain that is visible to the underlying network, and it must be reachable via routing in the underlying network. A TLOC can be directly reachable via an entry in the routing table of the physical network, or it must be represented by a prefix residing on the outside of a NAT device and must be included in the routing table. In comparison with BGP, the TLOC acts as the next hop for OMP routes.

- Service routes—Identifiers that tie an OMP route to a service in the network, specifying the location of the service in the network. Services include firewalls, Intrusion Detection Systems (IDPs), and load balancers. Service route information is carried in both service and OMP routes.

(OMP also advertises policies configured on the Cisco vSmart Controllers that are executed on Cisco vEdge devices including application-routing policy, cflowd flow templates, and data policy. For more information, see *Policy Overview*.)

The following figure illustrates the three types of OMP routes.



### OMP Routes

Each device at a branch or local site advertises OMP routes to the Cisco vSmart Controllers in its domain. These routes contain routing information that the device has learned from its site-local network.

A Cisco SD-WAN device can advertise one of the following types of site-local routes:

- Connected (also known as direct)
- Static
- BGP
- OSPF (inter-area, intra-area, and external)

OMP routes advertise the following attributes:

- TLOC—Transport location identifier of the next hop for the vRoute. It is similar to the BGP NEXT_HOP attribute. A TLOC consists of three components:
    - System IP address of the OMP speaker that originates the OMP route
    - Color to identify the link type
    - Encapsulation type on the transport tunnel
- Origin—Source of the route, such as BGP, OSPF, connected, and static, and the metric associated with the original route.
- Originator—OMP identifier of the originator of the route, which is the IP address from which the route was learned.
- Preference—Degree of preference for an OMP route. A higher preference value is more preferred.
- Service—Network service associated with the OMP route.
- Site ID—Identifier of a site within the Cisco SD-WAN overlay network domain to which the OMP route belongs.
- Tag—Optional, transitive path attribute that an OMP speaker can use to control the routing information it accepts, prefers, or redistributes.
- VPN—VPN or network segment to which the OMP route belongs.

You configure some of the OMP route attribute values, including the system IP, color, encapsulation type, carrier, preference, service, site ID, and VPN. You can modify some of the OMP route attributes by provisioning control policy on the Cisco vSmart Controller.

### TLOC Routes

TLOC routes identify transport locations. These are locations in the overlay network that connect to physical transport, such as the point at which a WAN interface connects to a carrier. A TLOC is denoted by a 3-tuple that consists of the system IP address of the OMP speaker, a color, and an encapsulation type. OMP advertises each TLOC separately.

TLOC routes advertise the following attributes:

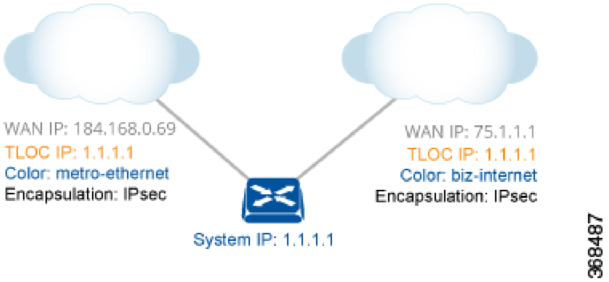
- TLOC private address—Private IP address of the interface associated with the TLOC.
- TLOC public address—NAT-translated address of the TLOC.
- Carrier—An identifier of the carrier type, which is generally used to indicate whether the transport is public or private.
- Color—Identifies the link type.
- Encapsulation type—Tunnel encapsulation type.
- Preference—Degree of preference that is used to differentiate between TLOCs that advertise the same OMP route.
- Site ID—Identifier of a site within the Cisco SD-WAN overlay network domain to which the TLOC belongs.

- Tag—Optional, transitive path attribute that an OMP speaker can use to control the flow of routing information toward a TLOC. When an OMP route is advertised along with its TLOC, both or either can be distributed with a community TAG, to be used to decide how send traffic to or receive traffic from a group of TLOCs.
- Weight—Value that is used to discriminate among multiple entry points if an OMP route is reachable through two or more TLOCs.

The IP address used in the TLOC is the fixed system address of the device itself. The reason for not using an IP address or an interface IP address to denote a TLOC is that IP addresses can move or change; for example, they can be assigned by DHCP, or interface cards can be swapped. Using the system IP address to identify a TLOC ensures that a transport end point can always be identified regardless of IP addressing.

The link color represents the type of WAN interfaces on a device. The Cisco SD-WAN solution offers predefined colors, which are assigned in the configuration of the devices. The color can be one of default, 3g, biz-internet, blue, bronze, custom1, custom2, custom3, gold, green, lte, metro-ethernet, mpls, private1, private2, public-internet, red, and silver.

The encapsulation is that used on the tunnel interface. It can be either IPsec or GRE.



The diagram to the right shows a device that has two WAN connections and hence two TLOCs. The system IP address of the router is 1.1.1.1. The TLOC on the left is uniquely identified by the system IP address 1.1.1.1, the color metro-ethernet, and the encapsulation IPsec, and it maps to the physical WAN interface with the IP address 184.168.0.69. The TLOC on the right is uniquely identified by the system IP address 1.1.1.1, the color biz-internet, and the encapsulation IPsec, and it maps to the WAN IP address 75.1.1.1.

You configure some of the TLOC attributes, including the system IP address, color, and encapsulation, and you can modify some of them by provisioning control policy on the Cisco vSmart Controller. See *Centralized Control Policy*.

### Service Routes

Service routes represent services that are connected to a Cisco vEdge device or to the local-site network in which the Cisco vEdge device resides. The Cisco vEdge device advertises these routes to Cisco vSmart Controllers using service address family NLRI. See *Service Chaining*.

## OMP Route Redistribution

OMP automatically redistributes the following types of routes that it learns either locally or from its routing peers:

- Connected
- Static

- OSPF intra-area routes
- OSPF inter-area routes

To avoid routing loops and less than optimal routing, redistribution of following types of routes requires explicit configuration:

- BGP
- OSPF external routes

To avoid propagating excessive routing information from the edge to the access portion of the network, the routes that devices receive via OMP are not automatically redistributed into the other routing protocols running on the routers. If you want to redistribute the routes received via OMP, you must enable this redistribution locally on each device.

OMP sets the origin and sub-origin type in each OMP route to indicate the route's origin (see the table below). When selecting routes, the Cisco vSmart Controllerand the router take the origin type and subtype into consideration.

*Table 2:*

| OMP Route Origin Type | OMP Route Origin Subtype |
|---|---|
| BGP | External Internal |
| Connected | — |
| OSPF | External-1 External-2 Intra-area Inter-area |
| Static | — |

OMP also carries the metric of the original route. A metric of 0 indicates a connected route.

### Administrative Distance

Administrative distance is the measure used to select the best path when there are two or more different routes to the same destination from multiple routing protocols. When the Cisco vSmart Controller or the router is selecting the OMP route to a destination, it prefers the one with the lower or lowest administrative distance value.

The following table lists the default administrative distances used by the Cisco SD-WAN devices:

*Table 3:*

| Protocol | Administrative Distance |
|---|---|
| Connected | 0 |
| Static | 1 |
| NAT (NAT and static routes cannot coexist in the same VPN; NAT overwrites static routes) | 1 |
| Learned from DHCP | 1 |

| Protocol | Administrative Distance |
|---|---|
| GRE | 5 |
| EBGP | 20 |
| OSPF | 110 |
| IBGP | 200 |
| OMP | 250 |

### OMP Best-Path Algorithm and Loop Avoidance

Cisco SD-WAN devices advertise their local routes to the Cisco vSmart Controller using OMP. Depending on the network topology, some routes might be advertised from multiple devices. Cisco SD-WAN devices use the following algorithm to choose the best route:

1. Select an ACTIVE route. An ACTIVE route is preferred over a STALE route. An active route is a route from a peer with which an OMP session is UP. A stale route is a route from a peer with which an OMP session is in Graceful Restart mode.
2. Check whether the OMP route is valid. If not, ignore it.
3. If the OMP route is valid and if it has been learned from the same Cisco SD-WAN device, select the OMP route with the lower administrative distance.
4. If the administrative distances are equal, select the OMP route with the higher OMP route preference value.
5. If the OMP route preference values are equal, select the OMP route with the higher TLOC preference value.
6. If the TLOC preference values are equal, compare the origin type, and select one in the following order (select the first match): Connected Static EBGP OSFP intra-area OSPF inter-area OSPF external IBGP Unknown
7. If the origin type is the same, select the OMP route that has the lower origin metric.
8. If the origin types are the same, select the OMP route with the lower router ID.
9. If the router IDs are equal, a Cisco vEdge device selects the OMP route with the lower private IP address. If a Cisco vSmart Controller receives the same prefix from two different sites and if all attributes are equal, it chooses both of them.

Here are some examples of choosing the best route:

- A Cisco vSmart Controller receives an OMP route to 10.10.10.0/24 via OMP from a Cisco vEdge device Cisco XE SD-WAN device with an origin code of OSPF, and it also receives the same route from another Cisco vSmart Controller, also with an origin code of OSPF. If all other things are equal, the best-path algorithm chooses the route that came from the Cisco vEdge device.
- A Cisco vSmart Controller learns the same OMP route, 10.10.10.0/24, from two Cisco vEdge devicesin the same site. If all other parameters are the same, both routes are chosen and advertised to other OMP peers. By default, up to four equal-cost routes are selected and advertised.

A Cisco vEdge device installs an OMP route in its forwarding table (FIB) only if the TLOC to which it points is active. For a TLOC to be active, an active BFD session must be associated with that TLOC. BFD sessions are established by each device which creates a separate BFD session with each of the remote TLOCs. If a BFD session becomes inactive, the Cisco vSmart Controller removes from the forwarding table all the OMP routes that point to that TLOC.

## OMP Graceful Restart

Graceful restart for OMP allows the data plane in the Cisco SD-WAN overlay network to continue functioning if the control plane stops functioning or becomes unavailable. With graceful restart, if the vSmart controller in the network goes down, or if multiple vSmart controllers go down simultaneously, Cisco XE SD-WAN devices and Cisco vEdge devices can continue forwarding data traffic. They do this using the last known good information that they received from the vSmart controller. When a vSmart controller is again available, its DTLS connection to the device is re-established, and the device then receives updated, current network information from the vSmart controller.

When OMP graceful restart is enabled, Cisco XE SD-WAN devices and Cisco vEdge devicesand a vSmart controller (that is, two OMP peers) cache the OMP information that they learn from their peer. This information includes OMP routes, TLOC routes, service routes, IPsec SA parameters, and centralized data policies. When one of the OMP peers is no longer available, the other peer uses the cached information to continue operating in the network. So, for example, when a device no longer detects the presence of the OMP connection to a vSmart controller, the device continues forwarding data traffic using the cached OMP information. The device also periodically checks whether the vSmart controller has again become available. When it does come back up and the device re-establishes a connection to it, the device flushes its local cache and considers only the new OMP information from the vSmart controller to be valid and reliable. This same scenario occurs when a vSmart controller no longer detects the presence of Cisco XE SD-WAN devices and Cisco vEdge devices.

# BGP and OSPF Routing Protocols

The Cisco SD-WAN overlay network supports BGP and OSPF unicast routing protocols. These protocols can be configured on Cisco XE SD-WAN devices and Cisco vEdge devices in any VPN except for VPN 0 and VPN 512 to provide reachability to networks at their local sites. Cisco XE SD-WAN devices and Cisco vEdge devices can redistribute route information learned from BGP and OSPF into OMP so that OMP can better choose paths within the overlay network.

When the local site connects to a Layer 3 VPN MPLS WAN cloud, Cisco XE SD-WAN devices and Cisco vEdge devices act as an MPLS CE device and establishes a BGP peering session to connect to the PE router in the L3VPN MPLS cloud.

When Cisco XE SD-WAN devices and Cisco vEdge devices at a local site do not connect directly to the WAN cloud but are one or more hops from the WAN and connect indirectly through a non-Cisco SD-WAN device, standard routing must be enabled on the devices' DTLS connections so that they can reach the WAN cloud. Either OSPF or BGP can be the routing protocol.

In both these types of topologies, the BGP or OSPF sessions run over a DTLS connection created on the loopback interface in VPN 0, which is the tranport VPN that is responsible for carrying control traffic in the overlay network. The Cisco vBond Orchestrator learns about this DTLS connection via the loopback interface and conveys this information to the Cisco vSmart Controller so that it can track the TLOC-related information. In VPN 0, you also configure the physical interface that connects the Cisco vEdge device to its neighbor—either the PE router in the MPLS case or the hub or next-hop router in the local site—but you do not establish a DTLS tunnel connection on that physical interface.

# Configure Unicast Overlay Routing

This topic describes how to provision unicast overlay routing.

### Service-Side Routing

Provisioning BGP and OSPF enables routing on the service side of the network.

To set up routing on a Cisco vEdge device, you provision one VPN or multiple VPNs if segmentation is required. Within each VPN, you configure the interfaces that participate in that VPN and the routing protocols that operate in that VPN.

Because Cisco vSmart Controllers never participate in a local site network, you never configure BGP or OSPF on these devices.

### Transport-Side Routing

To enable communication between Cisco SD-WAN devices, you configure OSPF or BGP on a loopback interface in VPN 0. The loopback interface is a virtual transport interface that is the terminus of the DTLS and IPsec tunnel connections required for  Cisco XE SD-WAN devices and Cisco vEdge devices to participate in the overlay network.

To configure service-side and  transport-side BGP using vManage, see the *Configure BGP using vManage* . To configure service-side and  transport-side BGP using CLI, see the *Configure BGP Using CLI* topic.

# Configure BGP Using vManage Templates

The Border Gateway Protocl (BGP) can be used for service-side routing to provide reachability to networks at the local site, and it can be used for transport-side routing to enable communication between Cisco SD-WAN devices when a device is not directly connected to the WAN cloud. Create separate BGP templates for the two BGP routing types.

To configure the BGP routing protocol using Cisco vManage templates:

1. Create a BGP feature template to configure BGP parameters.
2. Create a VPN feature template to configure VPN parameters for either service-side BGP routing (in any VPN other than VPN 0 or VPN 512) or transport-side BGP routing (in VPN 0).

### Create a BGP Template

1. In vManage, go to **Configuration** > **Templates**.
2. In the Device tab, click **Create Template**.
3. From the Create Template drop-down, select **From Feature Template**.
4. From the Device Model drop-down, select the type of device for which you are creating the template.
5. To create a template for **VPN 0** or **VPN 512**:
    a. Click the **Transport & Management VPN** tab located directly beneath the Description field, or scroll to the Transport & Management VPN section.
    b. Under **Additional VPN 0 Templates**, located to the right of the screen, click **BGP**.

c. From the BGP drop-down, click **Create Template**. The BGP template form displays. The top of the form contains fields for naming the template, and the bottom contains fields for defining BGP parameters.

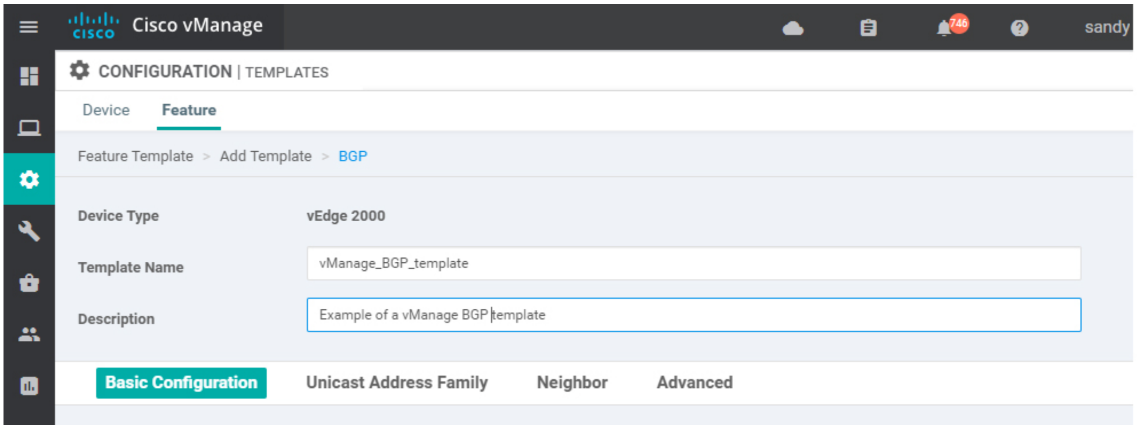
6. To create a template for VPNs **1** through **511**, and **513** through **65530**:

   a. Click the **Service VPN** tab located directly beneath the Description field, or scroll to the Service VPN section.

   b. Click the **Service VPN** drop-down.

   c. Under **Additional VPN Templates**, located to the right of the screen, click **BGP**.

   d. From the BGP drop-down, click **Create Template**. The BGP template form displays. The top of the form contains fields for naming the template, and the bottom contains fields for defining BGP parameters.



Device Type: vEdge 2000

Template Name: vManage_BGP_template

Description: Example of a vManage BGP template

7. In the **Template Name** field, enter a name for the template. The name can be up to 128 characters and can contain only alphanumeric characters.

8. In the **Template Description** field, enter a description of the template. The description can be up to 2048 characters and can contain only alphanumeric characters.

### Configure Basic BGP Parameters

To configure Border Gateway Protocol (BGP), select the **Basic Configuration** tab and configure the following parameters. Parameters marked with an asterisk are required to configure BGP.

| Parameter Name | Description |
|---|---|
| **Shutdown*** | Click **No** to enable BGP on the interface. |
| **AS number*** | Enter the local AS number. |
| **Router ID** | Enter the BGP router ID in decimal four-part dotted notation. |
| **Propagate AS Path** | Click **On** to carry BGP AS path information into OMP. |

| Parameter Name | Description |
|---|---|
| **Internal Routes Distance** | Enter a value to apply as the BGP route administrative distance for routes coming from one AS into another.<br>Range: 0 through 255<br>Default: 0 |
| **Local Routes Distance** | Specify the BGP route administrative distance for routes within the local AS. By default, a route received locally from BGP is preferred over a route received from OMP.<br>Range: 0 through 255<br>Default: 0 |
| **External Routes Distance** | Specify the BGP route administrative distance for routes learned from other sites in the overlay network.<br>Range: 0 through 255<br>Default: 0 |

For service-side BGP, you might want to configure Overlay Management Protocol (OMP) to advertise to the Cisco vSmart Controller any BGP routes that the device learns. By default, Cisco SD-WAN devices advertise to OMP both the connected routes on the device and the static routes that are configured on the device, but it does not advertise BGP external routes learned by the device. You configure this route advertisement in the OMP template for devices or Cisco SD-WAN software. See OMP.

For transport-side BGP, you must also configure a physical interface and a loopback interface in VPN 0. In addition, you should create a policy for BGP to advertise the loopback interface address to its neighbors, and apply the policy in the BGP instance or to a specific neighbor.

To save the feature template, click **Save**.

### Configure Unicast Address Family

To configure global BGP address family information, select the **IPv4 Unicast Address Family** tab and configure the following parameters:

| Tab/Parameter | Option | Sub-Option | Description |
|---|---|---|---|
| **IPv4 / IPv6** | Click **IPv4** to configure an IPv4 VPN interface. Click **IPv6** to configure an IPv6 interface. | | |
| **Maximum Paths** | Specify the maximum number of parallel IBGP paths that can be installed into a route table to enable IBGP multipath load sharing.<br>Range: 0 to 32 | | |
| **Mark as Optional Row** | Check **Mark as Optional Row** to mark this configuration as device-specific. To include this configuration for a device, enter the requested variable values when you attach a device template to a device, or create a template variables spreadsheet to apply the variables. | | |

| Tab/Parameter | Option | Sub-Option | Description |
|---|---|---|---|
| **Redistribute** | Click **Redistribute** > **New Redistribute**. | | |
| | **Mark as Optional Row** | Check **Mark as Optional Row** to mark this configuration as device-specific. To include this configuration for a device, enter the requested variable values when you attach a device template to a device, or create a template variables spreadsheet to apply the variables. | |
| | **Protocol** | Select the protocols from which to redistribute routes into BGP, for all BGP sessions. Options are: | |
| | | **static** | Redistribute static routes into BGP. |
| | | **connected** | Redistribute connected routes into BGP. |
| | | **ospf** | Redistribute Open Shortest Path First routes into BGP. |
| | | **omp** | Redistribute Overlay Management Protocol routes into BGP. |
| | | **nat** | Redistribute Network Address Translation routes into BGP. |
| | | **natpool-outside** | Redistribute outside NAT routes into BGP. |
| | | At a minimum, select the following:<br>• For service-side BGP routing, select **OMP**. By default, OMP routes are not redistributed into BGP.<br>• For transport-side BGP routing, select **Connected**, and then under **Route Policy**, specify a route policy that has BGP advertise the loopback interface address to its neighbors. | |
| | **Route Policy** | Enter the name of the route policy to apply to redistributed routes. | |
| | Click **Add** to save the redistribution information. | | |
| **Network** | Click **Network** > **New Network**. | | |
| | **Mark as Optional Row** | Check **Mark as Optional Row** to mark this configuration as device-specific. To include this configuration for a device, enter the requested variable values when you attach a device template to a device, or create a template variables spreadsheet to apply the variables. | |
| | **Network Prefix** | Enter a network prefix, in the format *prefix/length* to be advertised by BGP. | |
| | Click **Add** to save the network prefix. | | |

| Tab/Parameter | Option | Sub-Option | Description |
|---|---|---|---|
| **Aggregate Address** | Click **Aggregate Address** > **New Aggregate Address**. | | |
| | **Mark as Optional Row** | Check **Mark as Optional Row** to mark this configuration as device-specific. To include this configuration for a device, enter the requested variable values when you attach a device template to a device, or create a template variables spreadsheet to apply the variables. | |
| | **Aggregate Prefix**<br>**IPv6 Aggregate Prefix** | Enter the prefix of the addresses to aggregate for all BGP sessions in the format *prefix/length*. | |
| | **AS Set Path** | Click **On** to generate set path information for the aggregated prefixes. | |
| | **Summary Only** | Click **On** to filter out more specific routes from BGP updates. | |
| | Click **Add** to save the aggregate address. | | |

To save the feature template, click **Save**.

### Configure BGP Neighbors

To configure a neighbor, click **Neighbor** > **New Neighbor**, and configure the following parameters:

**Note** For BGP to function, you must configure at least one neighbor.

| Parameter Name | Options | Sub-Options | Description |
|---|---|---|---|
| **IPv4 / IPv6** | Click **IPv4** to configure IPv4 neighbors. Click **IPv6** to configure IPv6 neighbors. | | |
| **Address/IPv6 Address** | Specify the IP address of the BGP neighbor. | | |
| **Description** | Enter a description of the BGP neighbor. | | |
| **Remote AS** | Enter the AS number of the remote BGP peer. | | |

| Parameter Name | Options | Sub-Options | Description |
|---|---|---|---|
| **Address Family** | Click **On** and select the address family. Currently, the software supports only the BGP IPv4 unicast address family. Enter the address family information. | | |
| | **Address Family** | Select the address family. Currently, the software supports only the BGP IPv4 unicast address family. | |
| | **Maximum Number of Prefixes** | Specify the maximum number of prefixes that can be received from the neighbor.<br>Range: 1 through 4294967295<br>Default: 0 | |
| | | **Threshold** | Threshold at which to generate a warning message or restart the BGP connection. The threshold is a percentage of the maximum number of prefixes. You can specify either a restart interval or a warning only. |
| | | **Restart Interval** | How long to wait to restart the BGP connection.*Range:* 1 through 65535 minutes |
| | | **Warning Only** | Click **On** to display a warning message only without restarting the BGP connection. |
| | | **Route Policy In** | Click **On** and specify the name of a route policy to apply to prefixes received from the neighbor. |
| | | **Route Policy Out** | Click **On** and specify the name of a route policy to apply to prefixes sent to the neighbor. |
| **Shutdown** | Click **On** to enable the connection to the BGP neighbor. | | |

### Configure Advanced Neighbor Parameter

To configure advanced parameters for the neighbor, click **Neighbor** > **Advanced Options**.

| Parameter Name | Description |
|---|---|
| **Next-Hop Self** | Click **On** to configure the router to be the next hop for routes advertised to the BGP neighbor. |
| **Send Community** | Click **On** to send the local router's BGP community attribute to the BGP neighbor. |
| **Send Extended Community** | Click **On** to send the local router's BGP extended community attribute to the BGP neighbor. |
| **Negotiate Capability** | Click **On** to allow the BGP session to learn about the BGP extensions that are supported by the neighbor. |
| **Source Interface Address** | Enter the IP address of a specific interface of the neighbor that BGP is to use for the TCP connection to the neighbor. |
| **Source Interface Name** | Enter the name of a specific interface of the neighbor that BGP is to use for the TCP connection to the neighbor, in the format **ge** *port*/*slot*. |

| Parameter Name | Description |
|---|---|
| **EBGP Multihop** | Set the time to live (TTL) for BGP connections to external peers.<br>Range: 0 to 255<br>Default: 1 |
| **Password** | Enter a password to use to generate an MD5 message digest. Configuring the password enables MD5 authentication on the TCP connection with the BGP peer. The password is case-sensitive and can be up to 25 characters long. It can contain any alphanumeric characters, including spaces. The first character cannot be a number. |
| **Keepalive Time** | Specify the frequency at which keepalive messages are advertised to a BGP peer. These messages indicate to the peer that the local router is still active and should be considered available. Specify the keepalive time for the neighbor to override the global keepalive time.<br>Range: 0 through 65535 seconds<br>Default: 60 seconds (one-third the hold-time value) |
| **Hold Time** | Specify the interval after not receiving a keepalive message that the local BGP session considers its peer to be unavailable. The local router then terminates the BGP session to that peer. Specify the hold time for the neighbor to override the global hold time.<br>Range: 0 through 65535 seconds<br>Default: 180 seconds (three times the keepalive timer) |
| **Connection Retry Time** | Specify the number of seconds between retries to establish a connection to a configured BGP neighbor peer that has gone down.<br>Range: 0 through 65535 seconds<br>Default: 30 seconds |
| **Advertisement Interval** | For the BGP neighbor, set the minimum route advertisement interval (MRAI) between when BGP routing update packets are sent to that neighbor.<br>Range: 0 through 600 seconds<br>Default: 5 seconds for IBGP route advertisements; 30 seconds for EBGP route advertisements |

To save the feature template, click **Save**.

### Change the Scope of a Parameter Value

When you first open a feature template, for each parameter that has a default value, the scope is set to Default (a ✓), and the default setting or value is shown). To change the default or to enter a value, click the scope drop-down to the left of the parameter field and select one of the following:

| Parameter Name | Description |
|---|---|
| Device Specific | Use a device-specific value for the parameter. For device-specific parameters, you cannot enter a value in the feature template. You enter the value when you attach a device to a device template.<br>When you click **Device Specific**, the Enter Key box opens. This box displays a key which is a unique string that identifies the parameter in a CSV file that you create. This file is an Excel spreadsheet that contains one column for each key. The header row contains the key names (one key per column), and each row after that corresponds to a device and defines the values of the keys for that device. You upload the CSV file when you attach a device to a device template.<br>To change the default key, type a new string and move the cursor out of the Enter Key box.<br>Examples of device-specific parameters are system IP address, hostname, GPS location, and site ID. |
| Global | Enter a value for the parameter, and apply that value to all devices.<br>Examples of parameters that you might apply globally to a group of devices are DNS server, syslog server, and interface MTUs. |

### Configure Advanced BGP Parameters

To configure advanced parameters for BGP, click the **Advanced** tab and configure the following parameters:

| Parameter Name | Description |
|---|---|
| **Hold Time** | Specify the interval after not receiving a keepalive message that the local BGP session considers its peer to be unavailable. The local device then terminates the BGP session to that peer. This hold time is the global hold time.<br>Range: 0 through 65535 seconds<br>Default: 180 seconds (three times the keepalive timer) |
| **Keepalive** | Specify the frequency at which keepalive messages are advertised to a BGP peer. These messages indicate to the peer that the local device is still active and should be considered available. This keepalive time is the global keepalive time.<br>Range: 0 through 65535 seconds<br>Default: 60 seconds (one-third the hold-time value) |
| **Compare MED** | Click **On** to compare the device IDs among BGP paths to determine the active path. |
| **Deterministic MED** | Click **On** to compare multiple exit discriminators (MEDs) from all routes received from the same AS, regardless of when the route was received. |
| **Missing MED as Worst** | Click **On** to consider a path as the worst path if the path is missing a MED attribute. |
| **Compare Router ID** | Click **On** to always compare MEDs regardless of whether the peer ASs of the compared routes are the same. |

| Parameter Name | Description |
|---|---|
| **Multipath Relax** | Click **On** to have the BGP best-path process select from routes in different in ASs. By default, when you are using BGP multipath, the BGP best path process selects from routes in the same AS to load-balance across multiple paths. |

To save the feature, click **Save**.

# Configure BGP Using CLI

### Verify BGP Configuration

This topic describes how to configure BGP for service-side and transport-side for unicast overlay routing

### Configure Service-Side Routing

To set up routing on the Cisco vEdge device, you provision one VPN or multiple VPNs if segmentation is required. Within each VPN, you configure the interfaces that participate in that VPN and the routing protocols that operate in that VPN.

1. Configure a VPN.

   ```
   vEdge(config)# vpn vpn-id
   ```

   *vpn-id* can be any service-side VPN, which is a VPN other than VPN 0 and VPN 512. VPN 0 is the transport VPN and carries only control traffic, and VPN 512 is the management VPN.

2. Configure BGP to run in the VPN:

   a. Configure the local AS number:

      ```
      vEdge(config-vpn)# router bgp local-as-number
      ```

      You can specify the AS number in 2-byte ASDOT notation (1 through 65535) or in 4-byte ASDOT notation (1.0 through 65535.65535).

   b. Configure the BGP peer, specifying its address and AS number (the remote AS number), and enable the connection to the peer:

      ```
      vEdge(config-bgp)# neighbor address remote-as remote-as-number
      vEdge(config-bgp)# no shutdown
      ```

3. Configure a system IP address for the Cisco vEdge device:

   ```
   vEdge(config)# system system-ipaddress
   ```

### Example of BGP Configuration on a vEdge Router

```
vEdge# show running-config system
system
  system-ip 10.1.2.3
!
vEdge# show running-config vpn 1
vpn 1
  router
    bgp 1
      neighbor 11.1.2.3
        no shutdown
        remote-as 2
```

```
      !
    !
  !
  ip route 0.0.0.0/0 10.0.16.13
!
```

### Redistribute BGP Routes and AS Path Information

By default, routes from other routing protocols are not redistributed into BGP. It can be useful for BGP to learn OMP routes, because OMP learns routes to destinations throughout the overlay network. BGP on the Cisco vEdge device then advertises the OMP routes to all the BGP routers in the service-side of the network.

```
Device(config)# vpn vpn-id router bgp
vEdge(config-bgp)# address-family ipv4-unicast redistribute omp [route-policy policy-name]
```

You can also redistribute routes learned from other protocols into BGP:

```
Device(config-bgp)# address-family ipv4-unicast redistribute (connected | nat |
natpool-outside | ospf | static) [route-policy policy-name]
```

You can control redistribution of routes on a per-neighbor basis:

```
vEdge(config-bgp)# neighbor ip-address
vEdge(config-neighbor)# address-family ipv4-unicast redistribute (connected | nat |
natpool-outside | omp | ospf | static)
vEdge(config-neighbor)# route-policy policy-name (in | out)
```

In the BGP route redistribution commands, the optional route policy is applied to the routes that are redistributed into BGP or routes that are redistributed out from BGP.

You can configure the Cisco vEdge device to advertise BGP routes that it has learned, through OMP, from the Cisco vSmart Controller. Doing so allows the Cisco vSmart Controller to advertise these routes to other Cisco vEdge devices in the overlay network. You can advertise BGP routes either globally or for a specific VPN:

```
vEdge(config)# omp advertise bgp

vEdge(config)# vpn vpn-id omp advertise bgp
```

### BGP Route Advertisements

By default, when BGP advertises routes into OMP, BGP advertises each prefix's metric. BGP can also advertise the prefix's AS path:

```
Device(config)# vpn vpn-id router bgp
vEdge(config-bgp)# propagate-aspath
```

When you configure BGP to propagate AS path information, the router sends AS path information to routers that are behind the vEdge router (in the service-side network) that are running BGP, and it receives AS path information from these routers. If you are redistributing BGP routes into OMP or into another protocol, or if you are advertising BGP routes to OMP, the AS path information is included in the advertised BGP routes. If you configure BGP AS path propagation on some but not all vEdge routers in the overlay network, the routers on which it is not configured receive the AS path information but they do not forward it to the BGP routers in their local service-side network. Propagating AS path information can help to avoid BGP routing loops.

In networks that have both overlay and underlay connectivity—for example, when vEdge routers are interconnected by both a Cisco SD-WAN overlay network and an MPLS underlay network—you can assign an AS number to OMP itself. For vEdge routers running BGP, this overlay AS number is included in the AS path of BGP route updates. To configure the overlay AS:

```
Device(config)# omp
vEdge(omp)# overlay-as as-number
```

You can specify the AS number in 2-byte ASDOT notation (1 through 65535) or in 4-byte ASDOT notation (1.0 through 65535.65535). As a best practice, it is recommended that the overlay AS number be a unique AS number within both the overlay and the underlay networks. That use, select an AS number that is not used elsewhere in the network.

If you configure the same overlay AS number on multiple vEdge routers in the overlay network, all these routers are considered to be part of the same AS, and as a result, they do not forward any routes that contain the overlay AS number. This mechanism is an additional technique for preventing BGP routing loops in the network.

### Configure Transport-Side Routing

To configure transport-side routing, you configure a loopback interface, the physical interface, and the routing protocol in VPN 0.

1. Configure a physical interface in VPN 0:

   ```
   Device(config)# vpn 0 interface geslot/port ip address address
   vedge(config-interface)# no shutdown
   ```

2. Configure a loopback interface in VPN 0:

   ```
   Device(config)# vpn 0 interface loopbacknumber ip address address
   Device(config-interface)# no shutdown
   Device(config-interface)# tunnel-interface color color
   ```

3. Configure a BGP instance in VPN 0:

   ```
   Device(config)# vpn 0 router bgp local-as-number
   ```

4. Create a policy for BGP to advertise the loopback interface address to its neighbors:

   ```
   vEdge(config)# policy lists prefix-list prefix-list-name ip-prefix prefix
   prefix is the IP address of the loopback interface.
   ```

   *prefix* is the IP address of the loopback interface.

5. Configure a route policy that affects the loopback interface's prefix:

   ```
   Device(config)# policy route-policy policy-name sequence number match address
   prefix-list-name
   Device(config)# policy route-policy policy-name sequence number action accept
   Device(config)# policy route-policy policy-name default-action reject
   ```

6. Reference the policy in the BGP instance. To apply the policy such that the loopback address is advertised to all BGP neighbors:

   ```
   Device(config)# vpn 0 router bgp local-as-number address-family ipv4-unicast redistribute
    connected route-policy policy-name
   ```

   To apply the policy only to a specific neighbor:

   ```
   Device(config)# vpn 0 router bgp local-as-number neighbor neighbor-address address-family
    ipv4-unicast redistribute connected route-policy policy-name out
   ```

   Specify **out** in the second command so that BGP advertises the loopback prefix out to the neighbor.

### Example of BGP Transport-Side Configuration

Here is an example of a minimal BGP transport-side routing configuration in which the loopback address is advertised to all the vEdge router's BGP neighbors. Note that even though we did not configure any services on the tunnel interface, these services are associated with the tunnel by default and are included in the configuration. Because services affect only physical interfaces, you can ignore them on loopback interfaces.

```
vEdge# show running-config vpn 0
vpn 0
 router
  bgp 2
   router-id 172.16.255.18
   timers
    keepalive 1
    holdtime  3
   !
   address-family ipv4-unicast
    redistribute connected route-policy export_loopback
   !
   neighbor 10.20.25.16
    no shutdown
    remote-as 1
    timers
     connect-retry          2
     advertisement-interval 1
    !
   !
  !
 !
 interface ge0/1
  ip address 10.20.25.18/24
  no shutdown
 !
 interface loopback
  ip address 172.16.255.118/32
  tunnel-interface
   color lte
   allow-service dhcp
   allow-service dns
   allow-service icmp
   no allow-service sshd
   no allow-service ntp
   no allow-service stun
  !
  no shutdown
 !
!
policy
 lists
  prefix-list loopback_prefix
   ip-prefix 172.16.255.118/32
  !
 !
 route-policy export_loopback
  sequence 10
   match
    address loopback_prefix
   !
   action accept
   !
  !
  default-action reject
 !
!
```

# Configure OSPF Using vManage Templates

Use the OSPF template for all Cisco SD-WAN devices.

To configure OSPF on a device using Cisco vManage templates:

1. Create an OSPF feature template to configure OSPF parameters. OSPF can be used for service-side routing to provide reachability to networks at the local site, and it can be used for transport-side routing to enable communication between the Cisco SD-WAN devices when the router is not directly connected to the WAN cloud. Create separate OSPF templates for the two OSPF routing types.

2. Create a VPN feature template to configure VPN parameters for either service-side OSPF routing (in any VPN other than VPN 0 or VPN 512) or transport-side OSPF routing (in VPN 0). See the VPN help topic for more information.

### Create an OSPF Template

1. In vManage NMS, select **Configuration** > **Templates**.

2. In the Device tab, click **Create Template**.

3. From the Create Template drop-down, select **From Feature Template**.

4. From the Device Model drop-down, select the type of device for which you are creating the template. To create a template for VPN 0 or VPN 512:

   a. Click the **Transport & Management VPN** tab located directly beneath the Description field, or scroll to the Transport & Management VPN section.

   b. Under Additional VPN 0 Templates, located to the right of the screen, click **OSPF**.

   c. From the OSPF drop-down, click **Create Template**. The OSPF template form is displayed. The top of the form contains fields for naming the template, and the bottom contains fields for defining OSPF parameters.

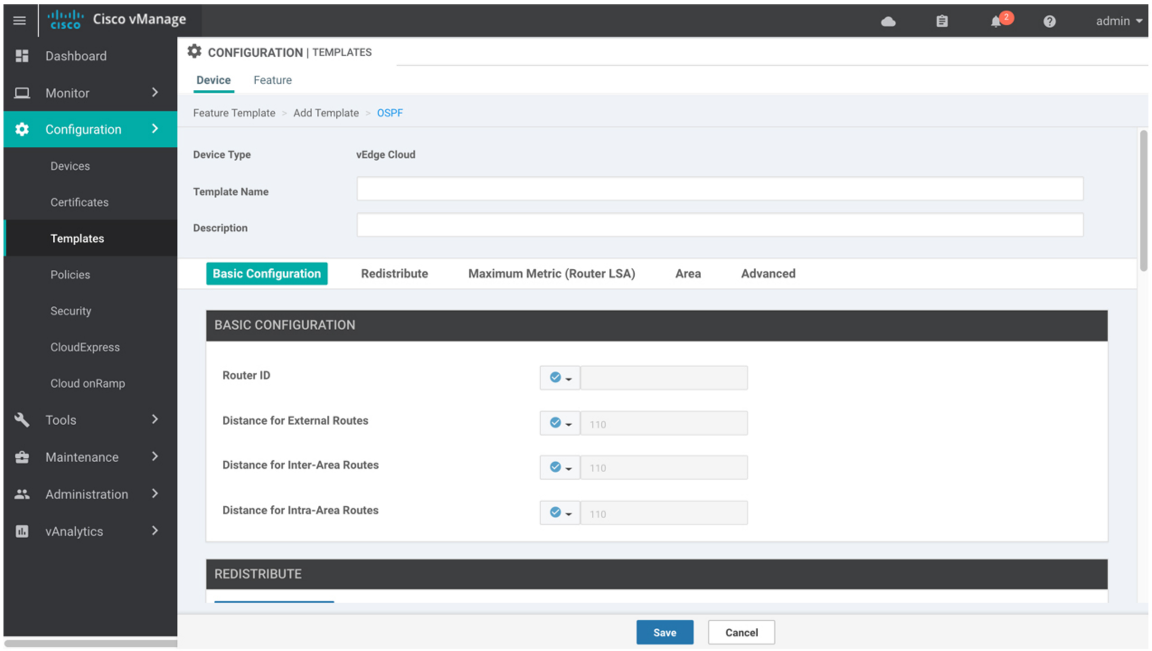
5. To create a template for VPNs 1 through 511, and 513 through 65530:

   a. Click the **Service VPN** tab located directly beneath the Description field, or scroll to the Service VPN section.

   b. Click the **Service VPN** drop-down.

   c. Under Additional VPN Templates, located to the right of the screen, click **OSPF**.

   d. From the OSPF drop-down, click **Create Template**. The OSPF template form is displayed. The top of the form contains fields for naming the template, and the bottom contains fields for defining OSPF parameters.



Device Type: vEdge Cloud

Template Name:

Description:

Router ID:

Distance for External Routes: 110

Distance for Inter-Area Routes: 110

Distance for Intra-Area Routes: 110

6. In the Template Name field, enter a name for the template. The name can be up to 128 characters and can contain only alphanumeric characters.

7. In the Template Description field, enter a description of the template. The description can be up to 2048 characters and can contain only alphanumeric characters.

When you first open a feature template, for each parameter that has a default value, the scope is set to Default (indicated by a check mark), and the default setting or value is shown. To change the default or to enter a value, click the scope drop-down to the left of the parameter field and select one of the following:

*Table 4:*

| Parameter Scope | Scope Description |
|---|---|
| Device Specific (indicated by a host icon) | Use a device-specific value for the parameter. For device-specific parameters, you cannot enter a value in the feature template. You enter the value when you attach a Cisco SD-WAN device to a device template .<br>When you click **Device Specific**, the Enter Key box opens. This box displays a key,which is a unique string that identifies the parameter in a CSV file that you create. This file is an Excel spreadsheet that contains one column for each key. The header row contains the key names (one key per column), and each row after that corresponds to a device and defines the values of the keys for that device. You upload the CSV file when you attach a Cisco SD-WAN device to a device template. For more information, see *Create a Template Variables Spreadsheet* .<br>To change the default key, type a new string and move the cursor out of the Enter Key box.<br>Examples of device-specific parameters are system IP address, hostname, GPS location, and site ID. |

| Parameter Scope | Scope Description |
|---|---|
| Global (indicated by a globe icon) | Enter a value for the parameter, and apply that value to all devices.<br>Examples of parameters that you might apply globally to a group of devices are DNS server, syslog server, and interface MTUs. |

### Configure Basic OSPF

To configure basic OSPF, select the **Basic Configuration** tab and then configure the following parameters. All these parameters are optional. For OSPF to function, you must configure area 0, as described below.

*Table 5:*

| Parameter Name | Description |
|---|---|
| Router ID | Enter the OSPF router ID in decimal four-part dotted notation. This is the IP address associated with the router for OSPF adjacencies. |
| Distance for External Routes | Specify the OSPF route administration distance for routes learned from other domains.<br>*Range:* 0 through 255*Default:* 110 |
| Distance for Inter-Area Routes | Specify the OSPF route administration distance for routes coming from one area into another.<br>*Range:* 0 through 255*Default:* 110 |
| Distance for intra-Area routes | Specify the OSPF route administration distance for routes within an area.<br>*Range:* 0 through 255*Default:* 110 |

To save the feature template, click **Save**.

### Redistribute Routes into OSPF

To redistribute routes learned from other protocols into OSPF on Cisco SD-WAN devices, select **Redistribute** > **Add New Redistribute** and configure the following parameters:

*Table 6:*

| Parameter Name | Description |
|---|---|
| Protocol | Select the protocol from which to redistribute routes into OSPF. Select from BGP, Connected, NAT, OMP, and Static. |
| Route Policy | Enter the name of a localized control policy to apply to routes before they are redistributed into OSPF. |

To add another OSPF route redistribution policy, click the plus sign (+).

To remove an OSPF route redistribution policy from the template configuration, click **the trash icon** to the right of the entry.

To save the feature template, click **Save**.

### Configure OSPF To Advertise a Maximum Metric

To configure OSPF to advertise a maximum metric so that other devices do not prefer the Cisco vEdge device as an intermediate hop in their Shortest Path First (SPF) calculation, select **Maximum Metric (Router LSA)** > **Add New Router LSA** and configure the following parameters:

*Table 7:*

| Parameter Name | Description |
|---|---|
| Type | Select a type:<br>• Administrative—Force the maximum metric to take effect immediately through operator intervention.<br>• On-Startup—Advertise the maximum metric for the specified time. |
| Advertisement Time | If you selected On-Startup, specify the number of seconds to advertise the maximum metric after the router starts up.<br>*Range:* 0, 5 through 86400 seconds*Default:* 0 seconds (the maximum metric is advertised immediately when the router starts up) |

To save the feature template, click **Save**.

### Configure OSPF Areas

To configure an OSPF area within a VPN on a Cisco SD-WAN device, select **Area** > **Add New Area**. For OSPF to function, you must configure area 0.

*Table 8:*

| Parameter Name | Description |
|---|---|
| Area Number | Enter the number of the OSPF area.<br>*Range:* 32-bit number |
| Set the Area Type | Select the type of OSPF area, Stub or NSSA. |
| No Summary | Select **On** to not inject OSPF summary routes into the area. |
| Translate | If you configured the area type as NSSA, select when to allow Cisco SD-WAN devices that are ABRs (area border routers) to translate Type 7 LSAs to Type 5 LSAs:<br>• Always—Router always acts as the translator for Type 7 LSAs. That is no other router, even if it is an ABR, can be the translator. If two ABRs are configured to always be the translator, only one of them actually ends up doing the translation.<br>• Candidate—Router offers translation services, but does not insist on being the translator.<br>• Never—Translate no Type 7 LSAs. |

To save the new area, click **Add**.

To save the feature template, click **Save**.

### Configure Interfaces in an OSPF Area

To configure the properties of an interface in an OSPF area, select **Area** > **Add New Area** > **Add Interface**. In the Add Interface popup, configure the following parameters:

*Table 9:*

| Parameter Name | Description |
|---|---|
| Interface Name | Enter the name of the interface, in the format **ge** *slot*/*port* or **loopback** *number*. |
| Hello Interval | Specify how often the router sends OSPF hello packets.<br>*Range:* 1 through 65535 seconds*Default:* 10 seconds |
| Dead Interval | Specify how often the Cisco vEdge device must receive an OSPF hello packet from its neighbor. If no packet is received, the Cisco vEdge deviceassumes that the neighbor is down.<br>*Range:* 1 through 65535 seconds*Default:* 40 seconds (4 times the default hello interval) |
| LSA Retransmission Interval | Specify how often the OSPF protocol retransmits LSAs to its neighbors.<br>*Range:* 1 through 65535 seconds*Default:* 5 seconds |
| Interface Cost | Specify the cost of the OSPF interface.<br>*Range:* 1 through 65535 |

To configure advanced options for an interface in an OSPF area, in the Add Interface popup, click **Advanced Options** and configure the following parameters:

*Table 10:*

| Parameter Name | Description |
|---|---|
| Designated Router Priority | Set the priority of the router to be elected as the designated router (DR). The router with the highest priority becomes the DR. If the priorities are equal, the node with the highest router ID becomes the DR or the backup DR.*Range:* 0 through 255*Default:* 1 |
| OSPF Network Type | Select the OSPF network type to which the interface is to connect:<br>• Broadcast network—WAN or similar network.<br>• Point-to-point network—Interface connects to a single remote OSPF router.<br>*Default:* Broadcast |
| Passive Interface | Select **On** or **Off** to specify whether to set the OSPF interface to be passive. A passive interface advertises its address, but does not actively run the OSPF protocol.*Default:* Off |

| Parameter Name | Description |
|---|---|
| Authentication | Specify the authentication and authentication key on the interface to allow OSPF to exchange routing update information securely. |
| • Authentication Type | Select the authentication type:<br>• Simple authentication—Password is sent in clear text.<br>• Message-digest authentication—MD5 algorithm generates the password. |
| • Authentication Key | Enter the authentication key. Plain text authentication is used when devices within an area cannot support the more secure MD5 authentication. The key can be 1 to 32 characters. |
| Message Digest | Specify the key ID and authentication key if you are using message digest (MD5). |
| • Message Digest Key ID | Enter the key ID for message digest (MD5 authentication). It can be 1 to 32 characters. |
| • Message Digest Key | Enter the MD5 authentication key in clear text or as an AES-encrypted key. It can be from 1 to 255 characters. |

To save the interface configuration, click **Save**.

To save the new area, click **Add**.

To save the feature template, click **Save**.

### Configure an Interface Range for Summary LSAs

To configure the properties of an interface in an OSPF area, select **Area** > **Add New Area** > **Add Range**. In the Area Range popup, click **Add Area Range**, and configure the following parameters:

***Table 11:***

| Parameter Name | Description |
|---|---|
| Address | Enter the IP address and subnet mask, in the format *prefix*/*length* for the IP addresses to be consolidated and advertised. |
| Cost | Specify a number for the Type 3 summary LSA. OSPF uses this metric during its SPF calculation to determine the shortest path to a destination.*Range:* 0 through 16777215 |
| No Advertise | Select **On** to not advertise the Type 3 summary LSAs or Off to advertise them. |

To save the area range, click **Save**.

To save the new area, click **Add**.

To save the feature template, click **Save**.

### Configure Other OSPF Properties

To configure other OSPF properties, select the **Advanced** tab and configure the following properties:

*Table 12:*

| Parameter Name | Description |
|---|---|
| Reference Bandwidth | Specify the reference bandwidth for the OSPF auto-cost calculation for the interface.<br>*Range:* 1 through 4294967 Mbps*Default:* 100 Mbps |
| RFC 1538 Compatible | By default, the OSPF calculation is done per RFC 1583. Select **Off** to calculate the cost of summary routes based on RFC 2328. |
| Originate | Click **On** to generate a default external route into an OSPF routing domain:<br>• Always—Select On to always advertise the default route in an OSPF routing domain.<br>• Default metric—Set the metric used to generate the default route.*Range:* 0 through 16777214*Default:* 10<br>• Metric type—Select to advertise the default route as an OSPF Type 1 external route or an OSPF Type 2 external route. |
| SPF Calculation Delay | Specify the amount of time between when the first change to a topology is received until performing the SPF calculation.<br>*Range*: 0 through 600000 milliseconds (60 seconds)*Default*: 200 milliseconds |
| Initial Hold Time | Specify the amount of time between consecutive SPF calculations.<br>*Range*: 0 through 600000 milliseconds (60 seconds)*Default*: 1000 milliseconds |
| Maximum Hold Time | Specify the longest time between consecutive SPF calculations.<br>*Range*: 0 through 600000*Default*: 10000 milliseconds (60 seconds) |
| Policy Name | Enter the name of a localized control policy to apply to routes coming from OSPF neighbors. |

To save the feature template, click **Save**.

# Configure OSPF Using CLI

This topic describes how to configure basic service-side and transport-side OSPF for Unicast overlay routing.

### Configure Basic Service-Side OSPF

To set up routing on the Cisco vEdge device, you provision one VPN or multiple VPNs if segmentation is required. Within each VPN, you configure the interfaces that participate in that VPN and the routing protocols that operate in that VPN.

To configure basic service-side OSPF functionality:

1. Configure a VPN for the OSPF network:

```
vEdge(config)# vpn vpn-id
```

*vpn-id* can be any VPN number except VPN 0 and VPN512. VPN 0 is the transport VPN and carries only control traffic, and VPN 512 is the management interface.

2. Configure OSPF area 0 and the interfaces that participate in that area:

```
vEdge(config-vpn)# router ospf
vEdge(config-ospf)# area 0
vEdge(config-area-0)# interface  interface-name
vEdge(config-interface)# ip-address  address
vEdge(config-interface)# no shutdown
vEdge (ospf-if)#  exit
```

3. Redistribute OMP routes into OSPF:

```
vEdge(config-ospf)# redistribute omp
```

   By default, OMP routes are not redistributed into OSPF.

4. Repeat Steps 1 through 3 for any additional VPNs.

5. If desired, configure OMP to advertise to the Cisco vSmart Controller any BGP and OSPF external routes that the Cisco vEdge device has learned:

```
vEdge(config)# omp
vEdge(config-omp)# advertise bgp
vEdge(config-omp)# advertise ospf external
```

### Example of Basic Service-Side OSPF Configuration

This configuration sets up VPN 10 with two interfaces, **ge2/0** and **ge3/0**. It enables OSPF routing on those interfaces in area 0, and it redistributes the OMP routes from the Cisco vSmart Controller into OSPF.

```
vpn 10
  router
    ospf
      redistribute omp
      area 0
        interface ge2/0
        exit
      interface ge3/0
       exit
     exit
    !
  !
  interface ge2/0
    ip address 10.0.5.12/24
    no shutdown
  !
  interface ge3/0
    ip address 10.0.2.12/24
    no shutdown
  !
```

### Configure OSPF Transport-Side Routing

To configure transport-side routing, you configure a loopback interface, the physical interface, and the routing protocol in VPN 0.

To configure OSPF transport-side routing:

1. Configure a physical interface in VPN 0:

```
vEdge(config)# vpn 0 interface geslot/port ip address address
vEdge(config-interface)# no shutdown
```

2. Configure a loopback interface in VPN 0 as a tunnel interface:

   ```
   vEdge(config)# vpn 0 interface loopbacknumber ip address address
   vEdge(config-interface)# no shutdown
   vEdge(config-interface)# tunnel-interface color color
   ```

3. Configure an OSPF instance in VPN 0:

   ```
   vEdge(config)# vpn 0 router ospf
   ```

4. Add the physical and loopback interfaces to the OSPF area:

   ```
   vEdge(config-ospf)# area number interface geslot/port
   vEdge(config-area)# interface loopbacknumber
   ```

### Example of Transport-Side OSPF Configuration

Here is any example of a minimal OSPF transport-side routing configuration. Note that even though we did not configure any services on the tunnel interface, these services are associated with the tunnel by default and are included in the configuration. Because services affect only physical interfaces, you can ignore them on loopback interfaces.

```
vEdge# show running-config vpn 0
vpn 0
 router
  ospf
   router-id 172.16.255.11
   timers spf 200 1000 10000
   area 0
    interface ge0/1
    exit
    interface loopback1
    exit
   exit
  !
 !
 interface ge0/1
  ip address 10.0.26.11/24
  no shutdown
 !
 interface loopback1
  ip address 10.0.101.1/32
  tunnel-interface
   color lte
   allow-service dhcp
   allow-service dns
   allow-service icmp
   no allow-service sshd
   no allow-service ntp
   no allow-service stun
  !
  no shutdown
 !
!
```

# Configure OMP Using vManage Templates

Use the OMP template to configure OMP parameters for all Cisco vEdge devices, and for Cisco vSmart Controllers.

OMP is enabled by default on all Cisco vEdge devices, Cisco vManage NMSs, and Cisco vSmart Controllers, so there is no need to explicitly enable OMP. OMP must be operational for the Cisco SD-WAN overlay network to function. If you disable it, you disable the overlay network.

**Note**

- Route advertisements in OMP are done either by applying the configuration at the global level or at the specific VPN level. See the Configure OMP Advertisements section in this topic.

### Create OMP Template

1. In Cisco vManage, select **Configuration** > **Templates**.
2. In the Device tab, click **Create Template**.
3. From the Create Template drop-down, select **From Feature Template**.
4. From the Device Model drop-down, select the type of device for which you are creating the template.
5. To create a custom template for OMP, select the Factory_Default_OMP_Template and click **Create Template**. The OMP template form is displayed. The top of the form contains fields for naming the template, and the bottom contains fields for defining OMP parameters. You may need to click a tab or the plus sign (+) to display additional fields.
6. In the Template Name field, enter a name for the template. The name can be up to 128 characters and can contain only alphanumeric characters.
7. In the Template Description field, enter a description of the template. The description can be up to 2048 characters and can contain only alphanumeric characters.

When you first open a feature template, for each parameter that has a default value, the scope is set to Default (indicated by a check mark), and the default setting or value is shown. To change the default or to enter a value, click the scope drop-down to the left of the parameter field and select one of the following:

*Table 13:*

| Parameter Scope | Scope Description |
|---|---|
| Device Specific (indicated by a host icon) | Use a device-specific value for the parameter. For device-specific parameters, you cannot enter a value in the feature template. You enter the value when you attach a Cisco SD-WAN device to a device template .<br>When you click Device Specific, the Enter Key box opens. This box displays a key, which is a unique string that identifies the parameter in a CSV file that you create. This file is an Excel spreadsheet that contains one column for each key. The header row contains the key names (one key per column), and each row after that corresponds to a device and defines the values of the keys for that device. You upload the CSV file when you attach a Cisco SD-WAN device to a device template. For more information, see *Create a Template Variables Spreadsheet* .<br>To change the default key, type a new string and move the cursor out of the Enter Key box.<br>Examples of device-specific parameters are system IP address, hostname, GPS location, and site ID. |

| Parameter Scope | Scope Description |
|---|---|
| Global (indicated by a globe icon) | Enter a value for the parameter, and apply that value to all devices.<br>Examples of parameters that you might apply globally to a group of devices are DNS server, syslog server, and interface MTUs. |

### Configure Basic OMP Options

To configure basic OMP options, select the **Basic Configuration** tab and configure the following parameters. All parameters are optional.

*Table 14:*

| Parameter Name | Description |
|---|---|
| Graceful Restart for OMP | Ensure that Yes is selected to enable graceful restart. By default, graceful restart for OMP is enabled. |
| Overlay AS Number (on vEdge routers only) | Specify a BGP AS number that OMOP advertises to the router's BGP neighbors. |
| Graceful Restart Timer | Specify how often the OMP information cache is flushed and refreshed. A timer value of 0 disables OMP graceful restart.*Range:* 0 through 604800 seconds (168 hours, or 7 days)*Default:* 43200 seconds (12 hours) |
| Number of Paths Advertised per Prefix | Specify the maximum number of equal-cost routes to advertise per prefix. Cisco vEdge devices advertise routes to Cisco vSmart Controllers, and the controllers redistributes the learned routes, advertising each route-TLOC tuple. A Cisco vEdge device can have up to four TLOCs, and by default advertises each route-TLOC tuple to the Cisco vSmart Controller. If a local site has two Cisco vEdge devices, a Cisco vSmart Controller could potentially learn eight route-TLOC tuples for the same route. If the configured limit is lower than the number of route-TLOC tuples, the best route or routes are advertised.*Range:* 1 through 16*Default:* 4 |
| ECMP Limit (on vEdge routers only) | Specify the maximum number of OMP paths received from the Cisco vSmart Controller that can be installed in the Cisco vEdge device'slocal route table. By default, a Cisco vEdge device installs a maximum of four unique OMP paths into its route table.*Range:* 1 through 32*Default:* 4 |
| Send Backup Paths (on vSmart Controllers only) | Click **On** to have OMP advertise backup routes to Cisco vEdge devices. By default, OMP advertises only the best route or routes. If you configure to send backup paths, OMP also advertises the first non-best route in addition to the best route or routes. |
| Shutdown | Ensure that **No** is selected to enable to Cisco SD-WAN overlay network. Click **Yes** to disable OMP and disable the Cisco SD-WAN overlay network. OMP is enabled by default. |
| Discard rejected (on vSmart controllers only) | Click **Yes** to have OMP discard routes that have been rejected on the basis of policy. By default, rejected routes are not discarded. |

To save the feature template, click Save.

### Configure OMP Timers

To configure OMP timers, select the **Timers** tab and configure the following parameters:

*Table 15:*

| Parameter Name | Description |
|---|---|
| Advertisement Interval | Specify the time between OMP Update packets.<br>*Range:* 0 through 65535 seconds*Default:* 1 second |
| Hold Time | Specify how long to wait before closing the OMP connection to a peer. If the peer does not receive three consecutive keepalive messages within the hold time, the OMP connection to the peer is closed.*Range:* 0 through 65535 seconds*Default:* 60 seconds |
| EOR Timer | Specify how long to wait after an OMP session has gone down and then come back up to send an end-of-RIB (EOR) marker. After this marker is sent, any routes that were not refreshed after the OMP session came back up are considered to be stale and are deleted from the route table.*Range:* 1 through 3600 seconds (1 hour)*Default:* 300 seconds (5 minutes) |

To save the feature template, click **Save**.

### Configure OMP Advertisements

**Note** Route advertisements in OMP are done either by applying the configuration at the global level or at the specific VPN level.

To advertise routes learned locally by the Cisco vEdge device to OMP, select the **Advertise** tab and configure the following parameters:

*Table 16:*

| Parameter Name | Description |
|---|---|
| Advertise | Click **On** or **Off** to enable or disable the Cisco vEdge device advertising to OMP the routes that it learns locally:<br>• BGP—Click **On** to advertise BGP routes to OMP. By default, BGP routes are not advertised to OMP.<br>• Connected—Click **Off** to disable advertising connected routes to OMP. By default, connected routes are advertised to OMP.<br>• OSPF—Click **On** and click **On** again in the External field that appears to advertise external OSPF routes to OMP. OSPF inter-area and intra-area routes are always advertised to OMP. By default, external OSPF routes are not advertised to OMP.<br>• Static—Click **Off** to disable advertising static routes to OMP. By default static routes are advertised to OMP.<br>To configure per-VPN route advertisements to OMP, use the VPN feature template . |

Click **Save**.

# Configure OMP Using CLI

By default, OMP is enabled on all Cisco vEdge devices and vSmart controllers. OMP must be operational for the Cisco SD-WAN overlay network to function. If you disable it, you disable the overlay network.

OMP Support on

support the following:

- IPv4 and IPv6 protocols, which are both turned on by default for VPN 0
- OMP route advertisements to BGP, EIGRP, OSPF, connected routes, and static routes

### Configure OMP Graceful Restart

OMP graceful restart is enabled by default on vSmart controllers and Cisco SD-WAN devices. OMP graceful restart has a timer that tells the OMP peer how long to retain the cached advertised routes. When this timer expires, the cached routes are considered to be no longer valid, and the OMP peer flushes them from its route table.

The default timer is 43,200 seconds (12 hours), and the timer range is 1 through 604,800 seconds (7 days). To modify the default timer value:

```
Device(config-omp)# timers graceful-restart-timer seconds
```

To disable OMP graceful restart:

```
Device(config-omp)# no omp graceful-restart
```

The graceful restart timer is set up independently on each OMP peer; that is, it is set up separately on each Cisco vEdge Deviceand vSmart controller. To illustrate what this means, let's consider a vSmart controller that uses a graceful restart time of 300 seconds, or 5 minutes, and a Cisco vEdge Device that is configured with a timer of 600 seconds (10 minutes). Here, the vSmart controller retains the OMP routes learned from that device for 10 minutes—the graceful restart timer value that is configured on the device and that the device has sent to the vSmart controller during the setup of the OMP session. The Cisco vEdge Device retains the routes it learns from the vSmart controller for 5 minutes, which is the default graceful restart time value that is used on the vSmart controller and that the controller sent to the device, also during the setup of the OMP session.

While a vSmart controller is down and a Cisco vEdge Device is using cached OMP information, if you reboot the device, it loses its cached information and hence will not be able to forward data traffic until it is able to establish a control plane connection to the vSmart controller.

### Advertise Routes to OMP

By default, a Cisco vEdge Device advertises connected, static routes, and OSPF inter-area and intra-area routes to OMP, and hence to the vSmart controller responsible for the device's domain. The device does not advertise BGP or OSPF external routes to OMP.

To have the device advertise these routes to OMP, and hence to the vSmart controller responsible for the device's domain, use the advertise command:

Route advertisements in OMP are done either by applying the configuration at the global level or at the specific VPN level. To enable certain protocol route advertisements in all VPNs, you must add the configuration at the global level as shown in the example below.

```
Device# config
Device(config)# omp
Device(config-omp)# advertise bgp
Device(config-omp)# commit
```

To enable route advertisements for a certain protocol in only a few VPNs, you must remove any global-level configuration and add a per-VPN-level configuration as shown below:

```
Device# config
Device(config)# omp
Device(config-omp)# no advertise bgp
Device(config)# vpn 2
Device(config-vpn-2)# omp advertise bgp
Device(config-omp)# vpn 4
Device(config-vpn-4)# omp advertise bgp
Device(config-omp)# commit
```

To disable certain protocol route advertisements in all or a few VPNs, you should make sure that the configuration is present at neither the global level nor the VPN level.

For OSPF, the route type can be **external**.

The **bgp**, **connected**, **ospf**, and **static** options advertise all learned or configured routes of that type to OMP. To advertise a specific route instead of advertising all routes for a protocol, use the **network** option, specific the prefix of the route to advertise.

For individual VPNs, you can aggregate routes from the specified prefix before advertising them into OMP. By default, the aggregated prefixes and all individual prefixes are advertised. To advertise only the aggregated prefix, include the **aggregate-only** option.

Route advertisements that you set with the **omp advertise** command apply to all VPNs configured on the device. Route advertisements that you set with the **vpn omp advertise** command apply only to the specific VPN. If you configure route advertisements with both commands, they are both applied.

By default, when BGP advertises routes into OMP, BGP advertises each prefix's metric. BGP can also advertise the prefix's AS path:

```
Device(config)# vpn vpn-id router bgp
Device(config-bgp)# propagate-aspath
```

When you configure BGP to propagate AS path information, the device sends AS path information to devices that are behind the Cisco vEdge Devices (in the service-side network) that are running BGP, and it receives AS path information from these routers. If you are redistributing BGP routes into OMP, the AS path information is included in the advertised BGP routes. If you configure BGP AS path propagation on some but not all devices in the overlay network, the devices on which it is not configured receive the AS path information but they do not forward it to the BGP routers in their local service-side network. Propagating AS path information can help to avoid BGP routing loops.

In networks that have both overlay and underlay connectivity—for example, when devices are interconnected by both a Cisco SD-WAN overlay network and an MPLS underlay network—you can assign as AS number to OMP itself. For devices running BGP, this overlay AS number is included in the AS path of BGP route updates. To configure the overlay AS:

```
Device(config)# omp
Device(omp)# overlay-as as-number
```

You can specify the AS number in 2-byte ASDOT notation (1 through 65535) or in 4-byte ASDOT notation (1.0 through 65535.65535). As a best practice, it is recommended that the overlay AS number be a unique AS number within both the overlay and the underlay networks. That use, select an AS number that is not used elsewhere in the network.

If you configure the same overlay AS number on multiple devices in the overlay network, all these devices are considered to be part of the same AS, and as a result, they do not forward any routes that contain the overlay AS number. This mechanism is an additional technique for preventing BGP routing loops in the network.

### Configure the Number of Advertised Routes

A Cisco vEdge Device can have up to six WAN interfaces, and each WAN interface has a different TLOC. (A WAN interface is any interface in VPN 0 (or transport VRF) that is configured as a tunnel interface. Both physical and loopback interfaces can be configured to be tunnel interfaces.) The device advertises each route–TLOC tuple to the Cisco vSmart Controller.

The Cisco vSmart Controller redistributes the routes it learns from Cisco vEdge Devices, advertising each route–TLOC tuple. If, for example, a local site has two devices, a Cisco vSmart Controller could potentially learn eight route–TLOC tuples for the same route.

By default, Cisco vEdge Devices and Cisco vSmart Controllers advertises up to four equal-cost route–TLOC tuples for the same route. You can configure them to advertise from 1 to 16 route–TLOC tuples for the same route:

```
Device(config-omp)# send-path-limit 14
```

If the limit is lower than the number of route–TLOC tuples, the Cisco vEdge Device or Cisco vSmart Controller advertises the best routes.

### Configure the Number of Installed OMP Paths

Cisco vEdge Devices install OMP paths that they received from the Cisco vSmart Controller into their local route table. By default, a Cisco vEdge Devices installs a maximum of four unique OMP paths into its route table. You can modify this number:

```
vEdge(config-omp)# ecmp-limit 2
```

The maximum number of OMP paths installed can range from 1 through 16.

### Configure the OMP Hold Time

The OMP hold time determines how long to wait before closing the OMP connection to a peer. If the peer does not receive three consecutive keepalive messages within the hold time, the OMP connection to the peer is closed. The default OMP hold time is 60 seconds but it can be configured to up to 65,535 seconds. To modify the OMP hold time interval:

```
Device(config-omp)# timers holdtime 75
```

The hold time can be in the range 0 through 65535 seconds.

The keepalive timer is one-third the hold time and is not configurable.

If the local device and the peer have different hold time intervals, the higher value is used.

If you set the hold time to 0, the keepalive and hold timers on the local device and the peer are set to 0.

The hold time must be at least two times the hello tolerance interval set on the WAN tunnel interface in VPN 0. To configure the hello tolerance interface, use the hello-tolerance command.

### Configure the OMP Update Advertisement Interval

By default, OMP sends Update packets once per second. To modify this interval:

```
Device(config-omp)# timers advertisement-interval 5000
```

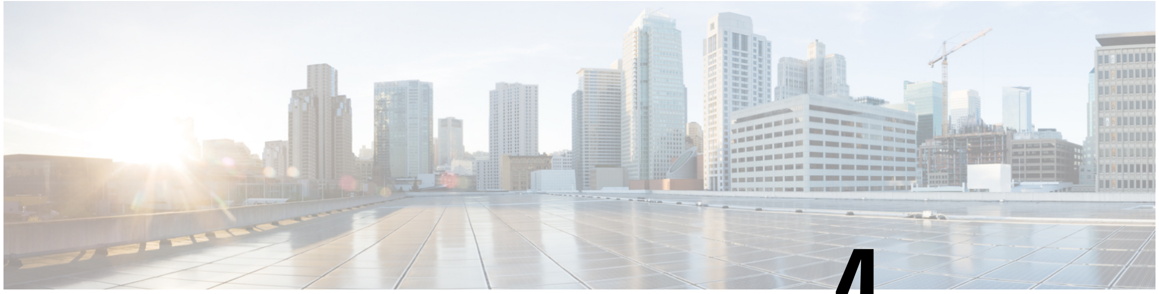
The interval can be in the range 0 through 65535 seconds.

### Configure the End-of-RIB Timer

After an OMP session goes down and then comes back up, an end-of-RIB (EOR) marker is sent after 300 seconds (5 minutes). After this maker is sent, any routes that were not refreshed after the OMP session came back up are considered to be stale and are deleted from the route table. To modify the EOR timer:

```
Device(config-omp)# timers eor-timer 300
```

The time can be in the range 1 through 3600 seconds (1 hour).



CHAPTER 4

# Multicast Overlay Routing

The Cisco SD-WAN multicast overlay implementation extends native multicast by creating a secure optimized multicast tree that runs on top of the overlay network.

The Cisco SD-WAN multicast overlay software uses Protocol Independent Multicast Sparse Mode (PIM-SM) for multicasting traffic on the overlay network. PIM-SM builds unidirectional shared trees rooted at a rendezvous point (RP), and each multicast group has one shared tree that is rooted at a single RP. Once a shared tree has been built such that a last-hop router learns the IP address for the multicast source, the router engages in a switchover from the shared tree to initiate the construction of a source (or shortest-path) tree. The source tree uses the lowest metric path between the source and last-hop router, which may be entirely, partially, or not at all congruent with the shared tree.

## Supported Protocols

Cisco SD-WAN overlay multicast network supports the Protocol Independent Multicast (PIM) and Internet Group Management Protocol (IGMP) and multicast template configurations on all the platforms.

### PIM

Viptela overlay multicast supports PIM version 2 (defined in RFC 4601 ), with some restrictions.

On the service side, the Viptela software supports native multicast. A vEdge router appears as a native PIM router and establishes PIM neighborship with other PIM routers at a local site. To properly extend multicast trees into the overlay network, a vEdge router may require other supporting routers in a local site. If a PIM-SM RP is required at a site, that function must be provided by a non-Viptela router, because the vEdge router currently has no native support for the rendervouz point functionality. Receivers residing downstream of a vEdge router can join multicast streams by exchanging IGMP membership reports directly with the device, and no other routers are required. This applies only to sites that have no requirement for supporting local sources or PIM SM rendezvouz points.

On the transport side, PIM-enabled vEdge routers originate multicast service routes (called multicast autodiscover routes),sending them via OMP to the vSmart controllers. The multicast autodiscover routes indicate whether the router has PIM enabled and whether it is a replicator. If the router is a replicator and the load threshold has been configured, this information is also included in the multicast autodiscover routes. Each PIM router also conveys information learned from the PIM join messages sent by local-site multicast-enabled routers, including multicast group state, source information, and RPs. These routes assist vEdge routers in performing optimized joins across the overlay when joining existing multicast sources.

vEdge routers support PIM source-specific mode (SSM), which allows a multicast source to be directly connected to the router.

### PIM Scalability Information

When configuring PIM, the following scalability limits apply:

- Any single vEdge router supports a maximum of 1024 multicast state entries. Note that a (*,G) and an (S,G) for the same group count as two entries.
- The 1024 multicast state entries are shared across all configured VPNs on a single vEdge router.
- Each state entry can contain a maximum of 64 service-side entries and a maximum of 256 transport-side entries in its outgoing interface list (OIL).

### Rendezvous Points

The root of a PIM multicast shared tree resides on a router configured to be a rendezvous point (RP). Each RP acts as the RP and the root of a shared tree (or trees) for specific multicast group ranges. In the Viptela overlay network, RPs are non-Viptela routers that reside in the local-site network. The RP function is typically assigned to one or two locations in the network; it is not required at every site. vEdge routers do not currently support the RP functionality, so non-Viptela routers must provide this function in the applicable sites.

The Viptela software supports the auto-RP protocol for distributing RP-to-group mapping information to local-site PIM routers. With this information, each PIM router has the ability to forward joins to the correct RP for the group that a downstream IGMP client is attempting to join. Auto-RP updates are propagated to downstream PIM routers if such routers are present in the local site.

### Replicators

For efficient use of WAN bandwidth, strategic vEdge routers can be deployed and configured as replicators throughout the overlay network. Replicators mitigate the requirement for an ingress router to replicate a multicast stream once for each receiver.

As discussed above, replicators advertise themselves, via OMP multicast-autodiscover routes, to the vSmart controllers in the overlay network. The controllers then forward the replicator location information to the PIM-enabled vEdge routers that are in the same VPN as the replicator.

A replicator vEdge router receives streams from multicast sources, replicates them, and forwards them to multicast receivers. The details of the replication process are discussed below, in the section Multicast Traffic Flow through the Overlay Network.

A replicator is typically vEdge router located at a colo site or another site with a higher-speed, or a high-speed, connection to the WAN transport network.

### Multicast Service Routes

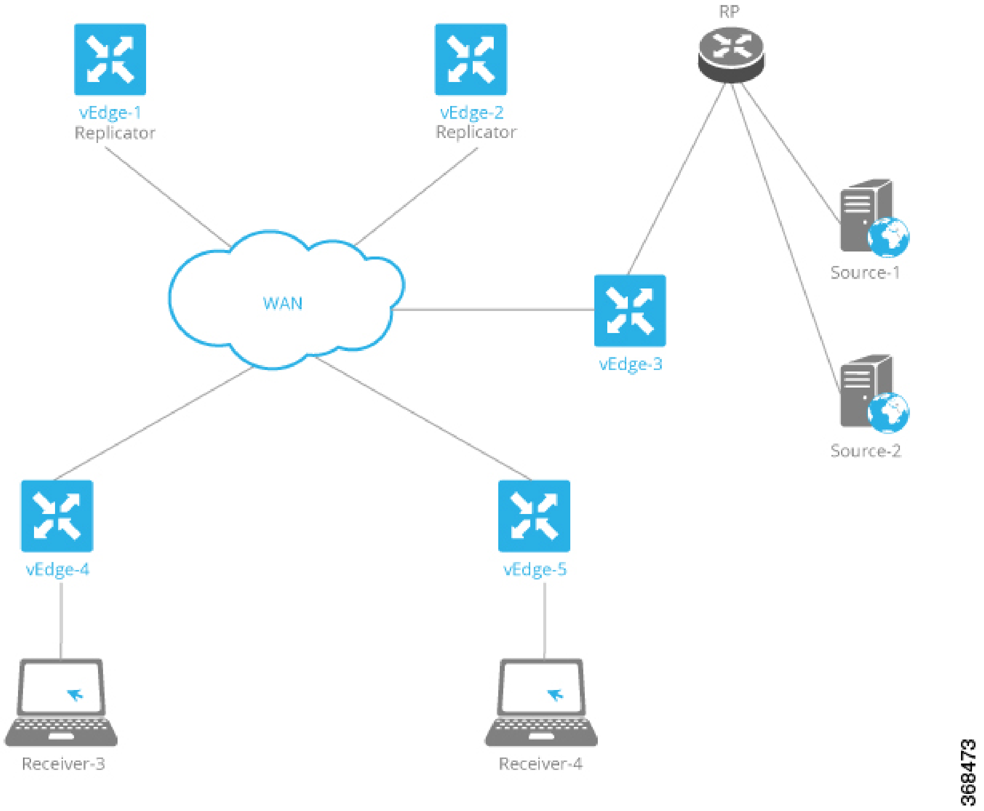
vEdge routers send multicast service routes to the vSmart controller via OMP. From these routes, the controller processes and forwards joins for requested multicast groups towards the source address as specified in the original PIM join message that helped originate the OMP multicast service route. The source address can be either the IP address of an RP if the originating router is attempting to join the shared tree or the IP address of the actual source of the multicast stream if the originating router is attempting to join the source tree.

## IGMP

Cisco SD-WAN overlay multicast routing supports the Internet Group Management Protocol (IGMP) version 2 (defined in RFC 2236 ). Cisco vEdge devices use IGMP to process receiver membership reports for the hosts in a particular VPN and to determine, for a given group, whether multicast traffic should be forwarded and state should be maintained. vEdge routers listen for both IGMPv1 and IGMPv2 group membership reports.

# Traffic Flow in Multicast Overlay Routing

Let's look at the high-level topology of the Cisco SD-WAN overlay network multicast solution to illustrate how traffic from multicast sources is delivered to multicast receivers. The topology contains five vEdge routers:



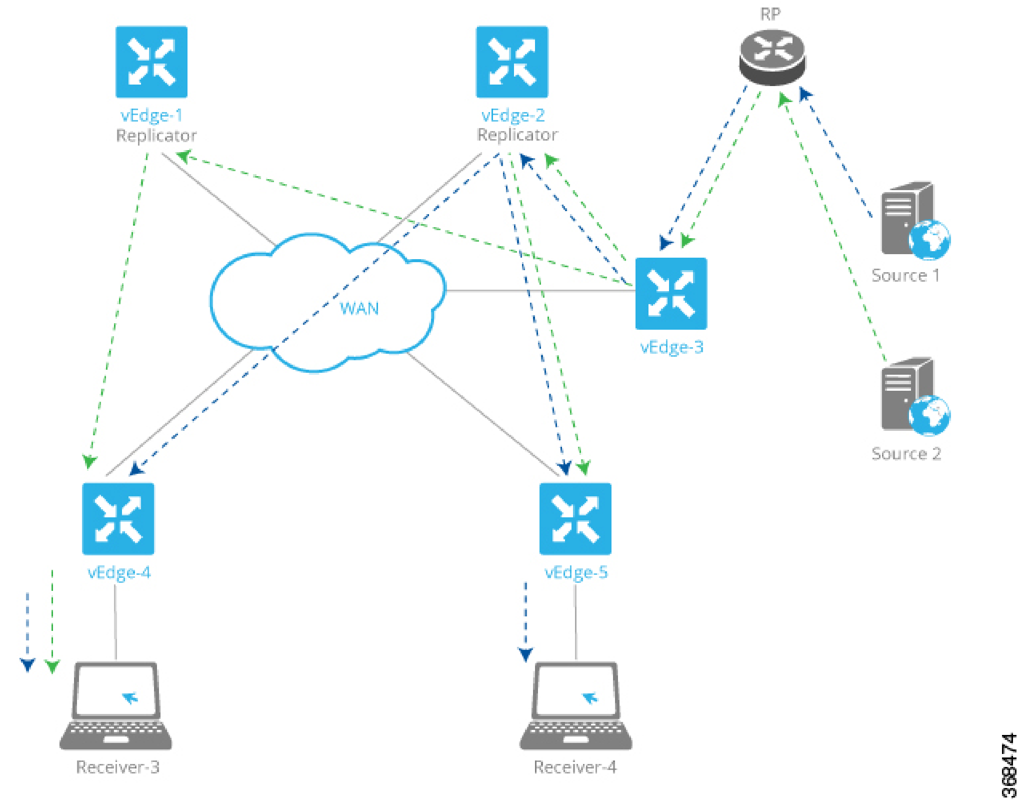
- vEdge router vEdge-3 is located at a site with two multicast sources, Source-1 and Source-2. This site also has a non-vEdge router that functions as a PIM-SM RP. Even though the vEdge-3 router is the ingress router for streams from these two multicast sources, it performs no packet replication. Instead, it forwards the multicast streams to replicators in the overlay network. The vEdge-3 router has learned the addresses of the replicators via OMP from a vSmart controller.

- vEdge routers vEdge-1 and vEdge-2 are two multicast replicators in the overlay network. Their job as replicators is to receive streams from multicast sources, replicate the streams, and then forward them to receivers. In this topology, the vEdge-3 router forwards the multicast streams from the two multicast sources in its local network to vEdge-1 or vEdge2, or both, and these routers then replicate and forward the streams to the receivers located in the local sites behind vEdge routers vEdge-4 and vEdge5. Which replicator receives a stream depends on the group address, the identity of the vEdge routers that joins that given group, and the current load of the replicator. The typical situation is that only a single replicator is replicating traffic for a given group, but this may vary depending on the physical scope of the given group.
- vEdge router vEdge4 is located at a site that has one multicast receiver, Receiver-3, which receives streams from Source-1 and Source-2.
- vEdge router vEdge5 is located at another site with one multicast receiver, Receiver-4. This receiver gets streams only from one source, Source-1.

Now, let’s examine how multicast traffic flows from the sources to the receivers.

The two multicast sources, Source-1 and Source-2, send their multicast streams (the blue stream from Source-1 and the green stream from Source-2) to the RP. Because the destination IP addresses for both streams are at remote sites, the RP forwards them to vEdge-3 for transmission onto the transport/WAN network. vEdge-3 has learned from the vSmart controller that the network has two replicators, vEdge-1 and vEdge-2, and so forwards the two multicast streams to them, without first replicating the streams.

The two replicators have learned from a vSmart controller the locations of multicast receivers for the two streams. The vEdge-1 replicator makes one copy of the green stream and forwards it to vEdge-4, which in turns forwards it to the Receiver-3. The vEdge-2 replicator makes one copy of the green stream, which it forwards to vEdge-5 (from which it goes on to Receiver-4), and it makes two copies of the blue stream, which it forwarsa to vEdge-4 and vEdge-5 (and which they then forward to the two receivers).



Now, let's look at the multicast configurations on the five vEdge routers:

- vEdge router vEdge-1 is a PIM replicator for a particular VPN. If we assume that no multicast sources, receivers, or RPs are located in its local network, the configuration of this router is simple: In the VPN, enable the replicator functionality, with the **router multicast-replicator local** command, and enable PIM, with the **router pim** command.

- vEdge router vEdge-2 also acts only as a replicator in the same VPN as vEdge-1, and you configure it with the same commands, **router multicast-replicator local** and **router pim**, when configuring the VPN. Each replicator can accept a maximum number of new PIM joins, and when this threshold value is reached, all new joins are sent to the second replicator. (If there is only one replicator, new joins exceeding the threshold are dropped.)

- vEdge router vEdge-4 runs PIM. You enable PIM explicitly on the service side within a VPN, specifying the service-side interface that connects to the multicast domain in the local network. So within the VPN, you include the **router pim interface** command. You can also enable auto-RP with the **router pim auto-rp** command. On the transport side, no explicit configuration is required. The vEdge router automatically directs multicast traffic—both OMP control plane messages and multicast streams—to VPN 0, which is the WAN transport VPN.

- vEdge router vEdge-5 is also configured to run PIM in the same way as vEdge-4: You configure the service-side interface name and RP information.

PIM must be enabled in the same VPN on all five of these vEdge routers so that the multicast streams can be transmitted and received.

# Configure Multicast Overlay Routing

For any vEdge routers to be able to participate in the multicast overlay network, you configure PIM on those routers. You can optionally configure IGMP to allow individual hosts on the service side to join multicast groups within a particular VPN.

### Limitations of Multicast Configuration

You cannot configure the following for multicast overlay routing:

- Data policy
- Access lists
- Mirroring

# Configure PIM Using vManage Templates

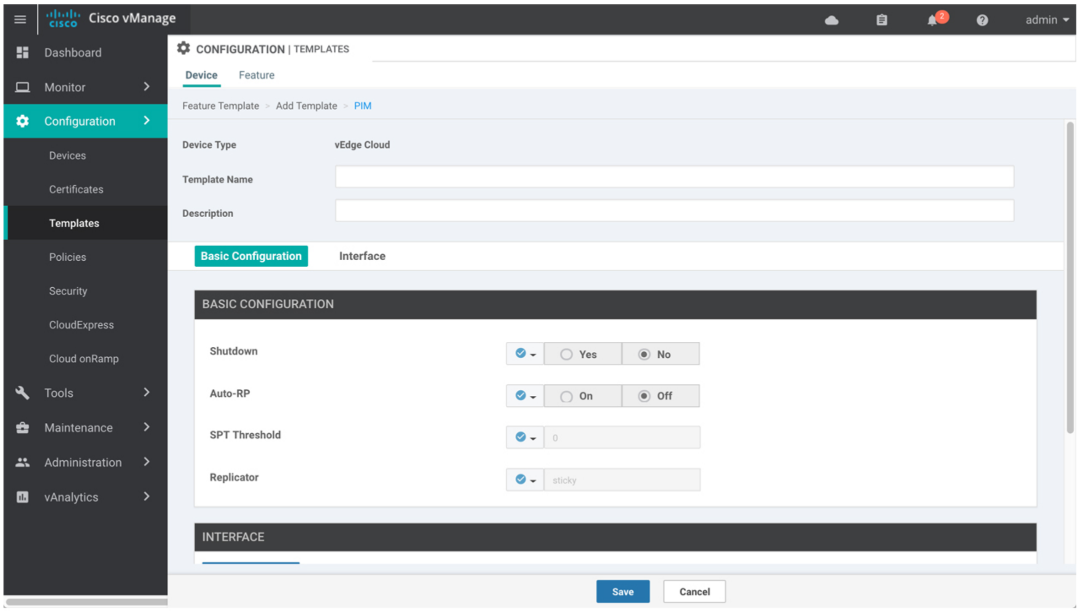
Use the PIM template for all vEdge Cloud and vEdge router devices.

Configure the PIM Sparse Mode (PIM-SM) protocol using vManage templates so that a router can participate in the Viptela multicast overlay network:

1. Create a PIM feature template to configure PIM parameters, as described in this topic.
2. Optionally, create an IGMP feature template to allow individual hosts on the service side to join multicast groups within a particular VPN. See Configure IGMP Using vManage Templates
3. Optionally, create a Multicast feature template to configure a vEdge router to be a multicast replicator. See the Multicast help topic.
4. Create a VPN feature template to configure parameters for the VPN that is running PIM. See the VPN help topic.

### Create a PIM Feature Template

1. In vManage NMS, select **Configuration** > **Templates**.
2. In the Device tab, click **Create Template**.
3. From the Create Template drop-down, select **From Feature Template**.
4. From the Device Model drop-down, select the type of device for which you are creating the template.
5. Click the **Service VPN** tab located directly beneath the Description field, or scroll to the Service VPN section.
6. Click the **Service VPN** drop-down.



Device Type vEdge Cloud

Template Name

Description

Shutdown ○ Yes ● No

Auto-RP ○ On ● Off

SPT Threshold 0

Replicator sticky

7. Under Additional VPN Templates, located to the right of the screen, click **PIM**.

8. From the PIM drop-down, click **Create Template**. The PIM template form is displayed. The top of the form contains fields for naming the template, and the bottom contains fields for defining PIM parameters.

9. In the Template Name field, enter a name for the template. The name can be up to 128 characters and can contain only alphanumeric characters.

10. In the Template Description field, enter a description of the template. The description can be up to 2048 characters and can contain only alphanumeric characters.

When you first open a feature template, for each parameter that has a default value, the scope is set to Default (indicated by a check mark), and the default setting or value is shown. To change the default or to enter a value, click the scope drop-down to the left of the parameter field and select one of the following:

*Table 17:*

| Parameter Scope | Scope Description |
|---|---|
| Device Specific (indicated by a host icon) | Use a device-specific value for the parameter. For device-specific parameters, you cannot enter a value in the feature template. You enter the value when you attach a Viptela device to a device template .<br>When you click Device Specific, the Enter Key box opens. This box displays a key, which is a unique string that identifies the parameter in a CSV file that you create. This file is an Excel spreadsheet that contains one column for each key. The header row contains the key names (one key per column), and each row after that corresponds to a device and defines the values of the keys for that device. You upload the CSV file when you attach a Viptela device to a device template. For more information, see Create a Template Variables Spreadsheet .<br>To change the default key, type a new string and move the cursor out of the Enter Key box.<br>Examples of device-specific parameters are system IP address, hostname, GPS location, and site ID. |
| Global (indicated by a globe icon) | Enter a value for the parameter, and apply that value to all devices.<br>Examples of parameters that you might apply globally to a group of devices are DNS server, syslog server, and interface MTUs. |

### Configure Basic PIM

To configure PIM, select the Basic Configuration tab and configure the following parameters. Parameters marked with an asterisk are required to configure PIM.

*Table 18:*

| Parameter Name | Description |
|---|---|
| Shutdown* | Ensure that No is selected, to enable PIM. |
| Auto-RP | Click On to enable auto-RP to enable automatic discovery of rendezvous points (RPs) in the PIM network so that the router receivea group-to-RP mapping updates. By default, auto-RP is disabled. |
| SPT Threshold | Specify the traffic rate, in kbps, at which to switch from the shared tree to the shortest-path tree (SPT). Configuring this value forces traffic to remain on the shared tree and travel via the RP instead of via the SPT. |
| Replicator | For a topology that includes multicast replicators, determine how the replicator for a multicast group is chosen:<br>• Random—Choose the replicator at random.<br>• Sticky—Always use the same replicator. This is the default. |

To save the feature template, click Save.

### Configure PIM Interfaces

If the router is just a multicast replicator and is not part of a local network that contains either multicast sources or receivers, you do not need to configure any PIM interfaces. The replicator learns the locations of multicast sources and receivers from the OMP messages it exchanges with the vSmart controller. These control plane messages are exchanged in the transport VPN (VPN 0). Similarly, other vEdge routers discover replicators dynamically, through OMP messages from the vSmart controller.

To configure PIM interfaces, select the Interface tab. Then click **Add New Interface** and configure the following parameters:

*Table 19:*

| Parameter Name | Description |
|---|---|
| Name | Enter the name of an interface that participates in the PIM domain, in the format **ge** *slot /port*. |
| Hello Interval | Specify how often the interface sends PIM hello messages. Hello messages advertise that PIM is enabled on the router.*Range:* 1 through 3600 seconds*Default:* 30 seconds |
| Join/Prune Interval | Specify how often PIM multicast traffic can join or be removed from a rendezvous point tree (RPT) or shortest-path tree (SPT). vEdge routers send join and prune messages to their upstream RPF neighbor.*Range:* 10 through 600 seconds*Default:* 60 seconds |

To edit an interface, click the pencil icon to the right of the entry.

To delete an interface, click the trash icon to the right of the entry.

To save the feature template, click Save.

### Release Information

# Configure PIM Using CLI

### Enable PIM at a Site with Multicast Sources

For a vEdge router located at a site that contains one or more multicast sources, you enable PIM on the service-side interface or interfaces. These are the interfaces that face the local-site network. You enable PIM per VPN, so you must configure PIM and PIM interfaces for all VPNs support multicast services. You cannot configure PIM in VPN 0 (the transport VPN facing the overlay network) or in VPN 512 (the management VPN).

For each VPN, you must configure the name of the service-side interface. You can optionally configure auto-RP to receive group-to-RP mapping updates.

To configure PIM at a site with multicast sources:

1. Configure a VPN for the PIM network:

   ```
   vEdge(config)# vpn vpn-id
   ```

   *vpn-id* can be any VPN number except VPN 0 (the transport VPN facing the overlay network) or VPN 512 (the management VPN).

2. Configure the interfaces in the VPN:

```
vEdge(config-vpn)# interface interface-name
vEdge(config-interface)# ip address prefix/length
vEdge(config-interface)# no shutdown
```

The interface names in the two **interface** names must be the same.

3. Configure PIM and the interfaces that participate in the PIM network:

   ```
   vEdge(config-vpn)# router pim
   vEdge(config-pim)# interface interface-name
   vEdge(config-interface)# no shutdown
   ```

   The interface name in the two **interface** commands must be the same.

4. Optionally, modify PIM timers on the interface. The default PIM hello interval is 30 seconds, and the default join/prune interval is 60 seconds.

   ```
   vEdge(config-interface)#  hello-interval seconds
   vEdge(config-interface)#  join-prune-interval seconds
   ```

   The hello interval can be in the range of 1 through 3600 seconds. The join/prune interval can be in the range of 10 through 600 seconds.

5. Optionally, enable automatic discover of rendezvous points (RPs) in the PIM network:

   ```
   vEdge(config-pim)#  auto-rp
   ```

Here is an example of a PIM configuration on a vEdge router:

```
vpn 10
 router
  pim
    interface ge1/1
      no shutdown
    auto-rp
```

### Enable PIM at a Site with Multicast Receivers

For a vEdge router located at a site that contains one or more multicast receivers, you enable PIM on the service-side interface or interfaces (the interfaces facing the local-site network). You enable PIM per VPN, so you must configure PIM and PIM interfaces for all VPNs support multicast services.

For each VPN, you must configure the name of the service-side interface.

To configure PIM at a site with multicast receivers:

1. Configure a VPN for the PIM network:

   ```
   vEdge(config)# vpn vpn-id
   ```

   *vpn-id* can be any VPN number except VPN 0 (reserved for control plane traffic) or VPN 512 (the management VPN).

2. Configure PIM and the interfaces that participate in the PIM network:

   ```
   vEdge(config-vpn)# router pim
   vEdge(config-pim)# interface interface-name
   ```

3. Configure the interface used by PIM in the PIM VPN:

   ```
   vEdge(config-vpn)# interface interface-name
   vEdge(config-interface)# ip address prefix/length
   vEdge(config-interface)# no shutdown
   ```

   The interface names in the two **interface** names must be the same.

4. By default, a vEdge router joins the shortest-path tree (SPT) immediately after the first packet arrives from a new source. To force traffic to remain on the shared tree and travel via the RP instead of via the SPT, configure the traffic rate at which to switch from the shared tree to the SPT:

   ```
   vEdge(config-vpn)# router pim spt-threshold kbps
   ```

   The rate can be from 0 through 100 kbps.

5. In a topology that includes multicast replicators, the Cisco SD-WAN software, by default, uses the same replicator for a multicast group. You can have the software choose the replicator randomly:

   ```
   vEdge(config-vpn)# router pim replicator-selection random
   ```

Here is an example of a PIM configuration on a vEdge router:

```
vEdge(config-vpn-2)# show full-configuration
vpn 2
 router
  pim
   interface ge0/7
   exit
  exit
 !
 interface ge0/7
  ip address 10.0.100.15/24
  no shutdown
 !
!
```

### Configure a Multicast Replicator

For a vEdge router that is a replicator, the configuration has two parts: you configure the router to be a replicator, and you enable PIM on each VPN that participates in a multicast domain.

To configure a replicator:

1. Configure a VPN for the PIM network:

   ```
   vEdge(config)# vpn vpn-id
   ```

   *vpn-id* can be any VPN number except VPN 0 (the transport VPN facing the overlay network) or VPN 512 (the management VPN).

2. Configure the replicator functionality on the local vEdge router:

   ```
   vEdge(config-vpn)# router multicast-replicator local
   ```

3. On the transport side, a single vEdge router acting as a replicator can accept a maximum of 1024 (*,G) and (S,G) joins. For each join, the router can accept 256 tunnel outgoing interfaces (OILs). To modify the number of joins the replicator can accept, change the value of the join threshold:

   ```
   vEdge(config-router)# multicast-replicator threshold number
   ```

4. Enable PIM on each VPN that participates in a multicast domain:

   ```
   vEdge(config)# vpn vpn-id
   vEdge(config-vpn)# router pim
   ```

   If the router is just a replicator and is not part of a local network that contains either multicast sources or receivers, you do not need to configure any interfaces in the PIM portion of the configuration. The replicator learns the locations of multicast sources and receivers from the OMP messages it exchanges with the vSmart controller. These control plane messages are exchanged in the transport VPN (VPN 0). Similarly, the other vEdge routers discover replicators dynamically, through OMP messages from the vSmart controller.

# Configure IGMP Using vManage Templates

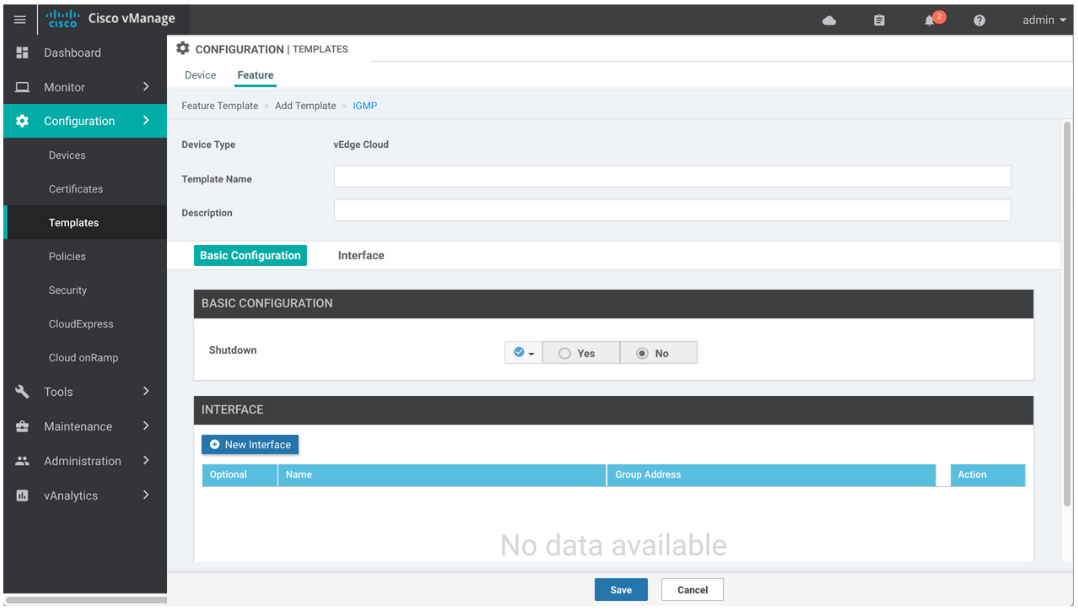
Use the IGMP template for all vEdge Cloud and vEdge router devices. Internet Group Management Protocol (IGMP) allows vEdge routers to join multicast groups within a particular VPN.

To configure IGMP using vManage templates:

1. Create an IGMP feature template to configure IGMP parameters.
2. Create the interface in the VPN to use for IGMP. See the VPN-Interface-Ethernet help topic.
3. Create a VPN feature template to configure VPN parameters. See the VPN help topic.

### Navigate to the Template Screen and Name the Template

1. In vManage NMS, select **Configuration** > **Templates**.
2. In the Device tab, click **Create Template**.
3. From the Create Template drop-down, select **From Feature Template**.
4. From the Device Model drop-down, select the type of device for which you are creating the template.
5. Click the **Service VPN** tab located directly beneath the Description field, or scroll to the Service VPN section.
6. Click the **Service VPN** drop-down.



Device Type vEdge Cloud

Template Name

Description

Shutdown ○ Yes ● No

| Optional | Name | Group Address | Action |
|---|---|---|---|

No data available

7. Under Additional VPN Templates, located to the right of the screen, click **IGMP**.

8. From the IGMP drop-down, click **Create Template**. The IGMP template form is displayed. The top of the form contains fields for naming the template, and the bottom contains fields for defining IGMP parameters.

9. In the Template Name field, enter a name for the template. The name can be up to 128 characters and can contain only alphanumeric characters.

10. In the Template Description field, enter a description of the template. The description can be up to 2048 characters and can contain only alphanumeric characters.

When you first open a feature template, for each parameter that has a default value, the scope is set to Default (indicated by a check mark), and the default setting or value is shown. To change the default or to enter a value, click the scope drop-down to the left of the parameter field and select one of the following:

*Table 20:*

| Parameter Scope | Scope Description |
|---|---|
| Device Specific (indicated by a host icon) | Use a device-specific value for the parameter. For device-specific parameters, you cannot enter a value in the feature template. You enter the value when you attach a Viptela device to a device template .<br>When you click Device Specific, the Enter Key box opens. This box displays a key, which is a unique string that identifies the parameter in a CSV file that you create. This file is an Excel spreadsheet that contains one column for each key. The header row contains the key names (one key per column), and each row after that corresponds to a device and defines the values of the keys for that device. You upload the CSV file when you attach a Viptela device to a device template. For more information, see Create a Template Variables Spreadsheet .<br>To change the default key, type a new string and move the cursor out of the Enter Key box.<br>Examples of device-specific parameters are system IP address, hostname, GPS location, and site ID. |
| Global (indicated by a globe icon) | Enter a value for the parameter, and apply that value to all devices.<br>Examples of parameters that you might apply globally to a group of devices are DNS server, syslog server, and interface MTUs. |

### Configure Basic IGMP Parameters

To configure IGMP, select the Basic Configuration tab to enable IGMP. Then, select the Interface tab and click Add New Interface to configure IGMP interfaces. All parameters listed below are required to configure IGMP.

*Table 21:*

| Parameter Name | Description |
|---|---|
| Shutdown | nsure that No is selected to enable IGMP. |

| Parameter Name | Description |
|---|---|
| Interface Name | Enter the name of the interface to use for IGMP.<br>To add another interface, click the plus sign (+). To delete an interface, click the trash icon to the right of the entry. |
| Join Group Address | Click Add Join Group Address, and enter the address of a multicast group for the interface to join.<br>Click Add to add the new interface |

To save the feature template, click **Save**.

## Configure IGMP Using CLI

Configure IGMP to allow individual hosts on the service side to join multicast groups within a particular VPN.

### Enable IGMP at a Site with Multicast Hosts

For VPNs in which you want to individual hosts to join multicast groups, you can enable IGMP on vEdge routers:

```
vEdge(config)#vpn vpn-id router igmp
vEdge(config-igmp)# interface interface-name
```

Ensure that the interface being used for IGMP is configured in the VPN:

```
vEdge(config)# vpn vpn-id
vEdge(config-vpn)# interface interface-name
vEdge(config-interface)# ip address prefix/length
vEdge(config-interface)# no shutdown
```

If desired, specify the multicast groups to initiate join requests with:

```
vEdge(config-igmp)# interface interface-name join-group group-ip-address
```

### Configure the Interface Bandwidth Allowed for Multicast Traffic

By default, multicast traffic can use up to 20 percent of the interface bandwidth. You can change this allocation to a value from 5 to 100 percent:

```
vEdge(config)# system multicast-buffer-percent percentage
```

This systemwide configuration applies to all multicast-enabled interfaces on the vEdge router.

# Multicast Routing CLI Reference

CLI commands for configuring and monitoring the IGMP, PIM, and Replicator routing protocols on vEdge routers.

### IGMP Configuration and Monitoring Commands

Use the following commands to configure IGMP within a VPN on a vEdge router:

```
vpn vpn-id
  router
    igmp
      interface interface-name
        join-group group-address
      [no] shutdown
```

Use the following commands to monitor IGMP:

- **clear igmp interface** —Clear the interfaces on which IGMP is enabled.
- **clear igmp protocol** —Flush all IGMP groups and relearn them.
- **clear igmp statistics** —Zero IGMP statistics.
- **show igmp groups** —Display information about multicast groups.
- **show igmp interface** —Display information about the interfaces on which IGMP is enabled.
- **show igmp statistics** —Display IGMP statistics.

### PIM and Multicast Replicator Configuration and Monitoring Commands

Use the following commands to configure PIM and multicast replicators within a VPN on a vEdge router:

```
vpn vpn-id
  router
    multicast-replicator local [threshold number]
```

```
vpn vpn-id
  router
    pim
      auto-rp
      interface interface-name
        hello-interval seconds
        join-prune-interval seconds
      replicator-selection
      [no] shutdown
      spt-threshold kbps
```

Use the following commands to monitor PIM and multicastreplicators:

- **clear ip mfib record** —Clear the statistics for a particular group, source, or VPN from the Multicast Forwarding Information Base (MFIB).
- **clear ip mfib stats** —Clear all statistics from the MFIB.
- **clear pim interface** —Relearn all PIM neighbors and joins.
- **clear pim neighbor** —Clear the statistics for a PIM neighbor.
- **clear pim protocol** —Clear all PIM protocol state.
- **clear pim statistics** —Clear all PIM-related statistcs and relearn all PIM neighbors and joins.
- **show ip mfib oil** —Display the list of outgoing interfaces from the MFIB.
- **show ip mfib stats** —Display packet transmission and receipt statistics for active entries in the MFIB.
- **show ip mfib summary** —Display a summary of all active entries in the MFIB.
- **show multicast replicator**— List information about multicast replicators.

- **show multicast rpf**—List multicast reverse-path forwarding information.
- **show multicast topology** —List information related to the multicast domain topology.
- **show multicast tunnel** —List information about the IPsec tunnels between multicast peers.
- **show omp multicast-auto-discover** —List the peers that support multicast.
- **show omp multicast-routes** —List the multicast routes that OMP has learned from PIM join messages.
- **show pim interface** —List the interfaces that are running PIM.
- **show pim neighbor** —List PIM neighbors.
- **show pim statistics** —Display all PIM-related statistics.

CHAPTER 5

# Segmentation

Network segmentation has existed for over a decade and has been implemented in multiple forms and shapes. At its most rudimentary level, segmentation provides traffic isolation. The most common forms of network segmentation are virtual LANs, or VLANs, for Layer 2 solutions, and virtual routing and forwarding, or VRF, for Layer 3 solutions.

There are many use cases for segmentation:

### Use Cases for Segmentation

- An enterprise wants to keep different lines of business separate (for example, for security or audit reasons).
- The IT department wants to keep authenticated users separate from guest users.
- A retail store wants to separate video surveillance traffic from transactional traffic.
- An enterprise wants to give business partners selective access only to some portions of the network.
- A service or business needs to enforce regulatory compliance, such as compliance with HIPAA, the U.S. Health Insurance Portability and Accountability Act, or with the Payment Card Industry (PCI) security standards.
- A service provider wants to provide VPN services to its medium-sized enterprises.
- An enterprise wants to set up a trial run of new service and wants to use a cloud service for development and system test.

### Limitations of Segmentation

One inherent limitation of segmentation is its scope. Segmentation solutions either are complex or are limited to a single device or pair of devices connected via an interface. As an example, Layer 3 segmentation provides the following:

1. Ability to group prefixes into a unique route table (RIB or FIB).
2. Ability to associate an interface with a route table so that traffic traversing the interface is routed based on prefixes in that route table.

This is a useful functionality, but its scope is limited to a single device. To extend the functionality throughout the network, the segmentation information needs to be carried to the relevant points in the network.

### How to Enable Network-Wide Segmentation

There are two approaches to providing this network-wide segmentation:

- Define the grouping policy at every device and on every link in the network (basically, you perform Steps 1 and 2 above on every device).
- Define the grouping policy at the edges of the segment, and then carry the segmentation information in the packets for intermediate nodes to handle.

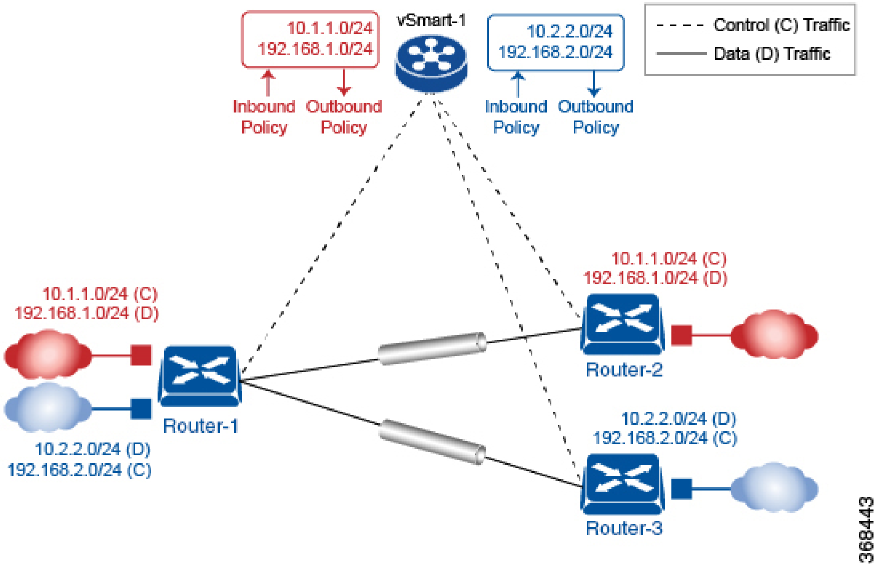
The first approach is useful if every device is an entry or exit point for the segment, which is generally not the case in medium and large networks. The second approach is much more scalable and keeps the transport network free of segments and complexity. MPLS-based Layer 3 VPNs are a popular example of segmentation at the edge.

# Segmentation in Cisco SD-WAN

In the Cisco SD-WAN overlay network, VPNs divide the network into different segments.

Cisco SD-WAN employs the more prevalent and scalable model of creating segments. Essentially, segmentation is done at the edges of a router, and the segmentation information is carried in the packets in the form of an identifier.

The figure below shows the propagation of routing information inside a VPN .



In this figure:

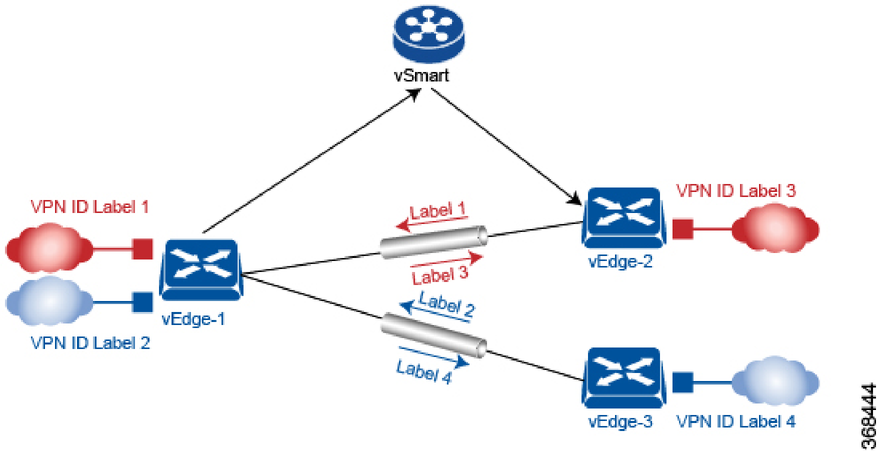
- Router-1 subscribes to two VPNs , red and blue.
    - The red VPN caters to the prefix 10.1.1.0/24 (either directly through a connected interface or learned via the IGP or BGP).
    - The blue VPN caters to the prefix 10.2.2.0/24 (either directly through a connected interface or learned via the IGP or BGP).
- Router-2 subscribes to the red VPN .
    - This VPN caters to the prefix 192.168.1.0/24 (either directly through a connected interface or learned via the IGP or BGP).
- Router-3 subscribes to the blue VPN .
    - This VPN caters to the prefix 192.168.2.0/24 (either directly through a connected interface or learned via the IGP or BGP).

Because each router has an OMP connection over a TLS tunnel to a vSmart controller, it propagates its routing information to the vSmart controller. On the vSmart controller, the network administrator can enforce policies to drop routes, to change TLOCs (which are overlay next hops) for traffic engineering or service chaining, or to change the VPN ID (see Policy Overview for more details). The network administrator can apply these policies as inbound and outbound policies on the vSmart controller.

All prefixes belonging to a single VPN are kept in a separate route table. This provides the Layer 3 isolation required for the various segments in the network. So, Router-1 has two VPN route tables, and Router-2 and Router-3 each have one route table. In addition, the vSmart controller maintains the VPN context of each prefix.

Separate route tables provide isolation on a single node. So now the question is how to propagate the routing information across the network.

In the Cisco SD-WAN solution, this is done using VPN identifiers, as shown in the figure below. AVPN ID carried in the packet identifies each VPN on a link. When you configure a VPN on a Router, the VPN has a label associated with it. The Router sends the label, along with the VPN ID, to the vSmart controller. The vSmart controller propagates this Router-to-VPN -ID mapping information to the other Routers in the domain. The remote Routers then use this label to send traffic to the appropriate VPN . The local Routers, on receiving the data with the VPN ID label, use the label to demultiplex the data traffic. This is similar to how MPLS labels are used. This design is based on standard RFCs and is compliant with regulatory procedures (such as PCI and HIPAA).



It is important to point out that the transport network that connects the routers is completely unaware of the VPNs . Only the routers know about VPNs ; the rest of the network follows standard IP routing.

# VPNs Used in Cisco SD-WAN Segmentation

The Cisco SD-WAN solution provides two default VPNs to separate traffic: Transport VPN and Management VPN.

### Transport VPNs

VPN 0 is the transport VPN. To enforce the inherent separation between services (such as prefixes that belong to the enterprise) and transport (the network that connects the vEdge routers), all the transport interfaces (that is, all the TLOCs) are kept in the transport VPN. This ensures that the transport network cannot reach the service network by default. Multiple transport interfaces can belong to the same transport VPN, and packets can be forwarded to and from transport interfaces. VPN 0 or transport VPN carries control traffic over secure DTLS or TLS connections between vSmart controllers and vEdge routers, and between vSmart controllers and vBond orchestrators

VPN 0 contains all interfaces for a device except for the management interface, and all the interfaces are disabled. For the control plane to establish itself so that the overlay network can function, you must configure WAN transport interfaces in VPN 0. On vEdge routers, the interfaces in VPN 0 connect to some type of transport network or cloud, such as the Internet, MPLS, or Metro Ethernet. For each interface in VPN 0, you must set an IP address, and you create a tunnel connection that sets the color and encapsulation for the WAN transport connection. (The encapsulation is used for the transmission of data traffic.) These three parameters—IP address, color, and encapsulation—define a TLOC (transport location) on the vEdge router. The OMP session running on each tunnel sends the TLOC to the vSmart controllers so that they can learn the overlay network topology. For VPN 0, you can also set other interface-specific and VPN-specific properties in VPN 0.

### Dual Stack Support on Transport VPNs

In the transport VPN (VPN 0), vEdge routers and vSmart controllers support dual stack. To enable dual stack, configure an IPv4 address and an IPv6 address on the tunnel interface. The vEdge router learns from the vSmart controller whether a destination supports IPv4 or IPv6 addresses. When forwarding traffic, the router chooses either the IPv4 or the IPv6 TLOC based on the destination address.

### Management VPNs

VPN 512 is the management VPN. It carries out-of-band network management traffic among the Cisco SD-WAN devices in the overlay network. By default, VPN 512 is configured and enabled. You can modify this configuration if desired.

### Service VPNs

To segment user networks and user data traffic locally at each site and to interconnect user sites across the overlay network, you create additional VPNs on Cisco vEdge devices. These VPNs are identified by a number that is not 0 or 512. To enable the flow of data traffic, you associate interfaces with each VPN, assigning an IP address to each interface. These interfaces connect to local-site networks, not to WAN transport clouds. For each of these VPNs, you can set other interface-specific properties, and you can configure features specific for the user segment, such as BGP and OSPF routing, VRRP, QoS, traffic shaping, and policing.

# Configure VPNs Using vManage Templates

## Create a VPN Template

**Step 1** In Cisco vManage NMS, choose **Configuration** > **Templates**.

**Step 2** In the Device tab, click **Create Template**.

**Step 3** From the Create Template drop-down, select **From Feature Template**.

**Step 4** From the **Device Model** drop-down, select the type of device for which you are creating the template.

**Step 5** To create a template for VPN 0 or VPN 512:

a. Click the **Transport & Management** VPN tab located directly beneath the Description field, or scroll to the Transport & Management VPN section.

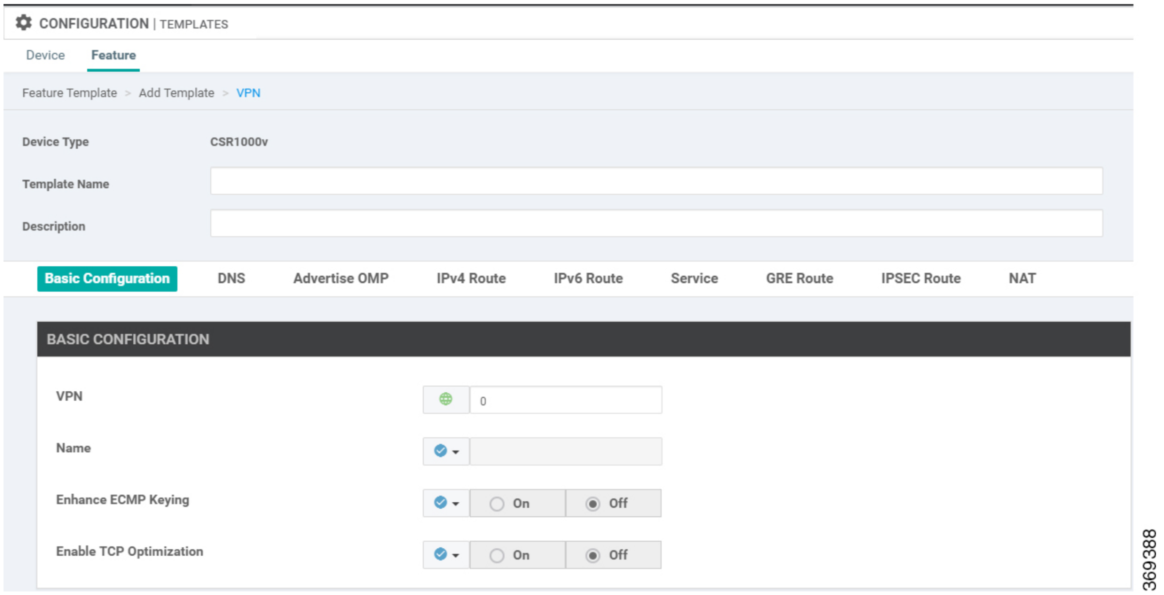
b. From the VPN 0 or VPN 512 drop-down, click **Create Template**. The VPN template form displays. The top of the form contains fields for naming the template, and the bottom contains fields for defining VPN parameters.

**Step 6** To create a template for VPNs 1 through 511, and 513 through 65530:

a. Click the **Service VPN** tab located directly beneath the Description field, or scroll to the Service VPN section.

b. Click the **Service VPN** drop-down.

c. From the VPN drop-down, click **Create Template**. The VPN template form displays. The top of the form contains fields for naming the template, and the bottom contains fields for defining VPN parameters.



Device Type: CSR1000v

Template Name:

Description:

VPN: 0

Name:

Enhance ECMP Keying: ○ On ● Off

Enable TCP Optimization: ○ On ● Off

**Step 7** In the Template Name field, enter a name for the template. The name can be up to 128 characters and can contain only alphanumeric characters.

**Step 8** In the Template Description field, enter a description of the template. The description can be up to 2048 characters and can contain only alphanumeric characters.

# Configure Basic VPN Parameters

To configure basic VPN parameters, choose the Basic Configuration tab and then configure the following parameters. Parameters marked with an asterisk are required to configure a VPN.

| Parameter Name | Description |
|---|---|
| VPN* | Enter the numeric identifier of the VPN.<br>Range for Cisco vEdge devices: 0 through 65530<br>Values for Cisco vSmart Controller and Cisco vManage devices: 0, 512 |
| Name | Enter a name for the VPN. |
| Enhance ECMP keying<br>(Cisco vEdge devices only) | Click **On** to enable the use in the ECMP hash key of Layer 4 source and destination ports, in addition to the combination of the source IP address, destination IP address, protocol, and DSCP field, as the ECMP hash key. ECMP keying is **Off** by default. |
| Enable TCP Optimization<br>Cisco vEdge devices only | Click **On** to enable TCP optimization for a service-side VPN (a VPN other than VPN 0 and VPN 512). TCP optimization fine-tunes TCP to decrease round-trip latency and improve throughput for TCP traffic. |

**Note** To complete the configuration of the transport VPN on a router, you must configure at least one interface in VPN 0.

To save the feature template, click **Save**.

# Configure Basic Interface Functionality

To configure basic interface functionality in a VPN, choose the **Basic Configuration** tab and configure the following parameters:

**Note** Parameters marked with an asterisk are required to configure an interface.

| Parameter Name | IPv4 or IPv6 | Options | Description |
|---|---|---|---|
| **Shutdown*** | Click **No** to enable the interface. | | |
| **Interface name*** | Enter a name for the interface. | | |
| **Description** | Enter a description for the interface. | | |
| **IPv4 / IPv6** | Click **IPv4** to configure an IPv4 VPN interface. Click **IPv6** to configure an IPv6 interface. | | |
| **Dynamic** | Click **Dynamic** to set the interface as a Dynamic Host Configuration Protocol (DHCP) client, so that the interface receives its IP address from a DHCP server. | | |
| | **Both** | **DHCP Distance** | Optionally, enter an administrative distance value for routes learned from a DHCP server. Default is 1. |
| | **IPv6** | **DHCP Rapid Commit** | Optionally, configure the DHCP IPv6 local server to support DHCP Rapid Commit, to enable faster client configuration and confirmation in busy environments.<br>Click **On** to enable DHCP rapid commit<br>Click **Off** to continue using the regular commit process. |
| **Static** | Click **Static** to enter an IP address that doesn't change. | | |
| | **IPv4** | **IPv4 Address** | Enter a static IPv4 address. |
| | **IPv6** | **IPv6 Address** | Enter a static IPv6 address. |
| **Secondary IP Address** | **IPv4** | Click **Add** to enter up to four secondary IPv4 addresses for a service-side interface. | |
| **IPv6 Address** | **IPv6** | Click **Add** to enter up to two secondary IPv6 addresses for a service-side interface. | |

| Parameter Name | IPv4 or IPv6 | Options | Description |
|---|---|---|---|
| **DHCP Helper** | **Both** | To designate the interface as a DHCP helper on a vEdge router, enter up to eight IP addresses, separated by commas, for DHCP servers in the network. A DHCP helper interface forwards BootP (broadcast) DHCP requests that it receives from the specified DHCP servers. | |
| **Block Non-Source IP** | **Yes** / **No** | Click **Yes** to have the interface forward traffic only if the source IP address of the traffic matches the interface's IP prefix range. Click **No** to allow other traffic. | |
| **Bandwidth Upstream** | For Cisco vEdge devices and vManage:<br>For transmitted traffic, set the bandwidth above which to generate notifications.<br>Range: 1 through (232 / 2) – 1 kbps | | |
| **Bandwidth Downstream** | For Cisco vEdge devices and vManage:<br>For received traffic, set the bandwidth above which to generate notifications.<br>Range: 1 through (232 / 2) – 1 kbps | | |

To save the feature template, click **Save**.

### CLI Equivalent

```
vpn vpn-id
  interface interface-name
    bandwidth-downstream kbps
    bandwidth-upstream kbps
    block-non-source-ip
    description text
    dhcp-helper ip-address
    (ip address ipv4-prefix/length| ip dhcp-client [dhcp-distance number])
    (ipv6 address ipv6-prefix/length | ipv6 dhcp-client [dhcp-distance number]
[dhcp-rapid-commit])
    secondary-address ipv4-address
    [no] shutdown
```

# Create a Tunnel Interface

On Cisco vEdge device s, you can configure up to four tunnel interfaces. This means that each Cisco vEdge device router can have up to four TLOCs. On Cisco vSmart Controllers and Cisco vManage, you can configure one tunnel interface.

For the control plane to establish itself so that the overlay network can function, you must configure WAN transport interfaces in VPN 0. The WAN interface will enable the flow of tunnel traffic to the overlay. You can add other parameters shown in the table below only after you configure the WAN interface as a tunnel interface.

To configure a tunnel interface, select the **Interface Tunnel** tab and configure the following parameters:

| Parameter Name | Cisco vEdge Devices Only | Description |
|---|---|---|
| Tunnel Interface | No | Click **On** to create a tunnel interface. |
| Color | No | Select a color for the TLOC. |
| Control Connection | Yes | If the Cisco vEdge device has multiple TLOCs, click **No** to have the tunnel not establish a TLOC. The default is On, which establishes a control connection for the TLOC. |
| Maximum Control Connections | Yes | Specify the maximum number of Cisco vSmart Controllers that the WAN tunnel interface can connect to. To have the tunnel establish no control connections, set the number to 0.<br>Range: 0 through 8<br>Default: 2 |
| Cisco vBond Orchestrator As Stun Server | Yes | Click On to enable Session Traversal Utilities for NAT (STUN) to allow the tunnel interface to discover its public IP address and port number when theCisco vEdge device router is located behind a NAT. |
| Exclude Controller Group List | Yes | Set the Cisco vSmart Controllers that the tunnel interface is not allowed to connect to.<br>Range: 0 through 100 |
| vManage Connection Preference | Yes | Set the preference for using a tunnel interface to exchange control traffic with the vManage NMS.<br>Range: 0 through 8<br>Default: 5 |
| Port Hop | No | Click **On** to enable port hopping, or click **Off** to disable it. If port hopping is enabled globally, you can disable it on an individual TLOC (tunnel interface). To control port hopping on a global level, use the System configuration template.<br>Default: Enabled<br>vManage NMS and Cisco vSmart Controller default: Disabled |
| Low-Bandwidth Link | Yes | Select to characterize the tunnel interface as a low-bandwidth link. |
| Allow Service | No | Select **On** or **Off** for each service to allow or disallow the service on the interface. |

To configure additional tunnel interface parameters, click **Advanced Options**:

| Parameter Name | Cisco vEdge devices Only | Description |
|---|---|---|
| GRE | Yes | Use GRE encapsulation on the tunnel interface. By default, GRE is disabled.<br>If you select both IPsec and GRE encapsulations, two TLOCs are created for the tunnel interface that have the same IP addresses and colors, but that differ by their encapsulation. |
| IPsec | Yes | Use IPsec encapsulation on the tunnel interface. By default, IPsec is enabled.<br>If you select both IPsec and GRE encapsulations, two TLOCs are created for the tunnel interface that have the same IP addresses and colors, but that differ by their encapsulation. |
| IPsec Preference | Yes | Specify a preference value for directing traffic to the tunnel. A higher value is preferred over a lower value.<br>Range: 0 through 4294967295<br>Default: 0 |
| IPsec Weight | Yes | Enter a weight to use to balance traffic across multiple TLOCs. A higher value sends more traffic to the tunnel.<br>Range: 1 through 255<br>Default: 1 |
| Carrier | No | Select the carrier name or private network identifier to associate with the tunnel.<br>Values: carrier1, carrier2, carrier3, carrier4, carrier5, carrier6, carrier7, carrier8, default<br>Default: default |
| Bind Loopback Tunnel | Yes | Enter the name of a physical interface to bind to a loopback interface. |
| Last-Resort Circuit | Yes | Select to use the tunnel interface as the circuit of last resort. |
| NAT Refresh Interval | No | Enter the interval between NAT refresh packets sent on a DTLS or TLS WAN transport connection.<br>Range: 1 through 60 seconds<br>Default: 5 seconds |
| Hello Interval | No | Enter the interval between Hello packets sent on a DTLS or TLS WAN transport connection.<br>Range: 100 through 10000 milliseconds<br>Default: 1000 milliseconds (1 second) |

| Parameter Name | Cisco vEdge devices Only | Description |
|---|---|---|
| Hello Tolerance | No | Enter the time to wait for a Hello packet on a DTLS or TLS WAN transport connection before declaring that transport tunnel to be down.<br>Range: 12 through 60 seconds<br>Default: 12 seconds |

To save the feature template, click **Save**.

# Configure DNS and Static Hostname Mapping

To configure DNS addresses and static hostname mapping, click the **DNS** tab and configure the following parameters:

<table>
<tr><th>Parameter Name</th><th>Options</th><th>Description</th></tr>
<tr><td><b>Primary DNS Address</b></td><td colspan="2">Select either <b>IPv4</b> or <b>IPv6</b>, and enter the IP address of the primary DNS server in this VPN.</td></tr>
<tr><td rowspan="4"><b>New DNS Address</b></td><td colspan="2">Click <b>New DNS Address</b> and enter the IP address of a secondary DNS server in this VPN. This field appears only if you have specified a primary DNS address.</td></tr>
<tr><td><b>Mark as Optional Row</b></td><td>Check <b>Mark as Optional Row</b> to mark this configuration as device-specific. To include this configuration for a device, enter the requested variable values when you attach a device template to a device, or create a template variables spreadsheet to apply the variables.</td></tr>
<tr><td><b>Hostname</b></td><td>Enter the hostname of the DNS server. The name can be up to 128 characters.</td></tr>
<tr><td><b>List of IP Addresses</b></td><td>Enter up to eight IP addresses to associate with the hostname. Separate the entries with commas.</td></tr>
<tr><td colspan="3">To save the DNS server configuration, click <b>Add</b>.</td></tr>
</table>

To save the feature template, click **Save**.

### CLI Equivalent

```
vpn vpn-id
  dns ip-address (primary | secondary)
  host hostname ip ip-address
```

# Configure Segmentation Using CLI

## Configure VPNs Using CLI

### Configure Transport VPN on a vEdge Router

On a vEdge router, the interfaces in VPN 0 connect to a WAN transport network. You must configure at least one tunnel interface on a vEdge router so that it can join the control plane and be part of the overlay network. If is not configured, that router cannot participate in the overlay network.

For a tunnel connection on a vEdge router, you must configure the three components of a TLOC—the interface's IP address and the tunnel's color and encapsulation. An OMP session runs over each tunnel connection, and it is OMP that distributes the device TLOCs to vSmart controllers. The controllers use the TLOCs to determine the overlay network topology and to determine the best routing paths across the overlay network. A vEdge router can have up to four TLOCs, so you can configure more than one tunnel connection.

In the transport VPN (VPN 0), vEdge routers support dual stack. To enable dual stack, configure an IPv4 address and an IPv6 address on the tunnel interface. The vEdge router learns from the vSmart controller whether a destination supports IPv4 or IPv6 addresses. When forwarding traffic, the router chooses either the IPv4 or the IPv6 TLOC based on the destination address.

To configure VPN 0 on a vEdge router:

1. Configure the WAN transport interface:

   ```
   vEdge(config)# vpn 0 interface interface-name
   vEdge(config-interface)#
   ```

   In the most common cases, *interface-name* is the name of a physical Gigabit Ethernet interface (**ge** *port / slot*). The interface name can also be **gre** *number*, **ipsec** *number*, **loopback** *string*, **natpool** *number*, or **ppp** *number*.

2. Configure a static IPv4 address for the interface:

   ```
   vEdge(config-interface)# ip address prefix/length
   vEdge(config-interface) #
   ```

   Or you can enable DHCP on the interface so that the interface learn its IP address dynamically:

   ```
   vEdge(config-interface)# ip dhcp-client [dhcp-distance number]
   vEdge(config-interface)#
   ```

   When an interface learns its IPv4 address from a DHCP server, it can also learn routes from the server. By default, these routes have an administrative distance of 1, which is the same as static routes. To change the default value, include the **dhcp-distance** option, specifying a distance from 1 through 255.

3. To enable dual stack, configure a static IPv6 address for the interface:

   ```
   vEdge(config-interface)# ipv6 address prefix/length
   vEdge(config-interface)#
   ```

   Or you can enable DHCPv6 on the interface so that the interface learn its IP address dynamically:

   ```
   vEdge(config-interface)# ipv6 dhcp-client [dhcp-distance number] [dhcp-rapid-commit]
   vEdge(config-interface)#
   ```

   When an interface learns its IPv6 address from a DHCPv6 server, it can also learn routes from the server. By default, these routes have an administrative distance of 1, which is the same as static routes. To change the default value, include the **dhcp-distance** option, specifying a distance from 1 through 255. To speed up the assignment of IPv6 addresses, include the **dhcp-rapid-commit** option.

4. Enable the interface:

```
vEdge(config-interface)# no shutdown
```

5. Configure the WAN transport tunnel connection:

```
vEdge(config-interface)# tunnel-interface
vEdge(config-tunnel-interface)#
```

6. Configure a color for the tunnel connection as an identifier for the tunnel:

```
vEdge(config-tunnel-interface)# color color
vEdge(config-tunnel-interface)#
```

*color* can be **3g**, **biz-internet**, **blue**, **bronze**, **custom1**, **custom2**, **custom3**, **default**, **gold**, **green**, **lte**, **metro-ethernet**, **mpls**, **private1** through **private6**, **public-internet**, **red**, and **silver**. The default color is **default**. The colors **metro-ethernet**, **mpls**, and **private1** through **private6** are referred to as *private colors*, because they use private addresses to connect to the remote side vEdge router in a private network. You can use these colors in a public network provided that there is no NAT device between the local and remote vEdge routers.

7. Configure the encapsulation to use on tunnel connection:

```
vEdge(config-tunnel-interface)# encapsulation (gre | ipsec)
vEdge(config-tunnel-interface)#
```

To configure both IPsec and GRE encapsulation, include two **encapsulation** commands. Note that if you do this, you are creating two TLOCs that have the same IP addresses and colors, but that have different encapsulation.

8. Configure any other properties specific to the tunnel interface, the interface, or VPN 0.

9. If you have a multi-TLOC environment, configure additional tunnel interfaces.

10. Enable DNS service in the VPN by configuring the IP address of a DNS server reachable from VPN 0:

```
vEdge(config-vpn-0)# dns ip-address (primary | secondary)
```

The address can be either an IPv4 or IPv6 address. By default, the IP address is for the primary DNS server.

11. If desired, configure IPv4 and IPv6 static routes in VPN 0:

```
vEdge(config-vpn-0)# ip route prefix/length next-hop [administrative-distance]
vEdge(config-vpn-0)# ipv6 route prefix/length next-hop [administrative-distance]
```

12. Activate the configuration:

```
vEdge(config)# commit
```

To display interface information, use the **show interface** command for IPv4 interfaces and **show ipv6 interfaces** for IPv6 interfaces. To display information about DHCP and DHCPv6 servers, use the **show dhcp interface** and **show ipv6 dhcp interface** commands.

When you are troubleshooting routing and forwarding problems on a vEdge router, you can configure the router to perform route consistency checks, to determine whether the routes in the router's route and forwarding tables are consistent:

```
vEdge(config-system)#route-consistency-check
```

This command checks only IPv4 routes. Route consistency checking requires a large amount of device CPU, so it is recommended that you enable it only when you trouble shooting an issue and that you disable it at other times.

Here is an example of a VPN 0 configuration, where **interface ge0/0** is the WAN transport interface. This example shows that dual stack is enabled on the router, because the tunnel interface has both an IPv4 and an IPv6 address. Notice that the remaining seven device interfaces are part of VPN 0, because we have not yet configured any other VPNs. Also notice that the management interface is not present in VPN 0.

```
vpn 0
 interface ge0/0
  ip address 10.0.0.8/24
  ipv6 address fd00:1234::/16
  tunnel-interface
   color biz-internet
   encapslation ipsec
   allow-service dhcp
   allow-service dns
   allow-service icmp
   no allow-service sshd
   no allow-service ntp
   no allow-service stun
  !
  no shutdown
 !
 interface ge0/1
  shutdown
 !
 interface ge0/2
  shutdown
 !
 interface ge0/3
  shutdown
 !
 interface ge0/4
  shutdown
 !
 interface ge0/5
  shutdown
 !
 interface ge0/6
  shutdown
 !
 interface ge0/7
  shutdown
 !
!
```

An interface can participate only in one VPN. So in an initial configuration, when VPN 0 is the only VPN that is configured, all the device's interfaces are present, by default, in VPN 0 (as shown in the output above). Then, when you create other VPNs to carry data traffic and configure interfaces in those VPNs, the interfaces used in the other VPNs are automatically removed from VPN 0. Here is an example in which **interface ge0/3** is used for VPN 1, so it has been automatically removed from the configuration of VPN 0:

```
vpn 0
interface ge0/0
  ip address 10.0.0.8/24
  tunnel-interface
   color biz-internet
   encapsulation ipsec
   allow-service dhcp
   allow-service dns
   allow-service icmp
   no allow-service sshd
   no allow-service ntp
   no allow-service stun
  !
  no shutdown
!
interface ge0/1
  shutdown
!
interface ge0/2
  shutdown
!
interface ge0/4
  shutdown
!
interface ge0/5
  shutdown
!
interface ge0/6
  shutdown
!
interface ge0/7
  shutdown
!
!
vpn 1
router
  ospf
   redistribute omp route-policy test-policy
   area 0
    interface ge0/3
    exit
   exit
  !
!
interface ge0/3
  ip address 10.10.10.1/24
  no shutdown
!
!
```

When you configure subinterfaces in a VPN that carries data traffic (that is, not VPN 0 and not VPN 512), the main interface must be configured with the **no shutdown** command so that it is enabled, and the main interface remains in VPN 0 once you configure the subinterface. For example, if in the VPN 1 configuration, you were to configure OSPF on VLAN 1, you can see that **interface ge0/3** remains present in VPN 0, while the subinterface **interface ge0/3.1** is used in VPN1:

```
vpn 0
 dns 1.2.3.4 primary
 interface ge0/0
  address 10.0.0.8/24
  tunnel-interface
   preference 100
   allow-service dhcp
   allow-service dns
   allow-service icmp
   allow-service sshd
   allow-service ntp
   allow-service stun
  !
  no shutdown
 !
 interface ge0/1
  shutdown
 !
 interface ge0/2
  shutdown
 !
 interface ge0/3
  no shutdown
 !
 interface ge0/4
  shutdown
 !
 interface ge0/5
  shutdown
 !
 interface ge0/6
  shutdown
 !
 interface ge0/7
  shutdown
 !
!
vpn 1
router
  ospf
   redistribute omp route-policy test-policy
   area 0
    interface ge0/3.1
    exit
   exit
  !
 !
interface ge0/3.1
  ip address 10.10.10.1/24
  no shutdown
 !
!
```

### Configure the Transport VPN on a vSmart Controller

Because vSmart controllers are responsible for determining the best routes through the overlay network (based on the TLOCs it learns and based on centralized policies), they handle only control plane traffic, in VPN 0. A vSmart controller can have only one interface in VPN 0, for which you set an IP address and you create a tunnel connection. This tunnel connection acts a control plane tunnel termination point.

In the transport VPN (VPN 0), vEdge routers support dual stack. To enable dual stack, configure an IPv4 address and an IPv6 address on the tunnel interface. The vEdge router learns from the vSmart controller whether a destination supports IPv4 or IPv6 addresses. When forwarding traffic, the router chooses either the IPv4 or the IPv6 TLOC based on the destination address.

To configure VPN 0 on a vSmart controller:

1. Configure the WAN transport interface:

   ```
   vSmart(config)# vpn 0 interface interface-name
   vSmart(config-interface)#
   ```

   *interface-name* is the name of a virtual Ethernet interface (**eth** *number*).

2. Configure a static IPv4 address for the interface:

   ```
   vSmart(config-interface)# ip address prefix/length
   vSmart(config-interface)#
   ```

   Or you can enable DHCP on the interface so that the interface learn its IP address dynamically:

```
vSmart(config-interface)# ip dhcp-client [dhcp-distancenumber]
vSmart(config-interface)#
```

When an interface learns its IPv4 address from a DHCP server, it can also learn routes from the server. By default, these routes have an administrative distance of 1, which is the same as static routes. To change the default value, include the **dhcp-distance** option, specifying a distance from 1 through 255.

3. To enable dual stack, configure a static Pv6 address for the interface:

   ```
   vSmart(config-interface)# ipv6 address prefix/length
   vSmart(config-interface)#
   ```

   Or you can enable DHCPv6 on the interface so that the interface learn its IP address dynamically:

   ```
   vSmart(config-interface)# ipv6 dhcp-client [dhcp-distance number] [dhcp-rapid-commit]
   vSmart(config-interface)#
   ```

   When an interface learns its IPv6 address from a DHCPv6 server, it can also learn routes from the server. By default, these routes have an administrative distance of 1, which is the same as static routes. To change the default value, include the **dhcp-distance** option, specifying a distance from 1 through 255. To speed up the assignment of IPv6 addresses, include the **dhcp-rapid-commit** option.

4. Enable the interface:

   ```
   vSmart(config-interface)# no shutdown
   ```

5. Enable DNS service in the VPN by configuring the IP address of a DNS server reachable from VPN 0:

   ```
   vSmart(config-vpn-0)# dns ip-address (primary | secondary)
   ```

   The address can be either an IPv4 or IPv6 address. By default, the IP address is for the primary DNS server.

6. If desired, configure IPv4 and IPv6 static routes in VPN 0:

   ```
   vSmart(config-vpn-0)# ip route prefix/length next-hop [administrative-distance]
   vSmart(config-vpn-0)# ipv6 route prefix/length next-hop [administrative-distance]
   ```

7. Configure any other properties specific to the tunnel interface, the interface, or VPN 0.

8. Activate the configuration:

   ```
   vSmart(config)# commit
   ```

To display interface information, use the **show interface** command for IPv4 interfaces and **show ipv6 interfaces** for IPv6 interfaces. To display information about DHCP and DHCPv6 servers, use the **show dhcp interface** and **show ipv6 dhcp interface** commands.

Here is an example of a VPN 0 configuration on a vSmart controller:

```
vSmart# show running-config vpn 0
vpn 0
 dns 1.2.3.4 primary
 interface eth0
  ip dhcp-client
  no shutdown
 !
 interface eth1
  ip address 10.0.5.19/24
  tunnel-interface
   allow-ssh
   allow-icmp
  !
  no shutdown
 !
```

```
 ip route 0.0.0.0/0 10.0.5.13
!
```

### Configure Data Traffic Exchange across Private WANs

When a vEdge router is connected to a private WAN, such as an MPLS or a metro Ethernet network, the carrier hosting the private network does not advertise the IP address of that vEdge router over the internet. (This IP address is associated with the TLOC on that vEdge router.) This means that remote vEdge routers are not able to learn how to reach that router and hence are not able to exchange data traffic with it directly over the private network.

To allow the vEdge router behind the private network to communicate directly over the private WAN with other vEdge routers, you direct the data traffic to a loopback interface rather than to the actual physical WAN interface. The overlay network can then advertise that the local router is reachable via its loopback address. To make it possible for the data traffic to actually be transmitted out the WAN interface, you bind the loopback interface to the physical WAN interface to the private network.

To configure VPN 0 so that it carries data traffic across private WANs:

**1.** Configure the loopback interface, assigning it an IP address:

```
vEdge(config)# vpn 0  loopback
number ip address prefix/length
vEdge(config-loopback)# no shutdown
```

**2.** Configure the loopback interface to be a transport interface:

```
vEdge(config-loopback)# tunnel-interface
```

**3.** Set the color of the loopback interface to be one of the primatel colors—**metro-ethernet**, **mpls**, and **private1** through **private6**. You must configure this same color on the loopback interfaces of all vEdge routers in the same private LAN.

```
vEdge(config-tunnel-interface)# color color
```

Use the **show interface** command to check that the loopback interface in configured properly, as a transport interface with the proper IP address and color.

If a single vEdge router is connected to two (or more) different private networks, create a loopback interface for each private network, associate a carrier name with the interface so that the router can distinguish between the two private WANs, and "bind" the loopback interface to the physical interface that connects to the appropriate private WAN:

**1.** Configure the loopback interface, assigning it an IP address:

```
vEdge(config)# vpn 0
loopback
number
ip address prefix/length
vEdge(config-loopback)# no shutdown
```

**2.** Configure the loopback interface to be a transport interface and bind it to a physical interface:

```
vEdge(config-loopback)# tunnel-interface bind
ge
slot/port
```

**3.** Configure a carrier name and TLOC color on the loopback interface:

```
vEdge(config-tunnel-interface)# carrier carrier-name
vEdge(config-tunnel-interface)# color color
```

4. On the physical interface, configure its IP address, and enable it:

   ```
   vEdge(config)# vpn 0 interface
   ge
   slot/port
   ip address prefix/length
   vEdge(config-ge)# no shutdown
   ```

### Configure the Management VPN (VPN 512)

In the Cisco SD-WAN overlay network, VPN 512 is the network management VPN. It carries out-of-band management traffic in the overlay network. VPN 512 is configured and enabled by default on all Cisco SD-WAN devices. It contains the interface used for management traffic. For vEdge routers, this interface is generally a Gigabit Ethernet (**ge**) interface, and for other Cisco SD-WAN devices it is an **eth** interface. DHCP is enabled by default on the management interface. The default configuration for VPN 512 on a vEdge router looks like this:

```
vpn 512
 interface ge0/0
  ip dhcp-client
  no shutdown
 !
!
```

VPN 512 must be present on all Cisco SD-WAN devices so that they are always reachable on the network. You can configure additional parameters for VPN 512 if you choose.

### Configure VPNs To Carry Data Traffic

VPNs other than VPN 0 and VPN 512 are used to carry data traffic across the overlay network. These VPNs are sometimes referred to as *service-side VPNs*. For these VPNs to operate, each one must have an operational interface (or subinterface). The remainder of what you configure in these VPNs depends on your network needs. You configure features specific for the user segment, such as BGP and OSPF routing, VRRP, QoS, traffic shaping, and policing.

To create a data traffic VPN:

1. Configure the VPN:

   ```
   vEdge(config)# vpn
   number
   vEdge(config-vpn)#
   ```

   The VPN number can be in the range 1 through 511, and 513 through 65535.

2. Configure at least one interface in the VPN and its IP address:

   ```
   vEdge(config-vpn)# interface
    interface-name
    ip address
    address/prefix
   vEdge(config-interface)#
   ```

   The interface name has the format **ge** *slot*/*port*, where the slot is generally 0 through 7 (depending on the device) and the port is 0 through 8. If you are configuring VLANs, specify a subinterface name in the format **ge** *slot*/*port* **.** *vlan*, where the VLAN number can be in the range 1 through 4094. (VLAN numbers 0 and 4095 are reserved.) The interface name can also be **gre** *number*, **ipsec** *number*, **loopback** *string*, **natpool** *number*, or **ppp** *number*.

3. Activate the interface:

```
vEdge(config-interface)# no shutdown
```

4. Enable DNS service in the VPN by configuring the IP address of a DNS server reachable from that VPN:

   ```
   vEdge(config-vpn)#  dns ip-address
   ```

5. If desired, configure IPv4 static routes in the VPN:

   ```
   vEdge(config-vpn)# ip route  prefix
   /
   length next-hop [administrative-distance]
   ```

6. Configure any other properties specific to the interface or to VPN.

7. Activate the configuration:

   ```
   vEdge(config)# commit
   ```

Here is an example of a configuration for VPN 1:

```
vpn 1
 dns 1.2.3.4 primary
 router
  ospf
   redistribute omp route-policy test-policy
   area 0
    interface ge0/3
    exit
   exit
  !
 !
 interface ge0/3
  ip address 10.10.10.1/24
  no shutdown
 !
!
```

### Dual-Stack Operation

When a Cisco SD-WAN device establishes an IPsec tunnel for control traffic between a local TLOC and a remote TLOC, or when a device establishes a BFD tunnel for data plane traffic between a local and a remote TLOC, an IPv6 tunnel is established in the following situations:

- The local device has only an IPv6 address, and the remote device has an IPv6 address.
- The remote device has only an IPv6 address, and the local device has an IPv6 address.

If both the local and remote devices have IPv4 addresses, IPsec and BFD always establish an IPv4 tunnel.

## Segmentation (VPNs ) Configuration Examples

Some straightforward examples of creating and configuring VPNs to help you understand the configuration procedure for segmenting networks.

### Create Basic VPNs

Creating the basic VPNs required by Cisco SD-WAN devices is a simple, straightforward process, consisting of these steps:

1. On the vEdge router:

- Create a VPN instance for the transport VPN. VPN 0 is reserved for the transport VPN.
- Create a VPN instance for the management VPN. VPN 512 is reserved for the management VPN.
- Create a VPN instance to use for routing.

2. On the vSmart controller:
    - Create a VPN instance for the transport VPN. VPN 0 is reserved for the transport VPN.
    - Create a VPN instance for the management VPN. VPN 512 is reserved for the management VPN.
    - Optionally, create policies to influence routing and access control within the VPN.

### Configuration on the vEdge Router

To create the basic VPNs on a vEdge router, you configure VPN 0 for transport, VPN 512 for management, and a third VPN (here, VPN 1) for carrying data traffic:

1. First, configure general system parameters:

```
vEdge(config)# system host-name host-name
vEdge(config-system)# system-ip ip-address
vEdge(config-system)# domain-id domain-id
vEdge(config-system)# site-id site-id
vEdge(config-system)# vbond (dns-name | ip-address)
```

2. In VPN 0, which is the transport VPN, configure the interface to the WAN transport cloud, to establish reachability between the vEdge router and the vSmart controller, and between vEdge routers:

    a. Configure an IP address for the interface:

    ```
    vEdge(config-interface)# vpn 0 interface interface-name ip address prefix/length
    ```

    b. Enable the interface:

    ```
    vEdge(config-interface)# no shutdown
    ```

    c. Enable a transport tunnel interface to carry control and data traffic, and configure the color and encapsulation for the tunnel:

    ```
    vEdge(config-interface)# tunnel-interface
    vEdge(config-tunnel-interface)# encapsulation (gre | ipsec)
    vEdge(config-tunnel-interface)# color color
    ```

    d. Configure a default route for the VPN:

    ```
    vEdge(config-vpn-0)# ip route 0.0.0.0/0 ip-address
    ```

3. Configure a VPN for data traffic:

    a. Create the VPN and assign it a identifier number. The identifier can be any number except 0 and 512.

    ```
    vEdge(config)# vpn vpn-id
    ```

    b. Add an interface to the VPN:

    ```
    vEdge(config-vpn-number)# interface interface-name ip address ip-address
    ```

    c. Enable the interface:

    ```
    vEdge(config-vpn-number)# no shutdown
    ```

4. Configure unixAR routing in the VPN. See Configuring Basic Unicast Overlay Routing  for more information.

5. Activate the configuration:

   ```
   vEdge(config)# commit
   ```

Here is the full configuration on the vEdge router:

```
system                    # Configure general system parameters
 host-name vedge
 system-ip 1.0.0.2
 domain-id 1
 site-id   20
 vbond 10.2.6.1
!
vpn 0                              # Create the tunnel interface and allow
  interface ge 0/0                   reachability to vSmart in transport VPN
    ip address 10.2.6.11/24
    tunnel-interface
      color default
      encapsulation ipsec
   !
   no shutdown
   !
 ip route 0.0.0.0/0 10.2.6.12
!
vpn 1                              # Create new VPN, add interfaces and routing
 interface ge 0/1
  ip address 10.100.1.1/24
  no shutdown
 !
!
router
  bgp 20
   neighbor 10.100.1.2
    no shutdown
    remote-as 20
    address-family ipv4_unicast
    !
   !
  !
 !
vpn 512
  interface mgmt0
    ip dhcp-client
    no shutdown
  !
!
```

### Configuration on the vSmart Controller

On the vSmart controller, you configure general system parameters and the two VPNs—VPN 0 for WAN transport and VPN 512 for network management—as you did for the Cisco XE SD-WAN device. Also, you generally create a centralized control policy that controls how the VPN traffic is propagated through the rest of the network. In this particular example, we create a central policy, shown below, to drop unwanted prefixes from propagating through the rest of the network. You can use a single vSmart policy to enforce policies throughout the network.

Here are the steps for creating the control policy on the vSmart controller:

1. Create a list of sites IDs for the sites where you want to drop unwanted prefixes:

```
vSmart(config)# policy lists site-list 20-30 site-id 20
vSmart(config-site-list-20-30)# site-id 30
```

2. Create a prefix list for the prefixes that you do not want to propagate:

```
vSmart(config)# policy lists prefix-list drop-list ip-prefix 10.200.1.0/24
```

3. Create the control policy:

```
vSmart(config)# policy control-policy drop-unwanted-routes sequence 10 match route
prefix-list drop-list
vSmart(config-match)# top
vSmart(config)# policy control-policy drop-unwanted-routes sequence 10 action reject
vSmart(config-action)# top
vSmart(config)# policy control-policy drop-unwanted-routes sequence 10 default-action
accept
vSmart(config-default-action)# top
```

4. Apply the policy to prefixes inbound to the vSmart controller:

```
vSmart(config)# apply-policy site-list 20-30 control-policy drop-unwanted-routes in
```

Here is the full policy configuration on the vSmart controller:

```
apply-policy
 site-list 20-30
  control-policy drop-unwanted-routes in
 !
!
policy
 lists
  site-list 20-30
   site-id 20
   site-id 30
  !
  prefix-list drop-list
   ip-prefix 10.200.1.0/24
  !
 !
 control-policy drop-unwanted-routes
  sequence 10
   match route
    prefix-list drop-list
   !
   action reject
   !
  !
  default-action accept
 !
!
```

### Control VPN Membership

You can create VPNs just at the sites of interest and can then keep them hidden so that the rest of the network does not even know about them and the routes from them. Such a network design provides a great deal of traffic isolation and flexibility. However, there might be cases where the network administrator might want to explicitly disallow the creation of VPNs on the vEdge router. An example is in a B2B partnership, when the vEdge router is not located at the customer premise. For these situations, the network administrator can choose to allow only certain VPNs on these vEdge routers. Effectively, you are controlling membership in the VPN.

You control VPN membership policy at the vSmart controller. In the example here, you create a policy that explicitly disallows VPN 1 at sites 20 and 30:

```
apply-policy
 site-list 20-30
  vpn-membership disallow-vpn1
 !
!
policy
 lists
  site-list 20-30
   site-id 20
   site-id 30
  !
 !
 vpn-membership disallow-vpn1
  sequence 10
   match vpn-id 1
   action reject
   !
  !
  default-action accept
 !
!
```

### Leak Routes across VPNs

In some situations it is desirable to leak routes from one VPN into another. Some examples include extranets, where you are making a portion of your intranet available to users outside your organization, B2B partnerships, and the network transition that occurs during a merger or acquisition. To leak routes across VPNs, you create a leaking control policy on the vSmart controller, a design that allows you to control route leaking from a central point in the network.

In this example, we create a control policy that allows an enterprise's VPN to import routes from a VPN list. Specifically, we:

- Create a control policy to match routes from a list of VPNs. Here, sequence 10 of the policy matches all routes from the VPNs of all business partners (BPs). The business partner VPN IDs are listed in the **All-BPs** list.
- Accept routes that match this policy, and import the prefixes into a new VPN called **Enterprise-BP**.
- Apply this policy towards the BP sites on vRoutes inbound to the vSmart controller.

```
policy
  lists
    site-list BP-Sites
      site-id 10
      site-id 20
    vpn-list All-BPs
      vpn 100
      vpn 101
    vpn-list Enterprise-BP
      vpn 200
  control-policy import-BPs-to-Enterprise
    sequence 10
     match route
      vpn-list All-BPs
     !
     action accept
      export-to vpn-list Enterprise-BP
      !
      !
     !
    default-action accept
   !
!
apply-policy
 site-list BP-Sites
  control-policy import-BPs-to-Enterprise in
 !
```

This policy matches all routes from all VPNs in the **All-BPs** VPN lists and populates these prefixes into the VPNs in the Enterprise-BP list. The routing table of the Enterprise-BP VPN will now contain all the prefixes of the BPs.

One advantage of importing routes in this way is access control. Keeping each BP in a separate VPN and creating an extranet policy ensures that the BPs cannot talk to each other.

# Use Case: Exchange Data Traffic within a Single Private WAN

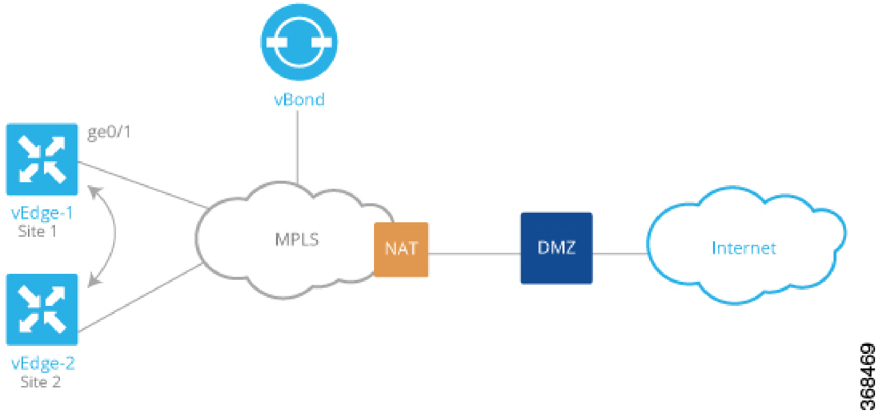
### Allow Data Traffic Exchange across Private WANs

When the WAN network to which a vEdge router is connected is a private network, such as an MPLS or a metro Ethernet network, and when the carrier hosting the private network does not advertise the router's IP address, remote vEdge routers on the same private network but at different sites can never learn how to reach that router and hence are not able to exchange data traffic with it by going only through the private network. Instead, the remote routers must route data traffic through a local NAT and over the Internet to a vBond orchestrator, which then provides routing information to direct the traffic to its destination. This process can add significant overhead to data traffic exchange, because the vBond orchestrator may physically be located at a different site or a long distance from the two vEdge routers and because it may be situated behind a DMZ.

To allow vEdge routers at different overlay network sites on the private network to exchange data traffic directly using their private IP addresses, you configure their WAN interfaces to have one of the private colors, **metro-ethernet**, **mpls**, and **private1** through **private6**. Of these private colors, the WAN interfaces on the vEdge routers must be marked with the same color so that they can exchange data traffic.

### Exchange Data Traffic within a Single Private WAN

To illustrate the exchange of data traffic across private WANs, let's look at a simple topology in which two vEdge routers are both connected to the same private WAN. The following figure shows that these two vEdge routers are connected to the same private MPLS network. The vEdge-1 router is located at Site 1, and vEdge-2 is at Site 2. Both routers are directly connected to PE routers in the carrier's MPLS cloud, and you want both routers to be able to communicate using their private IP addresses.



This topology requires a special configuration to allow traffic exchange using private IP addresses because:

- The vEdge routers are in different sites; that is, they are configured with different site IDs.
- The vEdge routers are directly connected to the PE routers in the carrier's MPLS cloud.
- The MPLS carrier does not advertise the link between the vEdge router and its PE router.

To be clear, if the situation were one of the following, no special configuration would be required:

- vEdge-1 and vEdge-2 are configured with the same site ID.
- vEdge-1 and vEdge-2 are in different sites, and the vEdge router connects to a CE router that, in turn, connects to the MPLS cloud.
- vEdge-1 and vEdge-2 are in different sites, the vEdge router connects to the PE router in the MPLS cloud, and the private network carrier advertises the link between the vEdge router and the PE router in the MPLS cloud.
- vEdge-1 and vEdge-2 are in different sites, and you want them to communicate using their public IP addresses.

In this topology, because the MPLS carrier does not advertise the link between the vEdge router and the PE router, you use a loopback interface on the each vEdge router to handle the data traffic instead of using the physical interface that connects to the WAN. Even though the loopback interface is a virtual interface, when you configure it on the vEdge router, it is treated like a physical interface: the loopback interface is a terminus for both a DTLS tunnel connection and an IPsec tunnel connection, and a TLOC is created for it.

This loopback interface acts as a transport interface, so you must configure it in VPN 0.

For the vEdge-1 and vEdge-2 routers to be able to communicate using their private IP addresses over the MPLS cloud, you set the color of their loopback interfaces to be the same and to one of private colors—**metro-ethernet**, **mpls**, and **private1** through **private6**.

Here is the configuration on vEdge-1:

```
vedge-1(config)# vpn 0
vedge-1(config-vpn-0)# interface loopback1
vedge-1(config-interface-loopback1)# ip address 172.16.255.25/32
vedge-1(config-interface-loopback1)# tunnel-interface
vedge-1(config-tunnel-interface)# color mpls
vedge-1(config-interface-tunnel-interface)# exit
vedge-1(config-tunnel-interface)# no shutdown
```

```
vedge-1(config-tunnel-interface)# commit and-quit
vedge-1# show running-config vpn 0
...
 interface loopback1
  ip-address 172.16.255.25/32
  tunnel-interface
   color mpls
  !
  no shutdown
 !
```

On vEdge-2, you configure a loopback interface with the same tunnel interface color that you used for vEdge-1:

```
vedge-2# show running-config vpn 0
vpn 0
 interface loopback2
  ip address 172. 17.255.26/32
  tunnel-interface
   color mpls
  no shutdown
 !
```

Use the **show interface** command to verify that the loopback interface is up and running. The output shows that the loopback interface is operating as a transport interface, so this is how you know that it is sending and receiving data traffic over the private network.

```
vedge-1# show interface

                                  IF      IF                                                                     TCP
                                  ADMIN   OPER    ENCAP                                      SPEED         MSS                 RX       TX
VPN  INTERFACE  IP ADDRESS        STATUS  STATUS  TYPE   PORT TYPE  MTU   HWADDR             MBPS   DUPLEX ADJUST  UPTIME      PACKETS  PACKETS
-------------------------------------------------------------------------------------------------------------------------------------------------
0    ge0/0      10.1.15.15/24     Up      Up      null   transport  1500  00:0c:29:7d:1e:fe  10     full   0       0:07:38:49  213199   243908
0    ge0/1      10.1.17.15/24     Up      Up      null   service    1500  00:0c:29:7d:1e:08  10     full   0       0:07:38:49  197      3
0    ge0/2      -                 Down    Down    null   service    1500  00:0c:29:7d:1e:12  -      -      0       -           1        1
0    ge0/3      10.0.20.15/24     Up      Up      null   service    1500  00:0c:29:7d:1e:1c  10     full   0       0:07:38:49  221      27
0    ge0/6      57.0.1.15/24      Up      Up      null   service    1500  00:0c:29:7d:1e:3a  10     full   0       0:07:38:49  196      3
0    ge0/7      10.0.100.15/24    Up      Up      null   service    1500  00:0c:29:7d:1e:44  10     full   0       0:07:44:47  783      497
0    loopback1  172.16.255.25/32  Up      Up      null   transport  1500  00:00:00:00:00:00  10     full   0       0:00:00:20  0        0
0    system     172.16.255.15/32  Up      Up      null   loopback   1500  00:00:00:00:00:00  10     full   0       0:07:38:25  0        0
1    ge0/4      10.20.24.15/24    Up      Up      null   service    1500  00:0c:29:7d:1e:26  10     full   0       0:07:38:46  27594    27405
1    ge0/5      56.0.1.15/24      Up      Up      null   service    1500  00:0c:29:7d:1e:30  10     full   0       0:07:38:46  196      2
512  eth0       10.0.1.15/24      Up      Up      null   service    1500  00:50:56:00:01:05  1000   full   0       0:07:45:55  15053    10333
```

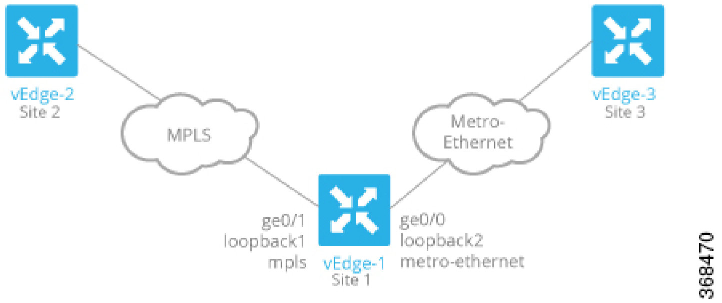
To allow vEdge routers at different overlay network sites on the private network to exchange data traffic directly, you use a loopback interface on the each vEdge router to handle the data traffic instead of using the physical interface that connects to the WAN. You associate the same tag, called a carrier tag, with each loopback interface so that all the routers learn that they are on the same private WAN. Because the loopback interfaces are advertised across the overlay network, the vEdge routers are able to learn reachability information, and they can exchange data traffic over the private network. To allow the data traffic to actually be transmitted out the WAN interface, you bind the loopback interface to a physical WAN interface, specifically to the interface that connects to the private network. Remember that this is the interface that the private network does not advertise. However, it is still capable of transmitting data traffic.

### Share a Common Service across Different VPNs

When services such as firewalls or load balances are spread across multiple VPNs, you can create a policy that forces traffic from one VPN to use the services in another VPN. See the service control examples in Service Chaining Configuration Examples.

# Use Case: Exchange Data Traffic between Two Private WANs

A variant of the topology illustrated above is the case in which a single vEdge router connects to two different private WANs, such as two different MPLS clouds provided by two different network carriers, or two different types of private WANs, as illustrated below. In this figure, the vEdge-1 router connects to one MPLS private WAN and one metro-Ethernet private WAN.



As in the previous example, you create loopback interfaces on the three routers. For vEdge-1, which connects to both of the private WANs, you create two loopback interfaces. For each one, you assign a color, as in the previous example. But you configure two more things: you assign a tag to identify the carrier, and you "bind" the loopback interface to the physical interface that connects to the private WAN. So, vEdge-1 has two loopback interfaces with these properties:

- Loopback1 has the color **mpls**, the carrier **carrier2**, and binds to physical interface ge0/1.
- Loopback 2 has the color **metro-ethernet** and the carrier **carrier1**, and binds to physical interface ge0/0.

The vEdge-2 router has a single loopback interface that connects to the MPLS private WAN. Its color is **mpls**, and its carrier is **carrier2**. Both these properties match those on the loopback1 interface on vEdge-1. However, because vEdge-2 connects to only one private WAN, there is no need to bind its loopback interface to a physical interface.

Finally, vEdge-3 has a single loopback interface with color **metro-ethernet** and carrier **carrier1**, matching the properties configured on the vEdge-1 loopback2 interface.

On vEdge-1, the configuration in VPN 0 looks like this:

```
vpn 0
 interface ge0/0
  ip address 10.1.15.15/24
  no shutdown
 !
 interface loopback2
  ip address 172.16.15.15/24
  tunnel-interface
   color   metro-ethernet
   carrier carrier1
   bind    ge0/0
  !
  no shutdown
 !

 interface ge0/1
  ip address 10.1.17.15/24
  no shutdown
 !
 interface loopback1
  ip address 172.16.17.15/24
  tunnel-interface
   color   mpls
   carrier carrier2
   bind    ge0/1
  !
  no shutdown
!
```

If you need to apply control policy to a particular private network, use the **match carrier** option when creating the control policy.

# Segmentation CLI Reference

CLI commands for monitoring segmentation (VPNs) .

- **show bgp** commands
- **show interface** commands
- **show ospf** commands

CHAPTER 6

# Forwarding and QoS

Forwarding is the transmitting of data packets from one router to another.

Quality of Service (QoS) is synonymous with class of service (CoS). You can enable QoS with localized data policies, which control the flow of data traffic into and out of the interfaces of Cisco vEdge devices and Cisco XE SD-WAN devices.

## Cisco SD-WAN Forwarding and QoS Overview

Forwarding takes the data packet and sends it over the transport to the remote side, specifying what to do with the packet. It specifies the interface through which packets are sent to reach the service side of a remote router.

Once the control plane connections of the Cisco SD-WAN overlay network are up and running, data traffic flows automatically over the IPsec connections between the routers. Because data traffic never goes to or through the centralized vSmart controller, forwarding only occurs between the Cisco vEdge devices as they send and receive data traffic.

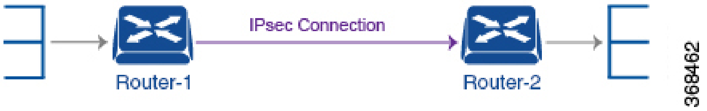
While the routing protocols running in the control plane provide a router the best route to reach the network that is on the service side of a remote router, there will be situations where it is beneficial to select more specific routes. Using forwarding, there are ways you can affect the flow of data traffic. Forwarding takes the data packet and sends it over the transport to the remote side, specifying what to do with the packet. It specifies the interface through which packets are sent to reach the service side of a remote router.

To modify the default data packet forwarding flow, you create and apply a centralized data policy or a localized data policy. With a centralized data policy, you can manage the paths along which traffic is routed through the network, and you can permit or block traffic based on the address, port, and DSCP fields in the packet's IP header. With a localized data policy, you can control the flow of data traffic into and out of the interfaces of a router, enabling features such as quality of service (QoS) and mirroring.

# Traffic Behavior With and Without QoS

### Default Behavior without Data Policy

When no centralized data policy is configured on the vSmart controller, all data traffic is transmitted from the local service-side network to the local router, and then to the remote router and the remote service-side network, with no alterations in its path. When no access lists are configured on the local router to implement QoS or mirroring, the data traffic is transmitted to its destination with no alterations to its flow properties.

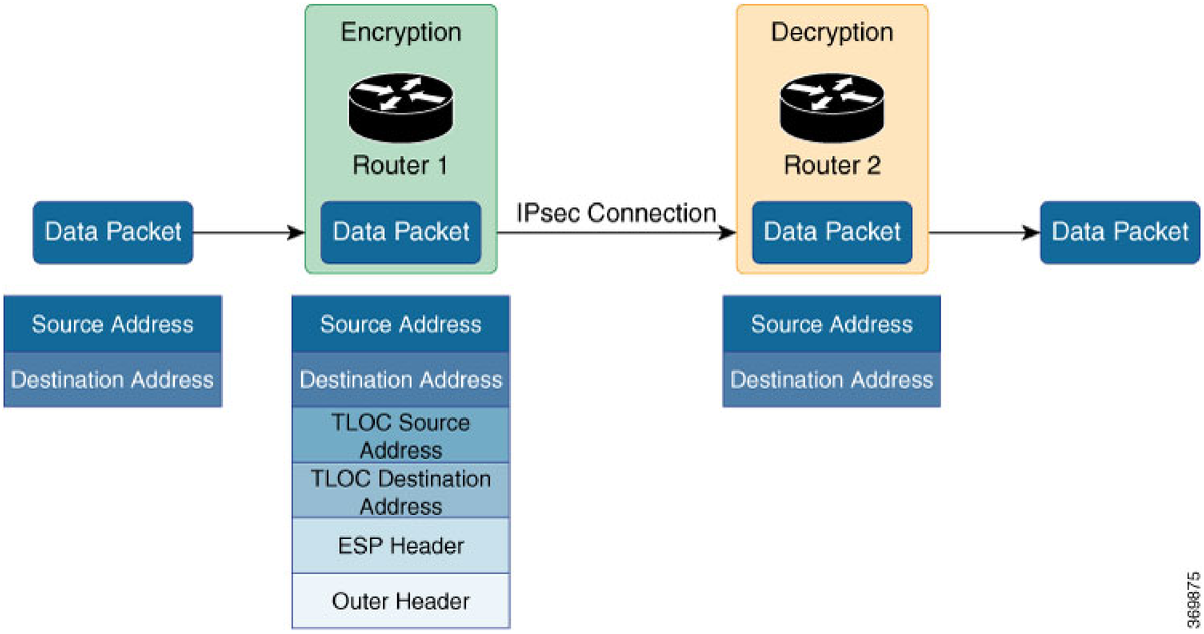


Let's follow the process that occurs when a data packet is transmitted from one site to another when no data policy of any type is configured:

- A data packet arriving from the local service-side network and destined for the remote service-side network comes to the router-1. The packet has a source IP address and a destination IP address.
- The router looks up the outbound SA in its VPN route table, and the packet is encrypted with SA and gets the local TLOC. (The router previously received its SA from the vSmart controller. There is one SA per TLOC. More specifically, each TLOC has two SAs, an outbound SA for encryption and an inbound SA for decryption.)
- ESP adds an IPsec tunnel header to the packet.
- An outer header is added to the packet. At this point, the packet header has these contents: TLOC source address, TLOC destination address, ESP header, destination IP address, and source IP address.
- The router checks the local route table to determine which interface the packet should use to reach its destination.
- The data packet is sent out on the specified interface, onto the network, to its destination. At this point, the packet is being transported within an IPsec connection.
- When the packet is received by the router on the remote service-side network, the TLOC source address and TLOC destination address header fields are removed, and the inbound SA is used to decrypt the packet.
- The remote router looks up the destination IP address in its VPN route table to determine the interface to use to reach to the service-side destination.

The figure below details this process.

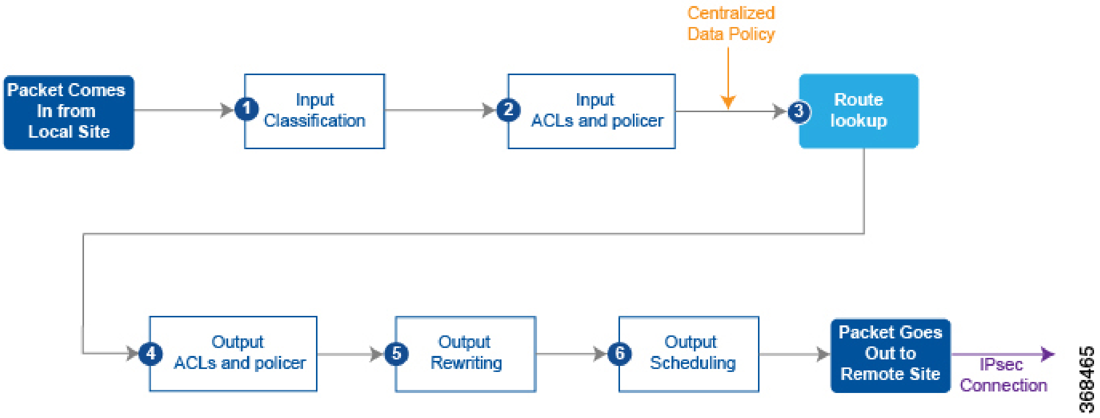
*Figure 1: Data Packet Transmission without Policy*



### Behavior Changes with QoS Data Policy

When you want to modify the default packet forwarding flow, you design and provision QoS policy. To activate the policy, you apply it to specific interfaces in the overlay network in either the inbound or the outbound direction. The direction is with respect to the routers in the network. You can have policies for packets coming in on an interface or for packets going out of an interface.

The figure below illustrates the QoS policies that you can apply to a data packet as it is transmitted from one branch to another. The policies marked Input are applied on the inbound interface of the router, and the policies marked Output are applied on the outbound interface of the router, before the packets are transmitted out the IPSec tunnel.



The table below describes each of the above steps.

| Step | Description | Command |
|---|---|---|
| 1 | Define class map to classify packets, by importance, into appropriate forwarding classes. Reference the class map in an access list. | **class-map** |
| 2 | Define policer to specify the rate at which traffic is sent on the interface. Reference the policer in an access list. Apply the access list on an inbound interface. | **policer** |
| 3 | The router checks the local route table to determine which interface the packet should use to reach its destination. | N/A |
| 4 | Define policer and reference the policer in an access list. Apply the access list on an outbound interface. | **policer** |
| 5 | Define QoS map to define the priority of data packets. Apply the QoS map on the outbound interface. | **qos-map** |
| 6 | Define rewrite-rule to overwrite the DSCP field of the outer IP header. Apply the rewrite-rule on the outbound interface. | **rewrite-rule** |

# How QoS Works

The QoS feature on the  Cisco XE SD-WAN devices and Cisco vEdge devices works by examining packets entering at the edge of the network. With localized data policy, also called access lists, you can provision QoS to classify incoming data packets into multiple forwarding classes based on importance, spread the classes across different interface queues, and schedule the transmission rate level for each queue. Access lists can be applied either in the outbound direction on the interface (as the data packet travels from the local service-side network into the IPsec tunnel toward the remote service-side network) or in the inbound direction (as data packets are exiting from the IPsec tunnel and being received by the local router.

To provision QoS, you must configure each router in the network. Generally, each router on the local service-side network examines the QoS settings of the packets that enter it, determines which class of packets are transmitted first, and processes the transmission based on those settings. As packets leave the network on the remote service-side network, you can rewrite the QoS bits of the packets before transmitting them to meet the policies of the targeted peer router.

### Classify Data Packets

You can classify incoming traffic by associating each packet with a forwarding class. Forwarding classes group data packets for transmission to their destination. Based on the forwarding class, you assign packets to output queues. The routers service the output queues according to the associated forwarding, scheduling, and rewriting policies you configure.

### Schedule Data Packets

You can configure a QoS map for each output queue to specify the bandwidth, delay buffer size, and packet loss priority (PLP) of output queues. This enables you to determine how to prioritize data packets for transmission to the destination. Depending on the priority of the traffic, you can assign packets higher or lower bandwidth, buffer levels, and drop profiles. Based on the conditions defined in the QoS map, packets are forwarded to the next hop.

On Cisco vEdge devices and Cisco XE SD-WAN devices, each interface has eight queues, which are numbered 0 to 7. Queue 0 is reserved, and is used for both control traffic and low-latency queuing (LLQ) traffic. For LLQ, any class that is mapped to queue 0 must also be configured to use LLQ. Queues 1 to 7 are available for data traffic, and the default scheduling for these seven queues is weighted round-robin (WRR). For these queues, you can define the weighting according to the needs of your network. When QoS is not configured for data traffic, queue 2 is the default queue.

### Rewrite Data Packets

You can configure and apply rewrite rules on the egress interface to overwrite the Differentiated Services Code Point (DSCP) value for packets entering the network. Rewrite rules allow you to map traffic to code points when the traffic exits the system. Rewrite rules use the forwarding class information and packet loss priority (PLP) used internally by the Cisco XE SD-WAN devices and Cisco vEdge devices to establish the DSCP value on outbound packets. You can then configure algorithms such as RED/WRED to set the probability that packets will be dropped based on their DSCP value.

### Police Data Packets

You can configure policers to control the maximum rate of traffic sent or received on an interface, and to partition a network into multiple priority levels.

Traffic that conforms to the policer rate is transmitted, and traffic that exceeds the policer rate is sent with a decreased priority or is dropped.

You can apply a policer to inbound or outbound interface traffic. Policers applied to inbound interface traffic allow you to conserve resources by dropping traffic that does not need to be routed through the network. Policers applied to outbound interface traffic control the amount of bandwidth used.

### Shaping Rate

You can configure shaping to control the maximum rate of traffic sent. You can configure the aggregate traffic rate on an interface to be less than the line rate so that the interface transmits less traffic than it is capable of transmitting. You can apply shaping to outbound interface traffic.

**Note** Shaping rate below 2M is not supported on the following Cisco vEdge devices: Cisco vEdge100b, Cisco vEdge100m, Cisco vEdge 1000, and Cisco vEdge 2000.

# QoS vManage

Any type of change in configuration will cause the QoS policy to be removed and added to an interface. As a result, there will be a sharp fall to 0 in the QoS monitor chart. The statistics depicted on the QoS monitoring chart for the configuration change time interval can be disregarded.

# Forwarding and QoS Configuration Examples

This section shows examples of how you can use access lists to configure quality of service (QoS), classifying data packets and prioritizing the transmission properties for different classes. Note that QoS is synonymous with class of service (CoS).

This example shows how to configure class of service (CoS) to classify data packets and control how traffic flows out of and into the interfaces on Cisco vEdge devices on the interface queues. To configure a QoS policy:

1. Map each forwarding class to an output queue.
2. Configure the QoS scheduler for each forwarding class.
3. Group the QoS schedulers into a QoS map.
4. Define an access list to specify match conditions for packet transmission and apply it to a specific interface.
5. Apply the queue map and the rewrite rule to the egress interface.

The sections below show examples of each of these steps.

# Map Each Forwarding Class to Output Queue

This example shows a data policy that classifies incoming traffic by mapping each forwarding class to an output queue. Here, traffic classified as "be" (Best Effort) is mapped to queue 2, traffic classified as "af1" (Assured Forwarding) is mapped to queue 3, and so on.

```
policy
 class-map
  class be queue 2
  class af1 queue 3
  class af2 queue 4
  class af3 queue 5
 !
!
```

# Configure QoS Scheduler for Each Forwarding Class

This example illustrates how to configure the QoS scheduler for each queue to define the importance of data packets.

Depending on the priority of the traffic, you assign the bandwidth, buffer level, and random early detection (RED) drop profile associated with the queue. Here, "af3" traffic has higher priority over other traffic classes and so is configured to have 40% bandwidth and 40% buffer. Traffic in class "af2" has 30% bandwidth and 30% buffer; traffic in class "af1" class has 20% bandwidth and 20% buffer and traffic in class "be" has 10% bandwidth and 10% buffer size reflecting the respective priority of the traffic on the network. All traffic classes are configured with a drop profile of RED, meaning that instead of waiting for the queue to be full, packets are dropped randomly based on the thresholds defined.

```
policy
 qos-scheduler af1
  class             af1
  bandwidth-percent 20
  buffer-percent    20
  drops             red-drop
 !
 qos-scheduler af2
  class             af2
  bandwidth-percent 30
  buffer-percent    30
  drops             red-drop
 !
 qos-scheduler af3
```

```
  class             af3
  bandwidth-percent 40
  buffer-percent    40
  drops             red-drop
 !
 qos-scheduler be
  class             be
  bandwidth-percent 10
  buffer-percent    10
  drops             red-drop
 !
```

# Group QoS Schedulers into a QoS Map

This example illustrates the grouping of "qos scheduler af1," "qos scheduler af2," and "qos scheduler be" into a single QoS map called "test."

```
qos-map test
  qos-scheduler af1
  qos-scheduler af2
  qos-scheduler be
 !
!
```

**Note** The sum of bandwidth-percent for qos-scheduler configured under the QoS map should not exceed 100.

The sum of buffer-percent for qos-scheduler configured under the QoS map should not exceed 100.

# Create Access Lists to Classify Data Packets

### Classify Data Packets into Appropriate Classes

This example shows how to classify data packets into appropriate forwarding classes based on match conditions. Here "access-list acl1" classifies data packets originating from the host at source address 10.10.10.1 and going to the destination host at 20.20.20.1 into the "be" class. Data packets with a DSCP value of 10 in the IP header field are classified in the "af1" class, TCP packets are classified in the "af3" class, and packets going to destination port 23, which carries Telnet mail traffic, are classified in the "af2" class. All other traffic is dropped.

```
policy
 access-list acl1
  sequence 1
   match
    source-ip      10.10.10.1/32
    destination-ip 10.20.20.1/32
   !
   action accept
    class be
   !
  !
  sequence 2
   match
    dscp 10
   !
   action accept
    class af1
   !
  !
  sequence 3
   match
    protocol 6
   !
   action accept
    class af3
   !
  !
  sequence 4
   match
    destination-port 23
   !
   action accept
    class af2
   !
  !
  default-action drop
 !
!
```

# Apply Access Lists

### Apply Access List to Specific Interface

This example illustrates how to apply the access list defined above on the input of a service interface. Here "access-list acl1" is applied on the input of interface ge0/4 in VPN 1.

```
vpn 1
 interface ge0/4
  ip address 10.20.24.15/24
  no shutdown
  access-list acl1 in
 !
!
```

# Configure and Apply Rewrite Rule

### Configure Rewrite Rule

This example shows how to configure the rewrite rule to overwrite the DSCP field of the outer IP header. Here the rewrite rule "transport" overwrites the DSCP value for forwarding classes based on the drop profile. Since all classes are configured with RED drop, they can have one of two profiles: high drop or low drop. The rewrite rule is applied only on the egress interface, so on the way out, packets classified as "af1" and a Packet Loss Priority (PLP) level of low are marked with a DSCP value of 3 in the IP header field, while "af1" packets with a PLP level of high are marked with 4. Similarly, "af2" packets with a PLP level of low are marked with a DSCP value of 5, while "af2" packets with a PLP level of high are marked with 6, and so on.

```
policy
 rewrite-rule transport
  class af1 low dscp 3
  class af1 high dscp 4
  class af2 low dscp 5
  class af2 high dscp 6
  class af3 low dscp 7
  class af3 high dscp 8
  class be low dscp 1
  class be high dscp 2
 !
!
```

### Apply the Queue Map and Rewrite Rule on an Interface

This example applies the queue map "test" and the rewrite rule "transport" to the egress interface ge0/0 in VPN 0. (Note that you can apply QOS maps to VLAN interfaces, also called subinterfaces from Cisco IOS XE SD-WAN Release 16.12.x and Cisco SD-WAN Release 19.1.x and later. Queue maps and rewrite rules are applied only on outgoing traffic.

```
vpn 0
 interface ge0/0
  ip address 10.1.15.15/24
  tunnel-interface
   preference 10
   weight     10
   color      lte
   allow-service dhcp
   allow-service dns
   allow-service icmp
   no allow-service sshd
   no allow-service ntp
   no allow-service stun
  !
  no shutdown
  qos-map  test
  rewrite-rule transport
 !
!
```

# Police Data Packets on Cisco vEdge Devices

This section shows two examples of policing data packets.

The first example illustrates how to configure a policer to rate limit traffic received on an interface. After you configure the policer, include it in an access list. Here "policer p1" is configured to have a maximum traffic rate of 1,000,000 bits per second and a maximum burst-size limit of 15000 bytes. Traffic exceeding these rate limits is dropped. The policer is then included in the access list "acl1," which is configured to accept all TCP or UDP traffic originating from the host at source 2.2.0.0 and going to the destination host at 10.1.1.0 on port 20 or 100.1.1.0 on port 30. You can use "access-list acl1" on the input or output of the interface to do flow-based policing.

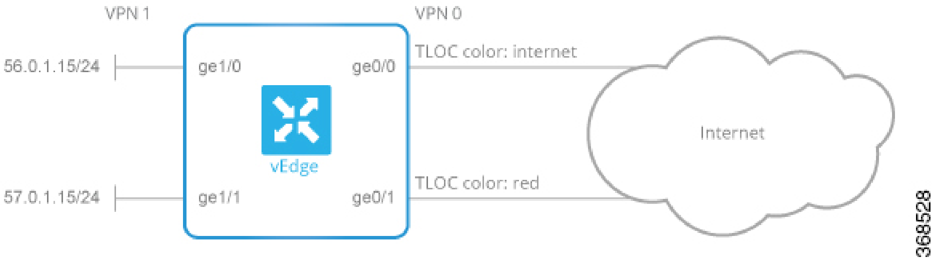
```
policy
 policer p1
  rate   1000000
  burst  15000
  exceed drop
 !
 access-list acl1
  sequence 1
   match
    source-ip        2.2.0.0/16
    destination-ip   10.1.1.0/24 100.1.1.0/24
    destination-port 20 30
    protocol         6 17 23
   !
   action accept
    policer p1
   !
  !
  default-action drop
 !
!
vpn 1
 interface ge0/4
  ip address 10.20.24.15/24
  no shutdown
  access-list acl1 in
 !
!
```

You can also apply a policer directly on an inbound or an outbound interface when you want to police all traffic ingressing or egressing this interface:

```
policy
 policer p1
  rate   1000000
  burst  15000
  exceed drop
 !
!
vpn 1
 interface ge0/4
  ip address 10.20.24.15/24
  no shutdown
  policer p1 in
 !
!

vpn 2
 interface ge0/0
  ip address 10.1.15.15/24
  no shutdown
  policer p1 out
 !
!
```

In the second example, we have a Cisco vEdge device with two WAN interfaces in VPN 0. The ge0/0 interface connects to a 30-MB link, and we want to always have 10 MB available for very high priority traffic. When lower-priority traffic bursts exceed 20 MB, we want to redirect that traffic to the second WAN interface, ge0/1.



Implementing this traffic redirection requires two policies:

- You apply an access list to the service-side interface that polices the incoming data traffic.
- You apply a data policy to the ge0/0 WAN interface that directs bursty traffic to the second WAN interface, ge0/1.

For the access list, the configuration snippet below is for interface ge1/0, in VPN 1. The policer monitors incoming traffic on the interface. When traffic exceeds 20 MB (configured in the **policer burst** command), we change the PLP from low to high (configured by the **policer exceed remark** command). You configure the following on the Cisco vEdge device:

```
policy
  policer bursty-traffic
    rate 1000000
    burst 20000
    exceed remark
  access-list policer-bursty-traffic
    sequence 10
      match
        source-ip 56.0.1.0/24
      action accept
        policer bursty-traffic
    default-action accept
vpn 1
  interface ge1/0
    ip address 56.0.1.14/24
    no shutdown
    access-list policer-bursty-traffic in
```

To display a count of the packets that have been remarked, issue the **show interface detail** or the **show system statistics** command on the Cisco vEdge device. The count is reported in the rx-policer-remark field.

The centralized data policy directs burst traffic away from the ge0/0 interface (color: internet) to interface ge0/1 (color: red). You apply this data policy to all the routers at a particular site, specifying the direction **from-service** so that the policy is applied only to traffic originating from the service side of the router. You configure the following on the vSmart controller:

```
policy
  lists
    site-list highest-priority-routers
      site-id 100
    vpn-list wan-vpn
      vpn 0
  data-policy highest-priority
    vpn-list wan-vpn
      sequence 10
        match
          plp high
          source-ip 56.0.1.0/24
        action accept
          count bursty-counter
          set local-tloc color red
    default-action accept
apply-policy
  site-list highest-priority-routers
    data-policy highest-priority from-service
```

# Reference: Forwarding and QoS CLI Commands

### Monitoring Commands

Use the following commands to monitor forwarding and QoS on a Cisco vEdge device:

```
show policy access-list-associations
show policy access-list-counters
show policy access-list-names
show policy access-list-policers
show policy data-policy-filter
show policy qos-map-info
show policy qos-scheduler-info
```

### Monitoring Commands

Use the following commands to monitor forwarding and QoS on a Cisco XE SD-WAN device:

```
show sdwan policy access-list-associations
show sdwan policy access-list-counters
show sdwan policy access-list-names
show sdwan policy access-list-policers
show sdwan policy data-policy-filter
show sdwan policy rewrite-associations
show policy-map interface GigabitEthernet0/0/2
```