
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 1
- What's New for Cisco IOS XE SD-WAN Release 16.12.x 1
- What's New for Cisco SD-WAN Release 19.2.x 3

CHAPTER 2
Bridging 5
- Bridging Overview 5
- Components of Bridging 5
- Configure Bridging 8
  - Configure Bridging Using vManage Templates 8
  - Configure Bridging and Bridge Domains Using CLI 12
- Configure IRB 14
- Bridging CLI Reference 15

CHAPTER 3
Unicast Overlay Routing 17
- Design Overlay Network Using vManage 17
- Supported Protocols 30
  - OMP Routing Protocol 30
    - OMP Route Advertisements 31
    - OMP Route Redistribution 34
    - OMP Graceful Restart 37
  - BGP and OSPF Routing Protocols 37
- Configure Unicast Overlay Routing 37
  - Configure BGP Using vManage Templates 38
  - Configure BGP Using CLI 44
  - Configure OSPF Using vManage Templates 48
  - Configure OSPF Using CLI 56
What's New for Cisco SD-WAN

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

- What's New for Cisco IOS XE SD-WAN Release 16.12.x, on page 1
- What's New for Cisco SD-WAN Release 19.2.x, on page 3

What's New for Cisco IOS XE SD-WAN Release 16.12.x

This section applies to Cisco XE SD-WAN routers.

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 XE SD-WAN Router

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systems and Interfaces</td>
<td>Support for Implementation of IEEE CFM was introduced in Cisco SD-WAN 19.1. IEEE CFM or Ethernet CFM is an end-to-end per-service Ethernet layer OAM protocol. CFM includes proactive connectivity monitoring, fault verification, and fault isolation for large Ethernet metropolitan-area networks (MANs) and WANs. Y.1731 is an ITU-T recommendation for OAM functions in Ethernet-based networks. The implementation of IEEE 802.1ag Standard-Compliant CFM and Y.1731 are now supported in Cisco IOS XE software.</td>
</tr>
<tr>
<td>NAT64-DIA (Direct Internet Access)</td>
<td>You can now configure RSA keys to secure communication between a client and a Cisco SD-WAN server. For related information, see SSH Authentication using vManage on Cisco XE SD-WAN Devices.</td>
</tr>
<tr>
<td>DHCP option support</td>
<td>This release extends support for DCHP options 43 and 191 to Cisco XE-SD-WAN routers. For related information, see Configure DHCP.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>UCS-E module Support</td>
<td>This feature adds a UCS-E template in vManage for configuring Cisco Unified Computing System (UCS) E-Series servers. For related information, see the Getting Started Guide.</td>
</tr>
</tbody>
</table>

**Bridging, Routing, Segmentation, and QoS**

| Subinterface QoS | A physical interface may be treated as multiple interfaces by configuring one or more logical interfaces called subinterfaces. This feature enables Quality of Service (QoS) policies to be applied to individual subinterfaces. For related information, see Connectivity and QoS Configuration Guide. |

**Policies**

| CoS marking (802.1P) | This feature allows you to mark packets using Class of Service (CoS) values. A low priority value was added for Cisco XE SD-WAN routers. For related information, see Localized Data Policy for IPv4. |
| Support for packet duplication for loss correction | This feature helps mitigate packet loss over noisy channels, thereby maintaining high application QoE for voice and video in particular. This feature is supported on Cisco XE SD-WAN routers as well as on Cisco vEdge routers. For related information, see Configure and Monitor Packet Duplication. |
| Integration with Cisco ACI | The SD-WAN and Cisco ACI integration functionality now supports predefined SLA cloud beds. It also supports dynamically generated mappings from a data prefix list and includes a VPN list to an SLA class that is provided by Cisco ACI. For related information, see Integration with Cisco ACI. |
| Encryption of Lawful Intercept Messages | Lawful intercept messages between a Cisco XE SD-WAN router and a Media Device can now be encrypted using static tunnel information. For related information, see Lawful Intercept. |

**Security**

| High-Speed Logging for Zone-Based Firewalls | High-Speed Logging (HSL) allows a firewall to log records with minimum impact to packet processing. For related information, see Firewall High-Speed Logging. |
| Self zone policy for Zone-Based Firewalls | Self-zone is an default zone in the firewall that is associated with the VPN for punt and inject interface. You can define policies to impose rules on the incoming and outgoing traffic. For related information, see Configure Firewall Policies Using vManage. |
| Secure Communication Using Pairwise IPsec Keys | This feature enables support to create and install private pairwise IPsec session keys to secure communication between IPsec devices and its peers. For related information, see IPsec Pairwise Keys Overview. |
This release adds support on the Cisco 1101 Series Integrated Services Router for the same SD-wan features that the Cisco 1100 ISR Router supports.

You can now configure a loopback transport interface on a Cisco IOS XE SD-WAN router to help in troubleshooting and diagnostics. For related information, see bind.

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

This section applies to Cisco vEdge routers.

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>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco 1101 Series Integrated Services Router</td>
<td>This release adds support on the Cisco 1101 Series Integrated Services Router for the same SD-wan features that the Cisco 1100 ISR Router supports.</td>
</tr>
</tbody>
</table>

### Commands

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loopback interface support for WAN (IPsec)</td>
<td>You can now configure a loopback transport interface on a Cisco IOS XE SD-WAN router to help in troubleshooting and diagnostics. For related information, see bind.</td>
</tr>
</tbody>
</table>

---

**Table 2: What's New for Cisco vEdge Router**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Systems and Interfaces</strong></td>
<td></td>
</tr>
<tr>
<td>Secure Shell Authentication Using RSA Keys</td>
<td>This feature enables secure shell authentication between a client and a Cisco SD-WAN server using RSA keys. For related information, see SSH Authentication using vManage on Cisco XE SD-WAN Devices.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Policy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support for packet duplication for loss correction</td>
<td>This feature provides support for packet duplication by overcoming packet loss. Packet duplication is supported on Cisco XE SD-WAN routers as well as on Cisco vEdge routers. For related information, see Configure and Monitor Packet Duplication.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Security</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPSec Pairwise Keys</td>
<td>This feature enables support to create and install private pairwise IPSec session keys to secure communication between IPSec devices and its peers. For related information, see IPSec Pairwise Keys Overview.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network Optimization and High Availability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support for disaster recovery</td>
<td>This feature provides support for disaster recovery by deploying vManage in primary/secondary mode. For detailed information, see Configure Disaster Recovery.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Share VNF Devices Across Service Chains</td>
<td>You now have the flexibility to share Virtual Network Function (VNF) devices across service chains to improve the resource utilization and reduce resource fragmentation. For related information, see Manage PNF Devices in Service Chains.</td>
</tr>
<tr>
<td>Monitor Service Chain Health</td>
<td>The service chain health is now monitored by running periodic checks on the service chain data path and reporting the overall status. Note: For service chain health monitoring to be enabled on all CSP devices in a cluster, ensure that you install the NFVIS version 3.12.1 or later. For related information, see Manage PNF Devices in Service Chains.</td>
</tr>
<tr>
<td>Manage PNF Devices in Service Chains</td>
<td>A service chain can now have a mix of Virtual Network function (VNF) and Physical Network Function (PNF) devices. You can add PNF devices to service chains and share them across service chains, service groups, and a single cluster. The PNF devices can overcome the performance and scaling issues caused by using only VNF devices in a service chain. Note: Due to higher cost and maintenance of PNF devices, it is recommended to have a single PNF device that is shared across multiple service chains. For related information, see Manage PNF Devices in Service Chains.</td>
</tr>
</tbody>
</table>
CHAPTER 2

Bridging

This chapter contains these topics:

- Bridging Overview, on page 5
- Components of Bridging, on page 5
- Configure Bridging, on page 8
- Bridging CLI Reference, on page 15

Bridging Overview

A vEdge router can act as a transparent bridge, switching traffic between LANs that are part of a VLAN at the local router’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 vEdge router.

To allow hosts in different bridge domains to communicate with each other, vEdge routers 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 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 vEdge routers. Specifically, they are the base interfaces, for example, **ge-0/0/0**. 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 vEdge router. 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 architecture emulates these functions. For a vEdge router 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 vEdge router 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.

Native 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 a 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 communication 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 vEdge router 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

### Configure Bridging Using vManage Templates

Use the Switch Port template to configure bridging domains Cisco IOS XE routers.

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

To configure the switch ports using vManage templates:

1. Create a Switch Port feature template, as described in this article.
2. To use the switch port for routing, associate it with an SVI. See the VPN Interface SVI help topic.

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

1. In vManage NMS, select the Configuration ► Templates screen.
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 Additional Templates tab located directly beneath the Description field, or scroll to the Additional Templates section.
6. Click the plus sign (+) next to Switch Port.
7. In the Switch Port drop-down, select the port number.
8. From the lower Switch Port drop-down, click Create Template. The Switch Port template form is displayed. The top of the form contains fields for naming the template, and the bottom contains fields for defining switch port 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:
Configure Bridging Using vManage Templates

Configure Basic Switch Port Parameters

To configure basic switch port parameters, select the Basic Configuration tab and configure the following parameters:

Table 4:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slot</td>
<td>Enter the number of the slot in which the Layer 2 switch port module is installed.</td>
</tr>
<tr>
<td>Module</td>
<td>Select the switch port module type, either 4 port or 8 port.</td>
</tr>
</tbody>
</table>

To save the feature template, click Save.

Associate Interfaces with the Switch Port

To associate an interface with the switch port, click the Interface tab and click Add New Interface.

The Wlan-GigabitEthernet0/1/8 interface applies only to C1111-8PW and C1111-8PLTExxW routers. When you configure this interface, select either C1111-8PW or C1111-8PLTExxW when you create a switch port, and select 8 port from the Module drop-down list. In addition, from the New Interface drop-down menu, make sure to select Wlan-GigabitEthernet0/1/8.
Table 5:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Name</td>
<td>Enter the name of the interface to associate with the bridging domain, in the format <strong>ge slot/port</strong>.</td>
</tr>
<tr>
<td>Shutdown</td>
<td>Click No to enable the interface. By default, an interface is disabled.</td>
</tr>
</tbody>
</table>
| Switch Port    | Select the switch port mode:  
  - **Access**—Configure the interface as an access port. You can configure only one VLAN on an access port, and the port can carry traffic for only one VLAN.  
    - **VLAN Name**—Enter a description for the VLAN.  
    - **VLAN ID**—Enter the VLAN number, which can be a value from 1 through 4094.  
  - **Trunk**—Configure the interface as a trunk port. You can configure one or more VLANs on a trunk port, and the port can carry traffic for multiple VLAN.  
    - **Allowed VLANs**—Enter the numbers of the VLANs for which the trunk can carry traffic. 
    - **Native VLAN ID**—Enter the number of the VLAN allowed to carry untagged traffic. |

To save the feature template, click Save.

Configure Other Interface Properties

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

Table 6:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age-Out Time</td>
<td>Enter how long an entry is in the MAC table before it ages out. Set the value to 0 to prevent entries from timing out. <strong>Range:</strong> 0, 10 through 1000000 seconds <strong>Default:</strong> 300 seconds</td>
</tr>
</tbody>
</table>
| Static MAC Address      | Click Add Static MAC Address to map a MAC address to a switch port. In the MAC Static Address field that appears, enter the following:  
  - **MAC Address**—Enter the static MAC address to map to the switch port interface.  
  - **Switch Port Interface Name**—Enter the name of the switch port interface.  
  - **VLAN ID**—Enter the number of the VLAN for the switch port.  
  Click Add to save the static MAC address mapping. |

To save the feature template, click Save.
Release Information

Introduced in vManage NMS in Release 18.3.

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 vEdge router 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 vEdge router 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-bridge)# 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-bridge)# 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)# 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
```

<table>
<thead>
<tr>
<th>BRIDGE</th>
<th>INTERFACE</th>
<th>VLAN</th>
<th>STATUS</th>
<th>TYPE</th>
<th>IFINDEX</th>
<th>MTU</th>
<th>RX PKTS</th>
<th>OCTETS</th>
<th>TX PKTS</th>
<th>OCTETS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>ge0/4</td>
<td>27</td>
<td>Up</td>
<td>vlan</td>
<td>41</td>
<td>1500</td>
<td>4 364</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

"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
```

---

**Bridging CLI Reference**

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

- Design Overlay Network Using vManage, on page 17
- Supported Protocols, on page 30
- Configure Unicast Overlay Routing, on page 37
- Use Case: Unicast Routing in Viptela Overlay Network, on page 65
- Unicast Overlay Routing CLI Reference, on page 68

Design Overlay Network Using vManage

Use the Network Design screen to create and manage an overlay network topology. From this screen, you can add circuits, data centers, and branch sites to a network topology, configure LAN, WAN, and management options for elements in the topology, review the topology, and perform related tasks. The network design operations are particularly useful for smaller-scale deployments that include data centers and branch sites.

Network design consists of these major workflows:

- Create network topology—Create circuits, data centers, and branch sites, in this order. A network topology must include at least one circuit and one data center.
- Configure device profiles—Configure global parameters and options for LAN, WAN, and management settings.
- Attach devices profiles—Attach device profiles to devices.
- Ongoing management—Add elements to the network topology and modify the configuration settings for elements as needed.

Access Network Design Options

To access options for creating or updating a network design, select Configuration ➤ Network Design.

The Network Design screen displays. This screen includes the following items:
• **Create Network Design** button—Displays if you have not yet created a network topology. Click to create elements for the network. For more information, see Configure Network Design Elements.

• **Manage Network Design** button—Displays if you have created a network topology. Click to modify configuration setting for elements in the network. For more information, see Configure Network Design Elements.

• **Attach Devices** button—Click to access options for attaching a device profile to a device, detach a device profile from a device, export device profile configuration values to a CSV file, or modify values in a device profile. For more information, see Attach, Detach, Export, Update Device Profiles.

• Last modified information—Date and time that the network design was last modified.

• **Device Attached Task** option—Displays if the system is in the process of attaching a device profile to devices or updating device profile configuration information. For more information, see Attach Device Profile or Change Device Profile Values.

• **Network design topology diagram**—Displays if you have created a network topology. Figure 1 shows an example diagram.

Figure 1. Network Design Topology Display
Unicast Overlay Routing

Design Overlay Network Using vManage

1. DataCenter_East
2. DataCenter_West
3. 1 Segment
4. 1 Segment
5. 2 Segments
6. 1 Segment
7. 1 Segment
8. 1 Segment
9. 3g (private)
10. lte (private)
   red (private)
   gold (private)
Table 7:

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Custom device profile for a device in a data center. Custom profiles are indicated by a solid border and the icon at the top left corner. If the partial name of a device profile displays, hover your mouse pointer over the name to see the full name. If a device profile is attached to 1 or more devices, the following icons and information display:</td>
</tr>
<tr>
<td></td>
<td>• <img src="image1" alt="icon" /> 1</td>
</tr>
<tr>
<td></td>
<td>— Indicates the number of devices that the profile is successfully attached to.</td>
</tr>
<tr>
<td></td>
<td>• <img src="image2" alt="icon" /> 0</td>
</tr>
<tr>
<td></td>
<td>— Indicates the number of devices that the profile failed to attach to. If there are failed attachments, the device is out of sync.</td>
</tr>
<tr>
<td></td>
<td>• <img src="image3" alt="icon" /> 0</td>
</tr>
<tr>
<td></td>
<td>— Indicates the number of devices that the profile is in the process of attaching to.</td>
</tr>
<tr>
<td>2</td>
<td>Name of a data center. If the partial name of a data center displays, hover your mouse pointer over the name to see the full name.</td>
</tr>
<tr>
<td>3</td>
<td>Standard device profile for a device in a data center. Standard profiles are indicated by a dashed border. If the partial name of a device profile displays, hover your mouse pointer over the name to see the full name. If a device profile is attached to 1 or more devices, icons and information display as described in Row 1 of this table.</td>
</tr>
<tr>
<td>4</td>
<td>Number of segments that are assigned to a data center or branch site. Hover your mouse pointer over the segment display to see the name of each segment.</td>
</tr>
<tr>
<td>5</td>
<td>TLOC connections between elements in the topology. A custom device profile does not display TLOC connections to other elements because its settings, such as LAN, WAN, and circuit configurations, have been converted to feature templates.</td>
</tr>
<tr>
<td>6</td>
<td>Circuit.</td>
</tr>
<tr>
<td>7</td>
<td>Name of a branch site. If the partial name of a branch site displays, hover your mouse pointer over the name to see the full name.</td>
</tr>
<tr>
<td>8</td>
<td>Standard device profiles for a device in a branch site. Standard profiles are indicated by a dashed border. If the partial name of a device profile displays, hover your mouse pointer over the name to see the full name. If a device profile is attached to 1 or more devices, icons and information display as described in Row 1 of this table.</td>
</tr>
<tr>
<td>9</td>
<td>A blue shaded icon with white arrows indicates that the device profile has been attached to 1 or more devices. Shaded circle with white arrows</td>
</tr>
<tr>
<td>10</td>
<td>An unshaded icon with blue arrows indicates that the device profile has not been attached to any devices.</td>
</tr>
</tbody>
</table>
Configure Network Design Elements

With the network design feature, you can create a new overlay network topology and modify existing elements in a topology. You perform these activities from the Network Design screen.

Creating a new network topology involves performing the following procedures in the order shown:

Table 8:

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Add circuits.</td>
<td>See Configure Circuits .</td>
</tr>
<tr>
<td>2</td>
<td>Add data centers.</td>
<td>See Configure Data Centers .</td>
</tr>
<tr>
<td>3</td>
<td>Add branch sites.</td>
<td>See Configure Branch Sites .</td>
</tr>
<tr>
<td>4</td>
<td>Configure global parameters.</td>
<td>See Configure Global Parameters .</td>
</tr>
<tr>
<td>5</td>
<td>Configure device profiles.</td>
<td>See Configure Device Profiles .</td>
</tr>
<tr>
<td>6</td>
<td>Attach device profiles.</td>
<td>See Attach Device Profile .</td>
</tr>
</tbody>
</table>

A network topology must include at least one circuit and one data center. After a network topology is created, you can modify its elements directly.

Configure Circuits

Each network topology must have at least 1 circuit and can have up to 18 circuits.

To configure circuits for a network topology, follow these steps:

1. Select Configuration ► Network Design and then click Create Network Design (which displays if you have not yet created a network topology) or Manage Network Design (which displays if you have created a network topology).

2. Click Circuits near the top of the Network Design screen.

A screen for configuring circuits displays. If any circuits have been created, this screen lists them. You can remove a circuit by clicking its corresponding delete icon.

1. Click Add New Circuit.

2. Select the Private or the Public radio button to indicate whether the circuit is private or public.

3. From the Circuit Color drop-down list, choose a predefined color to uniquely identify the transport location (TLOC) in a circuit.

The color can be default, 3g, biz-internet, blue, bronze, custom1, custom2, custom3, gold, green, lte, metro-ethernet, mpls, private1, private2, public-internet, red, or silver. The color you choose cannot be used for a TLOC in any other circuit in the topology.

1. Repeat Steps 2 through 5 as needed to add more circuits.
To remove a circuit that you added, click its corresponding **Delete** icon.

1. **Click Finish.**
2. **Click Save** on the Network Design screen.

Or, if you do not want to save the updates that you made, click **Cancel**.

### Configure Data Centers

Configuring a data center involves assigning a name and adding device profiles and segments to the data center. Each network topology must have at least one data center.

To configure data centers for a network topology, follow these steps:

1. Select **Configuration** ► **Network Design** and then click **Create Network Design** (which displays if you have not yet created a network topology) or **Manage Network Design** (which displays if you have created a network topology).

2. **Click Data Center** near the top of the Network Design screen.

This option appears dimmed if you have not added at least one circuit as described in **Configure Circuits**.

A screen for configuring data centers displays. If any data centers have been created, this screen lists them. If you are creating a network topology for the first time, skip to Step 4.

1. If any data centers are listed on the screen that displays, you can take any of these actions:
   - To add another data center, click **Add Data Center** and then continue to Step 4.
   - To view information about device profiles that have been added to a data center, click the **Devices** button to the right of the data center name.
   - To view information about segments that have been added to a data center, click the **Segments** button to the right of the data center name.
   - To update configuration items for a data center, including its name, device profiles, and segments, click the pencil icon to the right of the data center name and then continue to Step 4.
   - To remove a data center from the network topology, click the trash can icon to the right of the data center name and then skip to Step 8. You cannot delete a data center that includes any device profiles that are attached to one or more devices. To delete a data center in this situation, first detach the device profiles from devices. For instructions, see **Detach Device Profile**.

1. **In the Data Center Name field**, enter a unique name for the data center.

   This name cannot be used for any other data center, branch site, or device profile in the topology. The name can include letters, numbers, underscores, and hyphens, but no spaces or special characters.

1. Take the following actions to add device profiles to the data center or to update device profile configuration settings:
Each data center must have at least one device profile. A device profile is associated with a specific device type in the data center and provides configuration settings that are pushed to those device types.

1. If you are adding a new device profile, click Add a Device Profile.

2. In the Name field, enter a name for the device profile. This name cannot be used for any other device profile, data center, or branch site in the topology. The name can include letters, numbers, underscores, and hyphens, but no spaces or special characters.

3. From the Device Model drop-down list, choose the device type with which to associate the device profile.

4. Click the Circuits field to display a list of circuits that you created as described in Configure Circuits and then check the box next to each circuit that the device profile should be associated with. The circuit names that you check appear in the Circuits field. You can remove a circuit from this field by unchecking its check box or by clicking the X next to its name. You can use the same circuit in multiple data centers and branch sites.

5. Repeat Steps 5a through 5d as needed to add more device profiles.

6. Click Next.

1. Take the following actions to add one or more segments.

Each data center must have at least one segment. A segment is a service side VPN that is associated with all device profiles in the data center. You can use the same segment in multiple data centers and branch sites.

1. Click Add Segment and choose one of these options:
   • New Segment—Creates a new segment with a new name and VPN ID
   • Existing Segment—Lets you choose a segment that you already created

1. In the Segment Name field, take one of these actions:
   • If you chose New Segment, enter a name for the segment. The name can include letters, numbers, underscores, and hyphens, but no spaces or special characters.
   • If you chose Existing Segment, choose a segment from the drop-down list. The VPN Number field populates automatically with the VPN ID that was configured for the segment.

1. If you chose New Segment, in the VPN Number field, enter a LAN side VPN ID to associate with the segment. This value cannot be used for any other VPN number in the topology. Valid values are 1 through 65535, except 512.

2. Repeat Steps 6a through 6c as needed to add more segments. To remove a segment that you added, click its corresponding Delete icon.

3. Click Add.

The system displays a list of data centers.

1. Repeat Steps 2 through 6 as needed to add more data centers.
2. Click Finish.

3. Click Save on the Network Design screen.

Or, if you do not want to save the updates that you made, click Cancel.

**Configure Branch Sites**

Configuring a branch site involves assigning a name and adding device profiles and segments to the branch site. A network topology does not require branch sites.

To configure branch sites for a network topology, follow these steps:

1. Select Configuration ► Network Design and then click Create Network Design (which displays if you have not yet created a network topology) or Manage Network Design (which displays if you have created a network topology).

2. Click Branch Sites near the top of the Network Design screen.

This option appears dimmed if you have not added at least one circuit when you added a data center as described in Configure Data Center.

A screen for configuring branch sites displays. If any circuits have been created, this screen lists them. If you are creating a network design for the first time, skip to Step 4.

1. If any branch sites are listed on the screen that displays, you can take any of these actions:

   • To add another branch site, click Add Branch and then continue to Step 4.

   • To view information about device profiles that have been added to a branch site, click the Devices button to the right of the branch site name.

   • To view information about segments that have been added to a branch site, click the Segments button to the right of the branch site name.

   • To update configuration items for a branch site, including its name, device profiles, circuits, and segments, click the pencil item to the right of the branch site name and then continue to Step 4.

   • To remove a branch site from the network topology, click the trash can icon to the right of the branch site name and then skip to Step 8. You cannot delete a branch site that includes any device profiles that are attached to one or more devices. To delete a branch site in this situation, first detach device profiles from devices. For instructions, see Detach Device Profile.

1. In the Branch Name field, enter a name for the branch site.

This name cannot be used for any other branch site, data center, or device profile in the topology. The name can include letters, numbers, underscores, and hyphens, but no spaces or special characters.

1. Take the following actions to add or update device profiles.

   Each branch site must have at least one device profile. A device profile is associated with a specific device type in the branch site and provides configuration settings that are pushed to those device types.

   1. If you are adding a new device profile, click Add a Device Profile.
2. In the Name field, enter a name for the device profile. This name cannot be used for any other device profile, data center, or branch site in the topology. The name can include letters, numbers, underscores, and hyphens, but no spaces or special characters.

3. From the Device Model drop-down list, choose the device type with which to associate the device profile.

4. Click the Circuits field to display a list of circuits that you created as described in Configure Circuits and then check the box next to each circuit that the device profile should be associated with. The circuit names that you check appear in the Circuits field. You can remove a circuit from this field by unchecking its check box or by clicking the X next to its name. You can use the same circuit in multiple data centers and branch sites.

5. Repeat Steps 5a through 5d as needed to add more device profiles.

6. Click Next.

1. Take the following actions to add one or more segments.

Each branch site must have at least one segment. A segment is a service side VPN that is associated with all device profiles in the branch site. You can use the same segment in multiple branch sites and data centers.

1. Click Add Segment and choose one of these options:

   • New Segment—Creates a new segment with a new name and VPN ID
   • Existing Segment—Lets you choose a segment that you already created

1. In the Segment Name field, take one of these actions:

   • If you chose New Segment, enter a name for the segment. The name can include letters, numbers, underscores, and hyphens, but no spaces or special characters.
   • If you chose Existing Segment, choose a segment from the drop-down list. The VPN Number field populates automatically with the VPN ID that was configured for the segment.

1. If you chose New Segment, in the VPN Number field, enter a LAN side VPN ID to associate with the segment. This value cannot be used for any other VPN number in the topology. Valid values are 1 through 65535, except 512.

2. Repeat Steps 6a through 6c as needed to add more segments. To remove a segment that you added, click its corresponding Delete icon.

3. Click Add.

The system displays a list of branch sites.

1. Repeat Steps 2 through 6 as needed to add more branch sites.

2. Click Finish.

3. Click Save on the Network Design screen.

Or, if you do not want to save the updates that you made, click Cancel.
Configure Global Parameters

Global parameters are configuration settings that are used in all device profiles in a network topology. If you do not configure global parameters, factory default configuration settings are used for device profiles.

To configure global parameters, follow these steps:

1. Select Configuration ▶ Network Design and then click Create Network Design (which displays if you have not yet created a network topology) or Manage Network Design (which displays if you have created a network topology).

2. Click Global Parameters near the top of the Network Design screen and choose the desired template from the drop-down list that displays.

A screen for configuring the selected template displays.

1. Configure the template as described in the “Create a Device Template” section in Templates.

The template name and description are filled in automatically and cannot be changed. There is no option for selecting a device type because the template is used for all devices throughout your network.

1. Click Update.

2. Click Save on the Network Design screen.

Or, if you do not want to save the updates that you made, click Cancel.

Configure Device Profiles

You must configure a device profile for each router in a data center or branch site before the device profile can be attached to the router. Configuring a profile involves configuring its TLOC, LAN side, and management interfaces, and configuring related settings.

There are two types of device profiles:

• Standard device profile—Contains basic LAN, WAN, and management interface configuration options

• Custom device profile—Contains more advanced configuration options for a variety of items such as routing and other services for the interfaces

Each new device profile that you create is saved as a standard type. After you create a standard device profile and attach it to a device, you can convert it to a custom device profile as described in the following instructions.

To configure a device profile for a router in a network topology, follow these steps:

1. Select Configuration ▶ Network Design and then click Create Network Design (which displays if you have not yet created a network topology) or Manage Network Design (which displays if you have created a network topology).

1. In the network diagram that displays on the Network Design screen, click the image that represents the device for which you want to build or modify a device profile.

The image of the device displays in one of these ways:

• Blue shaded icon—Indicates that the device has a profile. When you hover your mouse pointer over this image, “Manage profile” displays.
If you choose this option for a standard device profile, the Manage Profile screen displays. From this screen, you can modify configuration settings for the device profile or convert it to a custom device profile. Continue to Step 3.

If you choose this option for a custom device profile, a template screen displays. Skip to Step 4.

• Unshaded icon—Indicates that the device does not yet have a profile. When you hover your mouse pointer over this image, “Build profile” displays.

If you choose this option, the Build Profile screen displays. From this screen, you can create a standard device profile. Skip to Step 5.

1. If you chose to manage a device profile for a standard device profile, take one of these actions:

   • To update existing options for the standard device profile, click the pencil icon that appears near the top right of the screen for managing a profile. The Build Profile screen displays. Skip to Step 5.

   • To convert the standard device profile to a custom device profile, click Custom Profile and then click Proceed in the dialog box that pops up. A template screen displays with some options pre-populated based on options that you have already configured for this device profile. Configure the options as desired. (For information about configuring a template, see the “Create a Device Template” section in Templates.) When you are finished, click Update and then skip to Step 17.

1. If you chose to manage a custom device profile, configure the options as desired. (For information about configuring a template, see the “Create a Device Template” section in Templates.) When you are finished, click Done and then skip to Step 17.

2. If you chose to build a device profile or to manage a standard device profile, In the Interface Name field, enter the name of a TLOC interface to associate with the circuit that is associated with this router.

3. Click one of these radio buttons:

   • DHCP—Selects a dynamic IP address for the interface
   • Static—Indicates that you will assign a static IP address to the interface and a prefix and next hop to the VPN later, as described in Attach Device Profile

1. (Optional) In the DNS server field, enter the IP address of the primary DNS server in the network.

2. Click Next.

3. In the Interface Name field, enter the name of a LAN side interface to associate with the segment.

4. (Optional) In the VLAN field, enter a sub-interface, if needed for your deployment.

5. Click one of these radio buttons:

   • None—Indicates that you will assign a static IP address to this interface later, as described in Attach Device Profile
   • DHCP—Indicates that you will assign a DHCP address pool to this interface later, as described in Attach Device Profile
   • DHCP Relay—Indicates that you will assign a DHCP helper address to this interface later, as described in Attach Device Profile

1. Click Next.
2. In the Interface Name field, enter the name for the management interface to associate with the device.

3. Click one of these radio buttons:
   - **DHCP**—Selects a dynamic IP address for the interface
   - **Static**—Indicates that you will assign a static IP address to the interface and a prefix and next hop to the VPN later, as described in Attach Device Profile

1. (Optional) In the DNS server field, enter the IP address of the primary DNS server in the network.

2. Click **Done**.

3. Click **Save** on the Network Design screen.

Or, if you do not want to save the updates that you made, click **Cancel**.

**Attach, Detach, Export, Update Device Profiles**

From the Network Design screen, you can perform the following tasks for existing device profiles.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attach a device profile to devices.</td>
<td>Makes the devices available to be controlled and configured through the SD-WAN.</td>
<td>See Attach Device Profile.</td>
</tr>
<tr>
<td>Detach a device profile from devices.</td>
<td>Puts the devices into CLI mode.</td>
<td>See Detach Device Profile.</td>
</tr>
<tr>
<td>Export device profile settings</td>
<td>Creates a CSV file that contains configuration information of a selected device profile. This task is useful for backing up device profile configuration information.</td>
<td>See Export Device Profile Settings.</td>
</tr>
<tr>
<td>Change configuration information for a device profile.</td>
<td>Updates device profile configuration information on the devices to which the profile is attached.</td>
<td>See Change Device Profile Values.</td>
</tr>
</tbody>
</table>

For information about creating a device profile, see Configure Device Profiles.

**Attach Device Profile**

Attaching a device profile to devices makes the devices available to be controlled and configured through the SD-WAN. A device to which a device profile is not attached is in CLI mode.

A device can have only one device profile. The same device profile can be attached to multiple devices.

To attach a device profile to devices, follow these steps:

1. Select **Configuration ► Network Design** and then click **Attach Device**.

2. In the network diagram that displays, click the device profile that you want to attach to devices and then choose **Attach Devices** from the pop-up list.
The Attach Devices window displays.

Configure options on this window as described in the “Attach Devices to a Device Template” section in Templates.

If, when you configured a device profile, if you configured static for a TLOC interface, or DHCP or DCHP relay for a VLAN subinterface, make sure to configure the static IP address, DHCP IP address, prefix information, and next hop information, as applicable.

After you configure devices, the Network Design screen displays and the configuration updates are pushed to the selected devices.

You can click the **Device Attached Task** option near the top right of the screen to view the progress of the configuration push operation.

**Detach Device Profile**

Detaching a device profile puts the devices to which it was attached into CLI mode.

To detach a device, follow these steps:

1. Select **Configuration ► Network Design** and then click **Detach Device**.
2. In the network diagram that displays, click the device profile that you want to detach from devices and then choose **Detach Devices** from the pop-up list.

The Detach Device window displays.

1. In the Available Devices column on the left, either select a group and search for one or more devices, select a device from the list, or click **Select All**.
2. Click the arrow pointing right to move the device to the Selected Devices column on the right.
3. Click **Detach**.

The device profile is detached from the devices that you selected.

**Export Device Profile Settings**

Exporting device profile settings creates a CSV file that contains the configuration information of a selected device profile. You can save this CSV file in the location of your choice. This export feature is useful for creating a backup of device profile configuration information.

A device profile must be attached to at least one device before you can export its configuration information.

To export a CSV file, follow these steps:

1. Select **Configuration ► Network Design** and then click **Export**.
2. In the network diagram that displays, click the device profile whose configuration information you want to export and then choose **Export CSV** from the pop-up list.
3. Follow the on-screen prompts to create the CSV file and save it to the location of your choice.

**Change Device Profile Values**

Changing device profile values updates device profile configuration information on the devices to which the profile is attached.
A device profile must be attached to at least one device before you can update its configuration information. To change device values, follow these steps:

1. Select Configuration ➤ Network Design and then click Profile.

2. In the network diagram that displays, click the device profile whose configuration values you want to update and then choose Change Device Values from the pop-up list.

3. In the window that displays, use the Search field and options to locate a device to which the profile is attached.

4. Click the More Actions icon to the right of the row for the applicable device and select Edit Device Template.

5. In the Update Device Template window that pops-up, modify values as desired, and then click Update.

6. Click Next.

7. Select a device from the list of devices that displays at the left of the window.

8. Click Configure Devices to push the configuration to all devices that the device profile is attached to.

The Network Design screen displays and the configuration updates are pushed to the selected devices. You can click the Device Attached Task option near the top right of the screen to view the progress of the configuration push operation.

**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 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 the vSmart controllers and vEdge routers 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 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 vSmart controllers and vEdge routers, 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 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 vSmart controller that are executed on vEdge routers, 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 vEdge router at a branch or local site advertises OMP routes to the vSmart controllers in its domain. These routes contain routing information that the vEdge router has learned from its site-local network.

A vEdge router 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 Viptela 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 vSmart controller. See Centralized Control Policy.

**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 Viptela 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 vEdge router 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 vEdge router. The Viptela solution offers predefined
colors, which are assigned in the configuration of the vEdge routers. 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 vEdge router 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 vSmart controller. See Centralized Control
Policy.

Service Routes

Service routes represent services that are connected to a vEdge router or to the local-site network in which
the vEdge router resides. The vEdge router advertises these routes to 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 vEdge routers 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 vEdge router.

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 vSmart controller and the vEdge routers take the origin type and subtype into consideration.

Table 10:

<table>
<thead>
<tr>
<th>OMP Route Origin Type</th>
<th>OMP Route Origin Subtype</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP</td>
<td>External, Internal</td>
</tr>
<tr>
<td>Connected</td>
<td>—</td>
</tr>
<tr>
<td>OSPF</td>
<td>External-1, External-2, Intra-area, Inter-area</td>
</tr>
<tr>
<td>Static</td>
<td>—</td>
</tr>
</tbody>
</table>

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 vSmart controller or vEdge 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 Viptela devices:

Table 11:

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Administrative Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connected</td>
<td>0</td>
</tr>
<tr>
<td>Static</td>
<td>1</td>
</tr>
<tr>
<td>NAT (NAT and static routes cannot coexist in the same VPN; NAT overwrites static routes)</td>
<td>1</td>
</tr>
<tr>
<td>Learned from DHCP</td>
<td>1</td>
</tr>
<tr>
<td>GRE</td>
<td>5</td>
</tr>
<tr>
<td>EBGP</td>
<td>20</td>
</tr>
<tr>
<td>OSPF</td>
<td>110</td>
</tr>
</tbody>
</table>
OMP Best-Path Algorithm and Loop Avoidance

vEdge routers advertise their local routes to the vSmart controller using OMP. Depending on the network topology, some routes might be advertised from multiple vEdge routers. Viptela devices use the following algorithm to choose the best route:

1. Check whether the OMP route is valid. If not, ignore it.
2. If the OMP route is valid and if it has been learned from the same Viptela device, select the OMP route with the lower administrative distance.
3. If the administrative distances are equal, select the OMP route with the higher OMP route preference value.
4. On vEdge routers only, if the OMP route preference values are equal, select the OMP route with the higher TLOC preference value.
5. 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
6. If the origin type is the same, select the OMP route that has the lower origin metric.
7. On vEdge routers only, if the origin types are the same, select the OMP route with the lower router ID.
8. If the router IDs are equal, a vEdge router selects the OMP route with the lower private IP address. If a vSmart controller receives the same prefix from two different sites and if all attributes are equal, the vSmart controller chooses both of them.

Here are some examples of choosing the best route:

- A vSmart controller receives a OMP route to 10.10.10.0/24 via OMP from a vEdge router with an origin code of OSPF, and it also receives the same route from another 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 vEdge router.
- A vSmart controller learns the same OMP route, 10.10.10.0/24, from two vEdge routers in 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 vEdge router 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 vEdge router, which creates a separate BFD session with each of the remote TLOCs. If a BFD session becomes inactive, the vSmart controller removes from the forwarding table all the OMP routes that point to that TLOC.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Administrative Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBGP</td>
<td>200</td>
</tr>
<tr>
<td>OMP</td>
<td>250</td>
</tr>
</tbody>
</table>
OMP Graceful Restart

Graceful restart for OMP allows the data plane in the Viptela 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, the vEdge routers 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 vEdge router is re-established, and the vEdge router then receives updated, current network information from the vSmart controller.

When OMP graceful restart is enabled, a vEdge router and 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 vEdge router no longer detects the presence of the OMP connection to a vSmart controller, the router continues forwarding data traffic using the cached OMP information. The router also periodically checks whether the vSmart controller has again become available. When it does come back up and the vEdge router re-establishes a connection to it, the router 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 a vEdge router.

BGP and OSPF Routing Protocols

The Cisco SD-WAN overlay network supports BGP and OSPF unicast routing protocols. These protocols can be configured on vEdge routers in any VPN except for VPN 0 and VPN 512 to provide reachability to networks at their local sites. vEdge routers 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, the vEdge router acts as an MPLS CE device and establishes a BGP peering session to connect to the PE router in the L3VPN MPLS cloud.

When the vEdge router or routers 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-Viptela hub router, standard routing must be enabled on the vEdge routers’ 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 transport VPN that is responsible for carrying control traffic in the overlay network. The vBond orchestrator learns about this DTLS connection via the loopback interface and conveys this information to the vSmart controller so that it can track the TLOC-related information. In VPN 0, you also configure the physical interface that connects the vEdge router 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 Unicat 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 the vEdge router, 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 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 the vEdge router and other Viptela 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 the vEdge router to participate in the overlay network.

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

### Configure BGP Using vManage Templates

Use the Border Gateway Protocol (BGP) template for all vEdge Cloud and vEdge router devices. 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 the vEdge router and other vEdge devices when the router 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 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 NMS, 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:
   1. Click the Transport & Management VPN tab located directly beneath the Description field, or scroll to the Transport & Management VPN section.
   2. Under Additional VPN 0 Templates, located to the right of the screen, click BGP.
   3. 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:
   1. Click the Service VPN tab located directly beneath the Description field, or scroll to the Service VPN section.
2. Click the Service VPN drop-down.

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

4. 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.

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.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Shutdown</strong>*</td>
<td>Click No to enable BGP on the interface.</td>
</tr>
<tr>
<td><strong>AS number</strong>*</td>
<td>Enter the local AS number.</td>
</tr>
<tr>
<td><strong>Router ID</strong></td>
<td>Enter the BGP router ID, in decimal four-part dotted notation.</td>
</tr>
<tr>
<td><strong>Propagate AS Path</strong></td>
<td>Click On to carry BGP AS path information into OMP.</td>
</tr>
<tr>
<td><strong>Internal Routes Distance</strong></td>
<td>Enter a value to apply as the BGP route administrative distance for routes coming from one AS into another. Range: 0 through 255 Default: 0</td>
</tr>
<tr>
<td><strong>Local Routes Distance</strong></td>
<td>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. Range: 0 through 255 Default: 0</td>
</tr>
</tbody>
</table>
Configure BGP using vManage Templates

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Routes Distance</td>
<td>Specify the BGP route administrative distance for routes learned from other sites in the overlay network.</td>
</tr>
<tr>
<td></td>
<td>Range: 0 through 255</td>
</tr>
<tr>
<td></td>
<td>Default: 0</td>
</tr>
</tbody>
</table>

For service-side BGP, you might want to configure Overlay Management Protocol (OMP) to advertise to the vSmart controller any BGP routes that the vEdge router learns. By default, a vEdge router advertises to OMP both the connected routes on the vEdge router and the static routes that are configured on the vEdge router, but it does not advertise BGP external routes learned by the vEdge router. You configure this route advertisement in the OMP template for vEdge routers or vEdge 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 BGP Neighbors**

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

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Options</th>
<th>Sub-Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 / IPv6</td>
<td>Click IPv4 to configure IPv4 neighbors. Click IPv6 to configure IPv6 neighbors.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address/IPv6 Address</td>
<td>Specify the IP address of the BGP neighbor.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Enter a description of the BGP neighbor.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remote AS</td>
<td>Enter the AS number of the remote BGP peer.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For BGP to function, you must configure at least one neighbor.
### Address Family
- **Options**: 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
- **Sub-Options**: Select the address family. Currently, the software supports only the BGP IPv4 unicast address family.

### Maximum Number of Prefixes
- **Description**: Specify the maximum number of prefixes that can be received from the neighbor.
  - **Range**: 1 through 4294967295
  - **Default**: 0

### Threshold
- **Description**: 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
- **Description**: How long to wait to restart the BGP connection. Range: 1 through 65535 minutes

### Warning Only
- **Description**: Click On to display a warning message only, without restarting the BGP connection.

### Route Policy In
- **Description**: Click On and specify the name of a route policy to apply to prefixes received from the neighbor.

### Route Policy Out
- **Description**: Click On and specify the name of a route policy to apply to prefixes sent to the neighbor.

### Configure Advanced Neighbor Parameter

To configure advanced parameters for the neighbor, click the Neighbor tab, and then click **Advanced Options**.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next-Hop Self</td>
<td>Click On to configure the router to be the next hop for routes advertised to the BGP neighbor.</td>
</tr>
<tr>
<td>Send Community</td>
<td>Click On to send the local router’s BGP community attribute to the BGP neighbor.</td>
</tr>
<tr>
<td>Send Extended Community</td>
<td>Click On to send the local router’s BGP extended community attribute to the BGP neighbor.</td>
</tr>
<tr>
<td>Negotiate Capability</td>
<td>Click On to allow the BGP session to learn about the BGP extensions that are supported by the neighbor.</td>
</tr>
<tr>
<td>Source Interface Address</td>
<td>Enter the IP address of a specific interface of the neighbor that BGP is to use for the TCP connection to the neighbor.</td>
</tr>
<tr>
<td>Source Interface Name</td>
<td>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 <code>ge/port/slot</code>.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EBGP Multihop</td>
<td>Set the time to live (TTL) for BGP connections to external peers. Range: 0 to 255 Default: 1</td>
</tr>
<tr>
<td>Password</td>
<td>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.</td>
</tr>
<tr>
<td>Keepalive Time</td>
<td>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 to be available. Specify the keepalive time for the neighbor, to override the global keepalive time. Range: 0 through 65535 seconds Default: 60 seconds (one-third the hold-time value)</td>
</tr>
<tr>
<td>Hold Time</td>
<td>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. Range: 0 through 65535 seconds Default: 180 seconds (three times the keepalive timer)</td>
</tr>
<tr>
<td>Connection Retry Time</td>
<td>Specify the number of seconds between retries to establish a connection to a configured BGP neighbor peer that has gone down. Range: 0 through 65535 seconds Default: 30 seconds</td>
</tr>
<tr>
<td>Advertisement Interval</td>
<td>For the BGP neighbor, set the minimum route advertisement interval (MRAI) between when BGP routing update packets are sent to that neighbor. Range: 0 through 600 seconds Default: 5 seconds for IBGP route advertisements; 30 seconds for EBGP route advertisements</td>
</tr>
</tbody>
</table>

To save the feature template, click Save.

**CLI Equivalent**

```plaintext
vpn vpn-id router bgp local-as-number
neighbor ip-address address-family ipv4-unicast
maximum-prefixes number [threshold] [restart minutes | warning-only]
route-policy policy-name (in | out)
capability-negotiate description string ebgp-multihop ttl
next-hop-self password md5-digest-string remote-as remote-as-number
send-community send-ext-community [no] shutdown
timers advertisement-interval number connect-retry seconds
holdtime seconds keepalive seconds update-source ip-address
```
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:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| 🔄 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.  
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.  
To change the default key, type a new string and move the cursor out of the Enter Key box.  
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.  
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:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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. This hold time is the global hold time.  
Range: 0 through 65535 seconds  
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 router is still active and should be considered to be available. This keepalive time is the global keepalive time.  
Range: 0 through 65535 seconds  
Default: 60 seconds (one-third the hold-time value) |
<p>| Compare MED | Click On to compare the router 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. |</p>
<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing MED as Worst</td>
<td>Click On to consider a path as the worst path if the path is missing a MED attribute.</td>
</tr>
<tr>
<td>Compare Router ID</td>
<td>Click On to always compare MEDs regardless of whether the peer ASs of the compared routes are the same.</td>
</tr>
<tr>
<td>Multipath Relax</td>
<td>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.</td>
</tr>
</tbody>
</table>

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

**Configure BGP Using CLI**

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 vEdge router, 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:

   1. 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).
      ```

   2. 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 vEdge router:
   
   ```
   vEdge(config)# system system-ipaddress
   ```

**Example of BGP Configuration on a vEdge Router**

```text
vEdge# show running-config system
system
  system-ip 10.1.2.3
!
vEdge# show running-config vpn 1
vpn 1
  router
```
bgp
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 vEdge router then advertises the OMP routes to all the BGP routers in the service-side of the network. To redistribute OMP routes into BGP so that these routes are advertised to all BGP routers in the service side of the network, configure redistribution in any VPN except VPN 0 or VPN 512:

vEdge(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:

vEdge(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) [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 vEdge router to advertise BGP routes that it has learned, through OMP, from the vSmart controller. Doing so allows the vSmart controller to advertise these routes to other vEdge routers 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

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

vEdge(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:

\[
\text{vEdge(config)} \# \text{omp} \\
\text{vEdge(omp)} \# \text{overlay-as as-number}
\]

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:**

\[
\text{vEdge(config)} \# \text{vpn 0 interface ge/slot/port ip address address} \\
\text{vEdge(config-interface)} \# \text{no shutdown}
\]

2. **Configure a loopback interface in VPN 0:**

\[
\text{vEdge(config)} \# \text{vpn 0 interface loopback number ip address address} \\
\text{vEdge(config-interface)} \# \text{no shutdown} \\
\text{vEdge(config-interface)} \# \text{tunnel-interface color color}
\]

3. **Configure a BGP instance in VPN 0:**

\[
\text{vEdge(config)} \# \text{vpn 0 router bgp local-as-number}
\]

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

\[
\text{vEdge(config)} \# \text{policy lists prefix-list prefix-list-name ip-prefix prefix} \\
\quad \text{prefix is the IP address of the loopback interface.}
\]

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

\[
\text{vEdge(config)} \# \text{policy route-policy policy-name sequence number match address prefix-list-name} \\
\text{vEdge(config)} \# \text{policy route-policy policy-name sequence number action accept} \\
\text{vEdge(config)} \# \text{policy route-policy policy-name default-action reject}
\]

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

\[
\text{vEdge(config)} \# \text{vpn 0 router bgp local-as-number address-family ipv4-unicast redistribute connected route-policy policy-name}
\]

To apply the policy only to a specific neighbor:

\[
\text{vEdge(config)} \# \text{vpn 0 router bgp local-as-number neighbor neighbor-address address-family ipv4-unicast redistribute connected route-policy policy-name out}
\]

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 vEdge Cloud and vEdge router devices.

To configure OSPF on vEdge routers using 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 vEdge router and other vEdge 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:
   1. Click the Transport & Management VPN tab located directly beneath the Description field, or scroll to the Transport & Management VPN section.
   2. Under Additional VPN 0 Templates, located to the right of the screen, click OSPF.
   3. 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:
   1. Click the Service VPN tab located directly beneath the Description field, or scroll to the Service VPN section.
   2. Click the Service VPN drop-down.
   3. Under Additional VPN Templates, located to the right of the screen, click OSPF.
   4. 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.
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 12:

<table>
<thead>
<tr>
<th>Parameter Scope</th>
<th>Scope Description</th>
</tr>
</thead>
</table>
| 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.  

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.  

To change the default key, type a new string and move the cursor out of the Enter Key box.  

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.  

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 13:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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.  

**Range:** 0 through 255  
**Default:** 110 |
| Distance for External Routes | Specify the OSPF route administration distance for routes learned from other domains.  

**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.  

**Range:** 0 through 255  
**Default:** 110 |
| Distance for intra-Area routes | Specify the OSPF route administration distance for routes within an area.  

**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 vEdge routers, select the Redistribute tab. Click **Add New Redistribute** and configure the following parameters:

**Table 14:**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>Select the protocol from which to redistribute routes into OSPF. Select from BGP, Connected, NAT, OMP, and Static.</td>
</tr>
<tr>
<td>Route Policy</td>
<td>Enter the name of a localized control policy to apply to routes before they are redistributed into OSPF.</td>
</tr>
</tbody>
</table>

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 routers do not prefer this vEdge router as an intermediate hop in their Shortest Path First (SPF) calculation, select the Maximum Metric (Router LSA) tab. Then click **Add New Router LSA** and configure the following parameters:

**Table 15:**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Type           | Select a type:  
|                | • Administrative—Force the maximum metric to take effect immediately, through operator intervention.  
|                | • 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.  
|                  | Range: 0, 5 through 86400 secondsDefault: 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 vEdge router, select the Area tab and click **Add New Area**. For OSPF to function, you must configure area 0.
### Table 16:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area Number</td>
<td>Enter the number of the OSPF area. <em>Range: 32-bit number</em></td>
</tr>
<tr>
<td>Set the Area Type</td>
<td>Select the type of OSPF area, Stub or NSSA.</td>
</tr>
<tr>
<td>No Summary</td>
<td>Select On to not inject OSPF summary routes into the area.</td>
</tr>
</tbody>
</table>
| Translate      | If you configured the area type as NSSA, select when to allow vEdge routers that are ABRs (area border routers) to translate Type 7 LSAs to Type 5 LSAs:  
  - 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.  
  - candidate—Router offers translation services, but does not insist on being the translator.  
  - never—Translate no Type 7 LSAs |

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

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

**CLI equivalent:**

```
vpn vpn-id  
router ospf area number nssa  
  no-summary  
  translate {always | candidate | never}  
  stub  
  no-summary
```

### Configure Interfaces in an OSPF Area

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

### Table 17:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Name</td>
<td>Enter the name of the interface, in the format <code>ge slot/port</code> or <code>loopback number</code>.</td>
</tr>
<tr>
<td>Hello Interval</td>
<td>Specify how often the router sends OSPF hello packets. <em>Range: 1 through 65535 seconds</em> <em>Default: 10 seconds</em></td>
</tr>
<tr>
<td>Dead Interval</td>
<td>Specify how often the vEdge router must receive an OSPF hello packet from its neighbor. If no packet is received, the vEdge router assumes that the neighbor is down. <em>Range: 1 through 65535 seconds</em> <em>Default: 40 seconds (4 times the default hello interval)</em></td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| LSA Retransmission Interval | Specify how often the OSPF protocol retransmits LSAs to its neighbors.  
  *Range*: 1 through 65535 seconds  
  *Default*: 5 seconds |
| Interface Cost         | Specify the cost of the OSPF interface.  
  *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 18:**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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:  
  - Broadcast network—WAN or similar network.  
  - Point-to-point network—Interface connects to a single remote OSPF router.  
  *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 |
| Authentication          | Specify the authentication and authentication key on the interface, to allow OSPF to exchange routing update information securely:  
  - **Authentication Type**  
    - Simple authentication—Password is sent in clear text.  
    - 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**.

**CLI equivalent:**

```
vpn vpn-id
router ospf area number
interface interface-name
authentication
  authentication-key key
  message-digest key
  type (message-digest | simple)
  cost number
dead-interval seconds
hello-interval seconds
network (broadcast | point-to-point)
  passive-interface
  priority number
retransmit-interval seconds
```

**Configure an Interface Range for Summary LSAs**

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

**Table 19:**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>Enter the IP address and subnet mask, in the format <code>prefix/length</code>, for the IP addresses to be consolidated and advertised.</td>
</tr>
<tr>
<td>Cost</td>
<td>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. <em>Range:</em> 0 through 16777215</td>
</tr>
<tr>
<td>No Advertise</td>
<td>Select On to not advertise the Type 3 summary LSAs or Off to advertise them.</td>
</tr>
</tbody>
</table>

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

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

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

**CLI equivalent:**

```
vpn vpn-id
router ospf area number range prefix/length
  cost number
  no-advertise
```

**Configure Other OSPF Properties**

To configure other OSPF properties, select the Advanced tab and configure the following properties:
Table 20:

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

To save the feature template, click Save.

CLI equivalent:

```plaintext
vpn vpn-id
router
ospf auto-cost reference-bandwidth mbps
compatible rfc1583
default-information
originverse (always | metric metric | metric-type type)
route-policy policy-name in
	spf delay initial-hold-time maximum-hold-time
```

Release Information

Introduced in vManage NMS in Release 15.2.
Configure OSPF Using CLI

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

Configure Basic Service-Side OSPF

To set up routing on the vEdge router, 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 vSmart controller any BGP and OSPF external routes that the vEdge router 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 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
```
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
    
  
  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 vEdge Cloud and vEdge router devices, and for vSmart controllers.

OMP is enabled by default on all vEdge routers, vManage NMSs, and vSmart controllers, so there is no need to explicitly enable OMP. OMP must be operational for the Viptela overlay network to function. If you disable it, you disable the overlay network.

Create OMP 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. 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 21:

<table>
<thead>
<tr>
<th>Parameter Scope</th>
<th>Scope Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Specific (indicated by a host icon)</td>
<td>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. 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. To change the default key, type a new string and move the cursor out of the Enter Key box. Examples of device-specific parameters are system IP address, hostname, GPS location, and site ID.</td>
</tr>
<tr>
<td>Global (indicated by a globe icon)</td>
<td>Enter a value for the parameter, and apply that value to all devices. Examples of parameters that you might apply globally to a group of devices are DNS server, syslog server, and interface MTUs.</td>
</tr>
</tbody>
</table>

Configure Basic OMP Options

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

Table 22:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graceful Restart for OMP</td>
<td>Ensure that Yes is selected to enable graceful restart. By default, graceful restart for OMP is enabled.</td>
</tr>
<tr>
<td>Overlay AS Number (on vEdge routers only)</td>
<td>Specify a BGP AS number that OMOP advertises to the router's BGP neighbors.</td>
</tr>
<tr>
<td>Graceful Restart Timer</td>
<td>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)</td>
</tr>
<tr>
<td>Number of Paths Advertised per Prefix</td>
<td>Specify the maximum number of equal-cost routes to advertise per prefix. vEdge routers advertise routes to vSmart controllers, and the controllers redistributes the learned routes, advertising each route-TLOC tuple. A vEdge router can have up to four TLOCs, and by default advertises each route-TLOC tuple to the vSmart controller. If a local site has two vEdge routers, a 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</td>
</tr>
</tbody>
</table>
Configure OMP Using vManage Templates

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

**Table 23:**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advertisement Interval</td>
<td>Specify the time between OMP Update packets.</td>
</tr>
<tr>
<td></td>
<td>Range: 0 through 65535 seconds Default: 1 second</td>
</tr>
<tr>
<td>Hold Time</td>
<td>Specify how long to wait before closing the OMP connection to a peer.</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td>EOR Timer</td>
<td>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)</td>
</tr>
</tbody>
</table>

To save the feature template, click Save.

Configure OMP Advertisements

To advertise routes learned locally by the vEdge router to OMP, select the Advertise tab and configure the following parameters:
Table 24:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Advertise (on vEdge routers only) | Click On or Off to enable or disable the vEdge router advertising to OMP the routes that it learns locally:  
  - BGP—Click On to advertise BGP routes to OMP. By default, BGP routes are not advertised to OMP.  
  - Connected—Click Off to disable advertising connected routes to OMP. By default, connected routes are advertised to OMP.  
  - 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.  
  - Static—Click Off to disable advertising static routes to OMP. By default static routes are advertised to OMP.  
To configure per-VPN route advertisements to OMP, use the VPN feature template.

To save the feature template, click Save.

Configure OMP Using CLI

By default, OMP is enabled on all vEdge routers 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.

Configure OMP Graceful Restart

OMP graceful restart is enabled by default on vSmart controllers and vEdge routers. 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 vEdge router and 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 vEdge router that is configured with a timer of 600 seconds (10 minutes). Here, the vSmart controller retains the OMP routes learned from that router for 10 minutes—the graceful restart timer value that is configured on the router and that the router has sent to the vSmart controller during the setup of the OMP session. The vEdge router 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 router, also during the setup of the OMP session.
While a vSmart controller is down and a vEdge router is using cached OMP information, if you reboot the vEdge router, 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 vEdge router advertises connected, static routes, and OSPF inter-area and intra-area routes to OMP, and hence to the vSmart controller responsible for the vEdge router's domain. The router does not advertise BGP or OSPF external routes to OMP.

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

To configure the routes that the vEdge router advertises to OMP for all VPNs configured on the router:

```plaintext
vEdge(config-omp)# advertise (bgp | connected | ospf type | static)
```

To configure the routes that the vEdge router advertises to OMP for a specific VPN on the router:

```plaintext
vEdge(config-vpn-omp)# advertise (aggregate prefix [aggregate-only] | bgp | connected | network prefix | ospf type | static)
```

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 router. 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:

```plaintext
vEdge(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, 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 Viptela overlay network and an MPLS underlay network—you can assign as 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:

```plaintext
vEdge(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
Configure OMP Using CLI

Configure OMP Using CLI

**Configure the Number of Advertised Routes**

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

The vSmart controller redistributes the routes it learns from vEdge routers, advertising each route–TLOC tuple. If, for example, a local site as two vEdge routers, a vSmart controller could potentially learn eight route–TLOC tuples for the same route.

By default, vEdge routers and vSmart controllers advertise 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 number
```

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

**Configure the Number of Installed OMP Paths**

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

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

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. To modify the OMP hold time interval:

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

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 update-interval seconds
```

The update interval can be in the range 0 through 65535 seconds.
Device(config-omp)# timers advertisement-interval seconds

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 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. To modify the EOR timer:

Device(config-omp)# timers eor-timer seconds

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

---

**Use Case: Unicast Routing in Viptela Overlay Network**

This example illustrates how to set up unicast routing in a Viptela overlay network. This network consists of one vBond orchestrator, one vSmart controller, and two vEdge routers at two different sites, as shown in the figure. The table following the figure shows the parameters for these devices.

![Diagram of Viptela Overlay Network](image)

**Table 25:**

<table>
<thead>
<tr>
<th></th>
<th>vBond Orchestrator</th>
<th>vSmart Controller</th>
<th>vEdge-1 Router</th>
<th>vEdge-2 Router</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public IP address</td>
<td>184.168.0.69</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Domain</td>
<td>—</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Site ID</td>
<td>—</td>
<td>50</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
### vSmart Controller Configuration

On the vSmart controller:

1. Configure the system properties of the vSmart controller:

```bash
vSmart1(config)# system system-ip 1.1.1.9
vSmart1(config)# domain-id 1
vSmart1(config)# site-id 50
vSmart1(config)# vbond 184.168.0.69
```

2. Configure the physical transport information:

   1. Configure the IP address of the WAN-facing interface:

```bash
vSmart1(config)# vpn 0 interface eth4 ip address 10.0.16.19/24
```

   2. Allow the interface to carry control-plane traffic:

```bash
vSmart1(config)# vpn 0 interface eth4 tunnel-interface
```

   3. Enable the interface:

```bash
vSmart1(config)# vpn 0 interface eth4 no shutdown
```

   4. Configure a default route to the WAN interface that connects to the transport cloud:

```bash
vSmart1(config)# vpn 0 ip route 0.0.0.0/0 10.0.16.19
```

3. Commit the configuration:

```bash
vSmart1(config)# commit
```

Here is the full configuration on the vSmart controller:

```bash
system system-ip 1.1.1.9 domain-id 1 site-id 50 vbond 184.168.0.69!
vpn 0
interface eth4 ip address 10.0.16.19/24 tunnel-interface ! no shutdown ! ip
route 0.0.0.0/0 10.0.16.19!
```
vEdge1 Router Configuration

On the vEdge1 router:

1. OMP is enabled by default on the vEdge router. This branch network is running BGP, and we want to advertise BGP-learned routes to the vSmart controller:

   ```
   vEdge1(config)#  omp advertise bgp
   ```

2. Configure the system properties of the vEdge1 router and the IP address of the vBond orchestrator:

   ```
   vEdge1(config)#  system system-ip 1.1.1.5
   vEdge1(config)#  domain-id 1
   vEdge1(config)#  site-id 1
   vEdge1(config)#  vbond 184.168.0.69
   ```

3. Configure the transport VPN and the transport interface:

   ```
   vEdge1(config)#  vpn 0 interface ge1/1 ip address 75.0.13.15/24
   vEdge1(config-ge1-1)#  tunnel-interface
   ```

4. Configure the default route in the transport VPN:

   ```
   vEdge1(config-vpn-0)#  ip route 0.0.0.0/0 75.0.13.15
   ```

5. For the VPN, configure BGP:

   1. Configure the local AS number:
   
   ```
   vEdge1(config)#  vpn 1 router bgp 1
   ```

   2. Have BGP advertise only unicast traffic:
   
   ```
   vEdge1(config-bgp-1)#  address-family ipv4_unicast
   ```

   3. Have BGP redistribute routes that it learns, via the OMP session, from the vSmart controller:
   
   ```
   vEdge1(config-bgp-1)#  redistribute omp
   ```

   4. Enable and configure the BGP peer:
   
   ```
   vEdge1(config-bgp-1)#  neighbor 10.0.17.17 remote-as 2
   ```

   5. Configure the interface between the vEdge1 router and its local-site router:
   
   ```
   vEdge1(config-bgp-1)#  vpn 1 interface ge0/1
   ```

6. Activate the configuration:

   ```
   vEdge1(config)#  commit
   ```

Here is the full configuration on the vEdge1 router:

```
OMP no shutdown advertise bgp!system system-ip 1.1.1.5 domain-id 1 site-id 1 vbond 184.168.0.69!vpn 0 interface ge1/1 ip address 75.0.13.15/24 tunnel-interface ! no shutdown ! ip route 0.0.0.0/0 75.0.13.15!vpn 1 router bgp 1 address-family ipv4_unicast redistribute omp ! neighbor 10.0.17.17 remote-as 2 ! ! interface ge0/1 ip address 10.0.19.15/24 !
```

vEdge2 Router Configuration

On the vEdge2 router:

1. OMP is enabled by default on the vEdge router. This branch network is running OSPF, and OSPF automatically redistributes its learned intra-area and inter-area routes to the vSmart controller. We also want to advertise OSPF external routes to the vSmart controller:
vEdge2(config)# omp advertise ospf external

2. Configure the system properties of the vEdge2 router and the IP address of the vBond orchestrator:

vEdge2(config)# system system-ip 1.1.1.6  
vEdge2(config)# domain-id 1  
vEdge2(config)# site-id 2  
vEdge2(config)# vbond 184.168.0.69

3. Configure the transport VPN and the transport interface and the default route:

vEdge2(config)# vpn 0 interface ge2/1 ip address 172.16.10.16/24  
vEdge2(config)# tunnel-interface

vEdge2(config-g2-1)# no shutdown

4. Configure the default route in the transport VPN:

vEdge2(config-vpn-0)# ip route 0.0.0.0/0 172.16.10.16

5. For the VPN, configure OSPF:

1. Configure area 0 and add the ge0/2 interface to this area:

vEdge2(config)# vpn 2 router ospf area 0  
interface ge0/2

2. Configure the interface between the vEdge2 router and its local-site router:

vEdge2(config)# vpn 2  
interface ge0/2 ip address 172.16.7.16/24  
vEdge2(config-interface-eth1)# no shutdown

6. Activate the configuration:

vEdge1(config)# commit

Here is the full configuration on the vEdge2 router:

omp no shutdown advertise ospf external!system system-ip 1.1.1.6 domain-id 1 site-id 2  
vbond 184.168.0.69!vpn 0 interface ge2/1 ip address 172.16.10.16/24 tunnel-interface  
! no shutdown ! ip route 0.0.0.0/0 172.16.10.16!vpn 2 router ospf area 0  
interface ge0/2 exit exit !! interface ge0/2 ip address 172.16.7.16/24  
no shutdown !!

# Unicost Overlay Routing CLI Reference

CLI commands for configuring and monitoring the BGP, OMP, OSPF, and routing protocols on vEdge routers.

Routing Protocol Configuration Commands

You configure routing protocols on vEdge routers. For BGP and OSPF, you configure them in each VPN:

omp...  
vpn vpn-id  
router  
bgp...  
igmp...  
multicast-replicator...  
ospf...  
pim...
**BGP Configuration and Monitoring Commands**

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

```
vpn vpn-id
  router
    bgp local-as-number
    address-family ipv4-unicast
      aggregate-address prefix/length [as-set] [summary-only]
      maximum-paths paths number
      network prefix/length
      redistribute {connected | nat | ospf | static} [route-policy policy-name]
      best-path
      as-path multipath-relax
      compare-router-id
      med {always-compare | deterministic | missing-as-worst}
      distance
        external number
        internal number
        local number
  neighbor ip-address
    address-family ipv4-unicast
      maximum-prefixes number [threshold] [restart minutes | warning-only]
      route-policy policy-name (in | out)
      capability-negotiate
      description text
      ebgp-multihop ttl
      next-hop-self
      password md5-digest-string
      remote-as remote-as-number
      send-community
      send-ext-community
      [no] shutdown
  timers
    advertisement-interval number
    connect-retry seconds
    holdtime seconds
    keepalive seconds
    update-source ip-address
  ! end neighbor configuration
  propagate-aspath
  router-id ip-address
  [no] shutdown
  timers
```

Use the following commands to monitor BGP on a vEdge router:

- **clear bgp all** — Reset the connections to all BGP neighbors in the specified VPN.
- **clear bgp neighbor** — Reset the connection to a specific BGP neighbor in the specified VPN.
- **debug bgp events** — Display the events that have occurred as part of the BGP finite state model.
- **debug bgp fsm** — Display the states in the BGP finite state model, which describes the actions that BGP takes and the BGP packets that are exchanged between BGP neighbors when they are establishing a peering session.
- **debug bgp ipcs** — Display information about BGP interprocess communication with other processes running on the vEdge router.
- **debug bgp packets v** — Display the packets that BGP is receiving from its peers.
• **show bgp neighbor** — Display information about all BGP connections to neighbors.
• **show bgp routes** — Display information about all routes learned by BGP.
• **show bgp summary** — Display summary information about BGP.

**OMP Configuration and Monitoring Commands**

By default, OMP is enabled on all vEdge routers and vSmart controllers. Use the following commands to modify the OMP configuration:

```plaintext
omp
  advertise (bgp | connected | ospf | static) (on vEdge routers only)
  discard-rejected (on vSmart controllers only)
  ecmp-limit number (on vEdge routers only)
  graceful-restart
  overlay-as as-number
  send-backup-paths (on vSmart controllers only)
  send-path-limit number
  [no] shutdown
  timers
    advertisement-interval seconds
    eor-timer seconds
    graceful-restart-timer seconds
    holdtime seconds
vpn
  omp
    advertise (aggregate prefix [aggregate-only] | bgp | connected | network prefix | ospf
             type | static) (on vEdge routers only)
```

Use the following commands to monitor OMP:

• **clear omp all** — Restart OMP and all OMP sessions.
• **clear omp peer** — Restart the OMP session to a specific peer
• **clear omp routes** — Flush the OMP routes from the route table and then re-install them.
• **clear omp tlocs** — Remove TLOCs from the tunnel table and then re-install them.
• **debug omp events [level level]** — Display information to help debug OMP events.
• **debug omp ipcs [level level]** — Display information to help debug OSPF interprocess communication with other processes running on the Viptela device.
• **debug omp packets [level level]** — Display information to help debug OMP packets.
• **show omp peers** — Display information about active OMP peering sessions.
• **show omp routes** — Display OMP route information.
• **show omp services** — Display OMP services information.
• **show omp summary** — Display summary information about OMP.
• **show omp tloc-paths** — Display TLOC path information.
• **show omp tlocs** — Display TLOC information.
OSPF Configuration and Monitoring Commands

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

```
vpn vpn-id
  router
    ospf
      area number
        interface interface-name
          authentication
            authentication-key key
            message-digest key
              type (message-digest | simple)
            cost number
            dead-interval seconds
            hello-interval seconds
            network (broadcast | point-to-point)
            passive-interface
            priority number
            retransmit-interval seconds
          ! end area interface
        nssa
          no-summary
          translate (always | candidate | never)
        range prefix/length
          cost number
          no-advertise
        stub
          no-summary
          ! end area
        auto-cost reference-bandwidth mbps
        compatible rfc1583
        default-information
          originate (always | metric metric | metric-type type)
        distance
          external number
          inter-area number
          intra-area number
        max-metric
        router-lsa (administrative | on-startup seconds)
        redistribute (bgp | connected | nat | omp | static) route-policy policy-name
        route-policy policy-name in
        router-id ip-v4-address
        timers
          spf delay initial-hold-time maximum-hold-time
```

Use the following commands to monitor OSPF on a vEdge router:

- **clear ospf all** — Reset all OSPF neighbors in the specified VPN.
- **debug ospf events** — Display information to help debug OSPF events in the specified VPN.
- **debug ospf ipcs** — Display information to help debug OSPF interprocess communication with other processes running on the vEdge router.
- **debug ospf ism** — Display information about the OSPF interface state machine in the specified VPN, to help debug issues with establishing an OSPF session on an interface.
- **debug ospf lsa** — Display information to help debug OSPF link-state advertisements in the specified VPN.
- **debug ospf nsm** — Display information about the OSPF neighbor state machine in the specified VPN, to help debug issues with establishing an OSPF session with a neighbor.
• **debug ospf nssa** — Display information to help debug an OSPF NSSA (Not-So-Stubby Area)

• **debug ospf packets** — Display information about OSPF packets in the specified VPN, to help debug issues related to the exchange of packets between OSPF routers.

• **show ospf database** — Display the entries in the OSPF link-state database.

• **show ospf database-summary** — Display a summary of the entries in the OSPF link-state database.

• **show ospf interface** — List the interfaces in the specified VPN that are running OSPF.

• **show ospf neighbor** — List the OSPF neighbors in the specified VPN.

• **show ospf process** — List the OSPF processes running in the specified VPN.

• **show ospf routes** — Display information about all routes learned by OSPF.
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.

The Cisco SD-WAN design optimizes multicast packet distribution throughout the overlay network by eliminating packet replication on the ingress router, that is, on the router connected to a multicast source. Instead, the ingress router forwards multicast streams to a vEdge router that is designated to be a replicator, and it is this router that forwards streams to multicast receivers. This design saves bandwidth and computational resources on the ingress router. For information on designing your overlay network, see Design Overlay Network Using vManage.

The figure below illustrates the Cisco SD-WAN design. Here, the ingress router vEdge-2 forwards the multicast streams from the source at its site to vEdge-1, which is a vEdge router that is designated to be a replicator. vEdge-1 replicates the stream and forwards it to the receiver, which is located behind vEdge-3 at Site 3.
Supported Protocols

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

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 rendezvous 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 rendezvous 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 routers 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 forwards 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.
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 26:

<table>
<thead>
<tr>
<th>Parameter Scope</th>
<th>Scope Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Specific</td>
<td>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. 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. To change the default key, type a new string and move the cursor out of the Enter Key box. Examples of device-specific parameters are system IP address, hostname, GPS location, and site ID.</td>
</tr>
<tr>
<td>Global</td>
<td>Enter a value for the parameter, and apply that value to all devices. Examples of parameters that you might apply globally to a group of devices are DNS server, syslog server, and interface MTUs.</td>
</tr>
</tbody>
</table>

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 27:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shutdown*</td>
<td>Ensure that No is selected, to enable PIM.</td>
</tr>
<tr>
<td>Auto-RP</td>
<td>Click On to enable auto-RP to enable automatic discovery of rendezvous points (RPs) in the PIM network so that the router receives group-to-RP mapping updates. By default, auto-RP is disabled.</td>
</tr>
<tr>
<td>SPT Threshold</td>
<td>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.</td>
</tr>
</tbody>
</table>
| Replicator        | For a topology that includes multicast replicators, determine how the replicator for a multicast group is chosen:  
|                   | - Random—Choose the replicator at random.  
|                   | - 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>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter the name of an interface that participates in the PIM domain, in the format ge slot/port.</td>
</tr>
<tr>
<td>Hello Interval</td>
<td>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</td>
</tr>
<tr>
<td>Join/Prune Interval</td>
<td>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</td>
</tr>
</tbody>
</table>

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:
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:

```
interface-name
ip address
prefix
```

```
no shutdown
```

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:

```
router pim replicator-selection random
```

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

```
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:

```
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:

```
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:

```
multicast-replicator threshold number
```

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

```
vpn
```

```
vpn-id
```

```
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.
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 29:

<table>
<thead>
<tr>
<th>Parameter Scope</th>
<th>Scope Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Specific (indicated by a host icon)</td>
<td>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. 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. To change the default key, type a new string and move the cursor out of the Enter Key box. Examples of device-specific parameters are system IP address, hostname, GPS location, and site ID.</td>
</tr>
<tr>
<td>Global (indicated by a globe icon)</td>
<td>Enter a value for the parameter, and apply that value to all devices. Examples of parameters that you might apply globally to a group of devices are DNS server, syslog server, and interface MTUs.</td>
</tr>
</tbody>
</table>

### 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 30:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shutdown</td>
<td>Ensure that No is selected to enable IGMP.</td>
</tr>
<tr>
<td>Interface Name</td>
<td>Enter the name of the interface to use for IGMP. To add another interface, click the plus sign (+). To delete an interface, click the trash icon to the right of the entry.</td>
</tr>
<tr>
<td>Join Group Address</td>
<td>Click Add Join Group Address, and enter the address of a multicast group for the interface to join. Click Add to add the new interface.</td>
</tr>
</tbody>
</table>

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 multicast replicators:

• 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 statistics 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.
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 useful functionality, but the 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, on page 92
- VPNs Used in Cisco SD-WAN Segmentation, on page 94
- Configure Segmentation, on page 94
- Segmentation (VPN) Configuration Examples, on page 110
- Segmentation CLI Reference, on page 118

Segmentation in Cisco SD-WAN

Cisco SD-WAN employs the more prevalent and scalable model of creating segments. Essentially, segmentation is done at the edges, on a vEdge 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:

- vEdge-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).

• vEdge-2 subscribes to the red VPN.

• vEdge-3 subscribes to the blue VPN.

Because each vEdge 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, vEdge-1 has two VPN route tables, and vEdge-2 and vEdge-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. A VPN ID carried in the packet identifies each VPN on a link. When you configure a VPN on a vEdge router, the VPN has a label associated with it. The vEdge router sends the label, along with the VPN ID, to the vSmart controller. The vSmart controller propagates this vEdge-to-VPN-ID mapping information to the other vEdge routers in the domain. The remote vEdge routers then use this label to send traffic to the appropriate VPN. The local vEdge 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 vEdge routers is completely unaware of the VPNs. Only the vEdge 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 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 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.

**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.

**Configure Segmentation**

In the Cisco SD-WAN overlay network, VPNs divide the network into different segments. By default, two VPNs are present in the configurations of all Cisco SD-WAN devices, and these VPNs serve specific purposes:

- 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 512 or management VPN 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.

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 vEdge routers. (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.
The topics in this section provide basic configuration procedures for the three types of VPNs.

**Configure VPNs Using vManage Templates**

**Create a VPN Template**

**Step 1** In vManage NMS, select 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:

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

2. 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:

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

2. Click the Service VPN drop-down.

3. 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.

**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.
**Configure Basic VPN Parameters**

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

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

**Note**

To complete the configuration of the transport VPN on a vEdge 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, select the Basic Configuration tab and configure the following parameters:

**Note**

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

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>IPv4 or IPv6</th>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shutdown*</td>
<td></td>
<td>Click No to enable the interface.</td>
<td></td>
</tr>
<tr>
<td>Parameter Name</td>
<td>IPv4 or IPv6</td>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>Interface name*</td>
<td></td>
<td></td>
<td>Enter a name for the interface. For IOS XE routers, you must: • Spell out the interface names completely (for example, GigabitEthernet0/0/0). • Configure all the router's interfaces, even if you are not using them, so that they are configured in the shutdown state and so that all default values for them are configured.</td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
<td>Enter a description for the interface.</td>
</tr>
<tr>
<td>IPv4 / IPv6</td>
<td></td>
<td></td>
<td>Click <strong>IPv4</strong> to configure an IPv4 VPN interface. Click <strong>IPv6</strong> to configure an IPv6 interface.</td>
</tr>
<tr>
<td>Dynamic</td>
<td></td>
<td></td>
<td>Click <strong>Dynamic</strong> to set the interface as a Dynamic Host Configuration Protocol (DHCP) client, so that the interface receives its IP address from a DHCP server.</td>
</tr>
<tr>
<td>DHCP Distance</td>
<td>Both</td>
<td></td>
<td>Optionally, enter an administrative distance value for routes learned from a DHCP server. Default is 1.</td>
</tr>
<tr>
<td>DHCP Rapid Commit</td>
<td>IPv6</td>
<td></td>
<td>Optionally, configure the DHCP IPv6 local server to support DHCP Rapid Commit, to enable faster client configuration and confirmation in busy environments. Click <strong>On</strong> to enable DHCP rapid commit. Click <strong>Off</strong> to continue using the regular commit process.</td>
</tr>
<tr>
<td>Static</td>
<td></td>
<td></td>
<td>Click <strong>Static</strong> to enter an IP address that doesn't change.</td>
</tr>
<tr>
<td>IPv4 Address</td>
<td>IPv4</td>
<td></td>
<td>Enter a static IPv4 address.</td>
</tr>
<tr>
<td>IPv6 Address</td>
<td>IPv6</td>
<td></td>
<td>Enter a static IPv6 address.</td>
</tr>
<tr>
<td>Secondary IP Address</td>
<td>IPv4</td>
<td></td>
<td>Click <strong>Add</strong> to enter up to four secondary IPv4 addresses for a service-side interface.</td>
</tr>
<tr>
<td>IPv6 Address</td>
<td>IPv6</td>
<td></td>
<td>Click <strong>Add</strong> to enter up to two secondary IPv6 addresses for a service-side interface.</td>
</tr>
<tr>
<td>DHCP Helper</td>
<td>Both</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Block Non-Source IP</td>
<td>Yes / No</td>
<td></td>
<td>Click <strong>Yes</strong> to have the interface forward traffic only if the source IP address of the traffic matches the interface's IP prefix range. Click <strong>No</strong> to allow other traffic.</td>
</tr>
<tr>
<td>Bandwidth Upstream</td>
<td></td>
<td></td>
<td>For vEdge routers and vManage NMSs: For transmitted traffic, set the bandwidth above which to generate notifications. Range: 1 through (232 / 2) – 1 kbps</td>
</tr>
</tbody>
</table>
Create a Tunnel Interface

On vEdge routers, you can configure up to four tunnel interfaces. This means that each vEdge router can have up to four TLOCs. On vSmart controllers and vManage NMSs, 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.

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

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>vEdge Routers Only</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vBond As Stun Server Only</td>
<td>Yes</td>
<td>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 the vEdge router is located behind a NAT.</td>
</tr>
<tr>
<td>Exclude Controller Group List</td>
<td>Yes</td>
<td>Set the vSmart controllers that the tunnel interface is not allowed to connect to. Range: 0 through 100</td>
</tr>
<tr>
<td>vManage Connection Preference</td>
<td>Yes</td>
<td>Set the preference for using a tunnel interface to exchange control traffic with the vManage NMS. Range: 0 through 8 Default: 5</td>
</tr>
<tr>
<td>Port Hop</td>
<td>No</td>
<td>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. vEdge router default: Enabled vManage NMS and vSmart controller default: Disabled</td>
</tr>
<tr>
<td>Low-Bandwidth Link</td>
<td>Yes</td>
<td>Select to characterize the tunnel interface as a low-bandwidth link.</td>
</tr>
<tr>
<td>Allow Service</td>
<td>No</td>
<td>Select On or Off for each service to allow or disallow the service on the interface.</td>
</tr>
</tbody>
</table>

To configure additional tunnel interface parameters, click Advanced Options:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>vEdge Routers Only</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRE</td>
<td>Yes</td>
<td>Use GRE encapsulation on the tunnel interface. By default, GRE is disabled. 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.</td>
</tr>
<tr>
<td>IPsec</td>
<td>Yes</td>
<td>Use IPsec encapsulation on the tunnel interface. By default, IPsec is enabled. 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.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>vEdge Routers Only</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| IPSec Preference | Yes | Specify a preference value for directing traffic to the tunnel. A higher value is preferred over a lower value.  
Range: 0 through 4294967295  
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.  
Range: 1 through 255  
Default: 1 |
| Carrier | No | Select the carrier name or private network identifier to associate with the tunnel.  
Values: carrier1, carrier2, carrier3, carrier4, carrier5, carrier6, carrier7, carrier8, default  
Default: default |
| Bind Loopback Tunnel | Yes | Enter the name of a physical interface to bind to a loopback interface. |
| Last-Resort Circuit (on vEdge routers) | 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.  
Range: 1 through 60 seconds  
Default: 5 seconds |
| Hello Interval | No | Enter the interval between Hello packets sent on a DTLS or TLS WAN transport connection.  
Range: 100 through 10000 milliseconds  
Default: 1000 milliseconds (1 second) |
| 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.  
Range: 12 through 60 seconds  
Default: 12 seconds |

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

**Configure DNS and Static Hostname Mapping**

To configure DNS addresses and static hostname mapping, select the DNS tab and configure the following parameters:
<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary DNS Address</td>
<td>Select either IPv4 or IPv6, and enter the IP address of the primary DNS server in this VPN.</td>
<td></td>
</tr>
<tr>
<td>New DNS Address</td>
<td>Click New DNS Address 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>
<td></td>
</tr>
<tr>
<td>Mark as Optional Row</td>
<td>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. See Create a Template Variables Spreadsheet.</td>
<td></td>
</tr>
<tr>
<td>Hostname</td>
<td>Enter the hostname of the DNS server. The name can be up to 128 characters.</td>
<td></td>
</tr>
<tr>
<td>List of IP Addresses</td>
<td>Enter up to eight IP addresses to associate with the hostname. Separate the entries with commas.</td>
<td></td>
</tr>
</tbody>
</table>

To save the DNS server configuration, click Add.

To save the feature template, click Save.

**CLI Equivalent**

```
vpn vpn-id
dns ip-address {primary | secondary}
host hostname ip ip-address
```

## Configure VPNs Using CLI

For the control plane to establish itself so that the overlay network can function, you must configure interfaces in VPN 0 to carry the control traffic necessary to establish and maintain the overlay network.

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.

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.

### Configure the 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:

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

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**:

```bash
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**:

```bash
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:

```bash
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:

```bash
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.

```conf
vpn 0
    interface ge0/0
      ip address 10.0.0.8/24
      ipv6 address fd00:1234::/16
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 ge0/3
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
capsulation 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
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 VPN 1:

```
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
```
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 as 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-name is the name of a virtual Ethernet interface (ethnumber).
   ```

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 [dhcpp-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 [dhcpp-distance number] [dhcpp-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
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:
   
   ```
   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.

6. 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]
   ```

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 loopbacknumber 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 is 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 loopbacknumber 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
route
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 (VPN) 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:

   1. Configure an IP address for the interface:

      vEdge(config-interface)# vpn 0 interface interface-name ip address prefix/length
2. Enable the interface:
   ```
   vEdge(config-interface)# no shutdown
   ```

3. 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
   ```

4. 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:
   1. Create the VPN and assign it a identifier number. The identifier can be any number except 0 and 512.
      ```
      vEdge(config)# vpn vpn-id
      ```
   2. Add an interface to the VPN:
      ```
      vEdge(config-vpn-number)# interface interface-name ip address ip-address
      ```
   3. Enable the interface:
      ```
      vEdge(config-vpn-number)# no shutdown
      ```


5. Activate the configuration:
   ```
   vEdge(config)# commit
   ```

Here is the full configuration on the vEdge router:

```bash
system host-name vedge
 host-name vedge
 system-ip 1.0.0.2
 domain-id 1
 site-id 20
 vbond 10.2.6.1

![system]

vpn 0 # Create the tunnel interface and allow reachability to vSmart in transport VPN

interface ge 0/0
 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

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 vEdge router. 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-action)# top
   vSmart(config)# policy control-policy drop-unwanted-routes sequence 10 action reject
   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 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:

```bash
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.

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:

```plaintext
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-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:

```plaintext
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.

```plaintext
vedge-1# show interface
```

<table>
<thead>
<tr>
<th>VPN</th>
<th>INTERFACE</th>
<th>IP ADDRESS</th>
<th>ADMIN STATUS</th>
<th>OPER STATUS</th>
<th>ENCAP</th>
<th>TYPE</th>
<th>PORT</th>
<th>MTU</th>
<th>HWADDR</th>
<th>SPEED</th>
<th>MSS</th>
<th>DUPLEX</th>
<th>ADJUST</th>
<th>UPTIME</th>
<th>RX PACKETS</th>
<th>TX PACKETS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>ge0/0</td>
<td>10.1.15.15/24</td>
<td>Up</td>
<td>Up</td>
<td>null</td>
<td>transport</td>
<td>1500</td>
<td>00:0c:29:7d:1e:fe</td>
<td>10</td>
<td>full</td>
<td>0</td>
<td>0:07:38:49</td>
<td>213199</td>
<td>243908</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>ge0/1</td>
<td>10.1.17.15/24</td>
<td>Up</td>
<td>Up</td>
<td>null</td>
<td>service</td>
<td>1500</td>
<td>00:0c:29:7d:1e:08</td>
<td>10</td>
<td>full</td>
<td>0</td>
<td>0:07:38:49</td>
<td>197</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>ge0/2</td>
<td></td>
<td>Down</td>
<td>Down</td>
<td>null</td>
<td>service</td>
<td>1500</td>
<td>00:0c:29:7d:1e:12</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>ge0/3</td>
<td>10.0.20.15/24</td>
<td>Up</td>
<td>Up</td>
<td>null</td>
<td>service</td>
<td>1500</td>
<td>00:0c:29:7d:1e:1c</td>
<td>10</td>
<td>full</td>
<td>0</td>
<td>0:07:38:49</td>
<td>221</td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>ge0/6</td>
<td>57.0.11.15/24</td>
<td>Up</td>
<td>Up</td>
<td>null</td>
<td>service</td>
<td>1500</td>
<td>00:0c:29:7d:1e:3a</td>
<td>10</td>
<td>full</td>
<td>0</td>
<td>0:07:38:49</td>
<td>196</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>ge0/7</td>
<td>10.0.100.15/24</td>
<td>Up</td>
<td>Up</td>
<td>null</td>
<td>service</td>
<td>1500</td>
<td>00:0c:29:7d:1e:44</td>
<td>10</td>
<td>full</td>
<td>0</td>
<td>0:07:44:47</td>
<td>783</td>
<td>497</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>loopback1</td>
<td>172.16.255.25/32</td>
<td>Up</td>
<td>Up</td>
<td>null</td>
<td>transport</td>
<td>1500</td>
<td>00:00:00:00:00:00:00</td>
<td>10</td>
<td>full</td>
<td>0</td>
<td>0:01:00:20</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>system</td>
<td>172.16.255.15/32</td>
<td>Up</td>
<td>Up</td>
<td>null</td>
<td>loopback</td>
<td>1500</td>
<td>00:00:00:00:00:00:00</td>
<td>10</td>
<td>full</td>
<td>0</td>
<td>0:01:38:25</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>ge0/4</td>
<td>10.20.24.15/24</td>
<td>Up</td>
<td>Up</td>
<td>null</td>
<td>service</td>
<td>1500</td>
<td>00:0c:29:7d:1e:26</td>
<td>10</td>
<td>full</td>
<td>0</td>
<td>0:07:38:46</td>
<td>27594</td>
<td>27405</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>ge0/5</td>
<td>56.0.1.15/24</td>
<td>Up</td>
<td>Up</td>
<td>null</td>
<td>service</td>
<td>1500</td>
<td>00:0c:29:7d:1e:38</td>
<td>10</td>
<td>full</td>
<td>0</td>
<td>0:07:38:46</td>
<td>196</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>eth0</td>
<td>10.0.1.15/24</td>
<td>Up</td>
<td>Up</td>
<td>null</td>
<td>service</td>
<td>1500</td>
<td>00:50:56:00:01:05</td>
<td>1000</td>
<td>full</td>
<td>0</td>
<td>0:07:45:55</td>
<td>15053</td>
<td>10333</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To allow vEdge routers at different overlay network sites on the private network to exchange data traffic directly, you use a loopback interface on 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.
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:

- Loopback 1 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:

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

### 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.

---

**Segmentation CLI Reference**

CLI commands for configuring and monitoring segmentation (VPNs).

**Segmentation Configuration Commands**

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

```
vpn vpn-id
  bandwidth-downstream kbps (on vEdge routers and vManage NMSs only)
  bandwidth-upstream kbps (on vEdge routers and vManage NMSs only)
  dns ip-address [primary | secondary]
  ecmp-hash-key layer4 (on vEdge routers only) host hostname ip ip-address
  interface interface-name
    access-list acl-list (on vEdge routers only)
    arp
    ip ip-address mac mac-address
    arp-timeout seconds (on vEdge routers only)
    autonegotiate (on vEdge routers only)
    block-non-source-ip (on vEdge routers only)
    clear-dont-fragment
    dead-peer-detection interval seconds retries number (on vEdge routers only)
    description text
    dhcp-helper ip-address (on vEdge routers only)
    dhcp-server (on vEdge routers only)
      address-pool prefix/length
      exclude ip-address
      lease-time seconds
      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 ip-address host-name hostname
dot1x
  accounting-interval seconds
  acct-req-attr attribute-number (integer integer | octet octet | string string)
  auth-fail-vlan vlan-id
  auth-order (mab | radius)
  auth-reject-vlan vlan-id
```
auth-req-attr attribute-number {integer integer | octet octet | string string}
control-direction direction
das
  client ip-address
  port port-number
  require-timestamp
  secret-key password
  time-window seconds
  vpn vpn-id
default-vlan vlan-id
guest-vlan vlan-id
host-mode {multi-auth | multi-host | single-host}
mac-authentication-bypass
  allow mac-addresses
  server
  nas-identifier string
  nas-ip-address ip-address
  radius-servers tag
  reauthentication minutes
  timeout
  inactivity minutes
  wake-on-lan
dead-peer-detection interval time-units retries number (on vEdge routers only)
duplex {full | half}
flow-control {bidirectional | egress | ingress}
ike (on vEdge routers only)
  authentication-type type
  local-id id
  pre-shared-secret password
  remote-id id
  cipher-suite suite
  group number
  mode mode
  rekey seconds
  version number
  (ip address ipv4-refix/length | ip dhcp-client [dhcp-distance number])
  (ipv6 address ipv6-refix/length | ipv6 dhcp-client [dhcp-distance number][dhcp-rapid-commit])
ip address-list prefix/length (on vSmart controller containers only)
ipsec (on vEdge routers only)
  cipher-suite suite
  perfect-forward-secrecy pfs-setting
  rekey seconds
  replay-window number
  keepalive seconds retries (on vEdge routers only)
mac-address mac-address
mtu bytes
nat (on vEdge routers only)
  block-icmp-error
  direction {inside | outside}
  log-translations
  [no] overload
  port-forward port-start port-number1 port-end port-number2
  proto {tcp | udp} private-ip-address ip address private-vpn vpn-id
  refresh {bi-directional | outbound}
  respond-to-ping
  static source-ip ip-address1 translate-ip ip-address2 (inside | outside)
  static source-ip ip-address1 translate-ip ip-address2 source-vpn vpn-id protocol
  (tcp | udp)
  source-port number translate-port number
tcp-timeout minutes
  udp-timeout minutes
  pmtu (on vEdge routers only)
policer policer-name (on vEdge routers only)
ppp (on vEdge routers only)
   ac-name name
   authentication (chap | pap) hostname name password password
pppoe-client (on vEdge routers only)
   ppp-interface name
profile profile-id (on vEdge routers only)
gos-map name (on vEdge routers only)
rewrite-rule name (on vEdge routers only)
secondary-address ipv4-address (on vEdge routers only)
shaping-rate name (on vEdge routers only)
[no] shutdown
speed speed
static-ingress-qos number (on vEdge routers only)
tcp-mas-adjust bytes
technology technology (on vEdge routers only)
tloc-extension interface-name (on vEdge routers only)
tunnel-interface
   allow-service service-name
   bind geslot/port (on vEdge routers only)
carrier carrier-name
color color [restrict]
connections-limit number encapsulation (gre | ipsec) (on vEdge routers only)
   preference number
   weight number
hello-interval milliseconds
hello-tolerance seconds
last-resort-circuit (on vEdge routers only)
low-bandwidth-link (on vEdge routers only)
max-control-connections number (on vEdge routers only)
nat-refresh-interval seconds
port-hop
vbond-as-stun-server (on vEdge routers only)
vmanage-connection-preference number (on vEdge routers only)
tunnel-destination (on vEdge routers only)
tunnel-destination (dns-name | ipv4-address) (IPsec interfaces; on vEdge routers only)
   (tunnel-source ip-address | tunnel-source-interface interface-name) (GRE interfaces; on vEdge routers only)
   (on vEdge routers only)
   upgrade-confirm minutes
   vrrp group-name (on vEdge routers only)
   priority number
timer seconds
track-omp
! end vpn interface
ip route ip-address/subnet next-hop-address
name text
omp
   advertise (aggregate prefix [aggregate-only] | bgp | connected | network prefix | ospf
type | static) (on vEdge routers only)
router (on vEdge routers only)
   bgp ...
   igmp ...
multicast-replicator local
   threshold number
   ospf ...
   pim ...
service service-name address ip-address (on vEdge routers only)

Segmentation Monitoring Commands

Use the following commands to monitor segmentation:
• show bgp commands
• show interface commands
• show ospf commands
Forwarding and QoS

Forwarding is the transmitting of data packets from one vEdge 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 vEdge routers.

- Cisco SD-WAN Forwarding and QoS Overview, on page 123
- Traffic Behavior With and Without QoS, on page 124
- How QoS Works, on page 125
- Forwarding and QoS Configuration Examples, on page 126
- Reference: Forwarding and QoS CLI Commands, on page 132

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 vEdge 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 vEdge routers. Because data traffic never goes to or through the centralized vSmart controller, forwarding only occurs between the vEdge routers as they send and receive data traffic.

While the routing protocols running in the control plane provide a vEdge router the best route to reach the network that is on the service side of a remote vEdge 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 vEdge router.

To modify the default data packet forwarding flow, you create and apply centralized data policy or localized data policy. With 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 localized data policy, you can control the flow of data traffic into and out of a vEdge router's interfaces, 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 vEdge router, and then to the remote vEdge router and the remote service-side network, with no alterations in its path. When no access lists are configured on the local vEdge 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 vEdge-1 router. The packet has a source IP address and a destination IP address.
- The vEdge router looks up the outbound SA in its VPN route table, and the packet is encrypted with SA and gets the local TLOC. (The vEdge 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 vEdge 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 vEdge 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 vEdge router looks up the destination IP address in its route table to determine the interface to use to reach to the service-side destination.

The figure below details this process.

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 vEdge 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 to the vEdge router, and the policies marked Output are applied on the outbound interface to the vEdge router, before the packets are transmitted out the IPSec tunnel.

The table below describes each of the above steps.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Define class map to classify packets, by importance, into appropriate forwarding classes. Reference the class map in an access list.</td>
<td>class-map</td>
</tr>
<tr>
<td>2</td>
<td>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.</td>
<td>policer</td>
</tr>
<tr>
<td>3</td>
<td>vEdge router checks the local route table to determine which interface the packet should use to reach its destination.</td>
<td>N/A</td>
</tr>
<tr>
<td>4</td>
<td>Define policer and reference the policer in an access list. Apply the access list on an outbound interface.</td>
<td>policer</td>
</tr>
<tr>
<td>5</td>
<td>Define QoS map to define the priority of data packets. Apply the QoS map on the outbound interface.</td>
<td>qos-map</td>
</tr>
<tr>
<td>6</td>
<td>Define rewrite-rule to overwrite the DSCP field of the outer IP header. Apply the rewrite-rule on the outbound interface.</td>
<td>rewrite-rule</td>
</tr>
</tbody>
</table>

**How QoS Works**

The QoS feature on the vEdge routers 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 vEdge router.)
To provision QoS, you must configure each vEdge 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 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 vEdge 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 hardware vEdge routers and Cloud vEdge virtualized routers, 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. All control traffic is transmitted. 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.

**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 vEdge routers 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.

---

**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 in to the interfaces on a vEdge router and 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.
5. Apply the access list to a specific interface.
6. Apply the queue map and the rewrite rule to the egress interface.

The sections below show examples of each of these steps.

**Map 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.

```plaintext
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.

```plaintext
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
```
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
```

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 20.20.20.1/32
      !
      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
```
### 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 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 cannot apply QOS maps to VLAN interfaces, also called subinterfaces.) 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
```
Police Data Packs

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
!
```

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
  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:
In the second example, we have a vEdge router 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 belows if 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 vEdge router:

```plaintext
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 vEdge router. The count is reported in the `rx-policer-remark` field.
The centralized data policy directs bursty 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 \textit{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:

\begin{verbatim}
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
\end{verbatim}

**Reference: Forwarding and QoS CLI Commands**

**Configuration Commands**

Use the following commands to configure forwarding and QoS on a vEdge router.

\begin{verbatim}
policy
    class-map
        class class-name queue number
    cloud-qos
    cloud-qos-service-side
    mirror mirror-name
        remote-dest ip-address source ip-address
    policer policer-name
        rate bandwidth
        burst types
        exceed action
    qos-map map-name
        qos-scheduler scheduler-name
        qos-scheduler scheduler-name
            class class-name
            bandwidth-percent percentage
            buffer-percent percentage
            drops (red-drop | tail-drop)
            scheduling (llq | wrr)
            rewrite-rule rule-name

policy
    access-list acl-name
        default-action action
        sequence number
        match
            class class-name
            destination-ip prefix/length
            destination-port number
            dscp number
\end{verbatim}
Monitoring Commands

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

```
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
```
Protocols in Cisco SD-WAN

This chapter discusses the protocols supported in Cisco SD-WAN.

- BFD, on page 135
- EIGRP, on page 138
- Other Supported Protocols, on page 143

BFD

Use the BFD template for vEdge routers and Cisco IOS XE routers.

The BFD protocol, which detects link failures as part of the Cisco SD-WAN high availability solution, is enabled by default on all vEdge routers, and you cannot disable it.

Navigate to the Template Screen

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. To create a custom template for BFD, select the Factory_Default_BFD_Template and click Create Template. The BFD template form is displayed. The top of the form contains fields for naming the template, and the bottom contains fields for defining BFD 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 31:

<table>
<thead>
<tr>
<th>Parameter Scope</th>
<th>Scope Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Specific (indicated by a host icon)</td>
<td>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. 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. To change the default key, type a new string and move the cursor out of the Enter Key box. Examples of device-specific parameters are system IP address, hostname, GPS location, and site ID.</td>
</tr>
<tr>
<td>Global (indicated by a globe icon)</td>
<td>Enter a value for the parameter, and apply that value to all devices. Examples of parameters that you might apply globally to a group of devices are DNS server, syslog server, and interface MTUs.</td>
</tr>
</tbody>
</table>

Configure BFD for Application-Aware Routing

To configure the BFD timers used by application-aware routing, click the Basic Configuration tab and configure the following parameters:

Table 32:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplier</td>
<td>Specify the value by which to multiply the poll interval, to set how often application-aware routing acts on the data plane tunnel statistics to figure out the loss and latency and to calculate new tunnels if the loss and latency times do not meet configured SLAs. Range: 1 through 6 Default: 6</td>
</tr>
<tr>
<td>Poll Interval</td>
<td>Specify how often BFD polls all data plane tunnels on a vEdge router to collect packet latency, loss, and other statistics used by application-aware routing. Range: 1 through 4,294,967,296 (2^32 – 1) milliseconds Default: 600,000 milliseconds (10 minutes)</td>
</tr>
</tbody>
</table>

To save the feature template, click Save.

CLI equivalent:

```
bfd app-route
   multiplier number
   poll-interval milliseconds
```
Configure BFD on Transport Tunnels

To configure the BFD timers used on transport tunnels, click the Color tab. Next, click Add New Color, and configure the following parameters:

Table 33:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color</td>
<td>From the drop-down, choose the color of the transport tunnel for data traffic moving between vEdge routers. The color identifies a specific WAN transport provider. <em>Values:</em> 3g, biz-internet, blue, bronze, custom1, custom2, custom3, default, gold, green, lte, metro-ethernet, mpls, private1 through private6, public-internet, red, silver <em>Default:</em> default</td>
</tr>
<tr>
<td>Hello Interval</td>
<td>Specify how often BFD sends Hello packets on the transport tunnel. BFD uses these packets to detect the liveness of the tunnel connection and to detect faults on the tunnel. <em>Range:</em> 100 through 60000 milliseconds <em>Default:</em> 1000 milliseconds (1 second)</td>
</tr>
<tr>
<td>Multiplier</td>
<td>Specify how many Hello packet intervals BFD waits before declaring that a tunnel has failed. BFD declares that the tunnel has failed when, during all these intervals, BFD has received no Hello packets on the tunnel. This interval is a multiplier of the Hello packet interval time. <em>Range:</em> 1 through 60 <em>Default:</em> 7 (for hardware vEdge routers), 20 (for vEdge Cloud software routers)</td>
</tr>
<tr>
<td>Path MTU</td>
<td>Click On to enable path MTU discovery for the transport tunnel, or Off to disable. When PMTU discovery is enabled, the path MTU for the tunnel connection is checked periodically, about once per minute, and it is updated dynamically. When PMTU discovery is disabled, the expected tunnel MTU is 1472 bytes, but the effective tunnel MTU is 1468 bytes. <em>Default:</em> Enabled</td>
</tr>
</tbody>
</table>

Add

Click Add to save the data traffic transport tunnel color.

To add another color, click Add New Color.

A table lists the transport tunnel colors.

To edit a color, click the Pencil icon. The Update Color popup is displayed. After you make the desired changes, click Save Changes.

To remove a color, click the trash icon to the right of the entry.

To save the feature template, click Save.

CLI equivalent:

```bash
bfd color color
  hello-interval milliseconds
  multiplier number
  pmtu-discovery
```

EIGRP

Cisco release 19.1 supports Enhanced Interior Gateway Routing Protocol (EIGRP) on Cisco IOS XE devices. EIGRP is an open standard IGP routing protocol that provides advantages such as:
• Increased network width from 15 to 100 hops
• Fast convergence
• Incremental updates, minimizing bandwidth
• Protocol-independent neighbor discovery
• Easy scaling

If your EIGRP network includes vEdge routers, you may need additional software. Refer to SD-WAN 19.1 release notes for configuration information.

To configure EIGRP routing protocol using vManage templates:

1. Create an EIGRP feature template to configure EIGRP parameters, described here.
2. Create a VPN feature template to configure VPN parameters for service-side routing (any VPN other than VPN 0 or VPN 512). See VPN.
3. Create a device template and apply the templates to the correct devices. See Templates.

Create an EIGRP Template

Step 1  From the vManage menu, select Configuration > Templates.
Step 2  Click Feature.
Step 3  Click Add Template.
Step 4  Select a Cisco IOS XE device from the list.
Step 5  From the Other Templates section, click EIGRP.

The EIGRP Feature template opens. The top of the form contains fields for naming the template, and the bottom contains fields for defining EIGRP parameters.
Step 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.

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

---

**Basic Configuration**

Click the **Basic Configuration** tab to configure the local autonomous system (AS) number for the template.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Autonomous System ID</strong>*</td>
<td>Enter the local AS number.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Range:</strong> 1-65,535</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default:</strong> None</td>
</tr>
</tbody>
</table>

**Equivalent CLI Commands**

```            
vpn vpn-id     
router       
eigrp name    
address-family ipv4 vrf vrf-name autonomous-system number
```

**IP4 Unicast Address Family**

To configure the global EIGRP address family, click the **Unicast Address Family** tab.
Redistribute Tab

To redistribute routes from one protocol (routing domain) into an EIGRP routing domain, click New Redistribute and enter the following parameter values:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mark as Optional Row</td>
<td>Click Optional 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. See Create a Template Variables Spreadsheet.</td>
<td></td>
</tr>
<tr>
<td>Protocol *</td>
<td>Select the protocols from which to redistribute routes into EIGRP, for all EIGRP sessions.</td>
<td>bgp</td>
</tr>
<tr>
<td>Route Policy *</td>
<td>Enter the name of the route policy to apply to redistributed routes.</td>
<td></td>
</tr>
</tbody>
</table>

Click Add to save the redistribution information.

Network Tab

To advertise a prefix into the EIGRP routing domain, click the Network tab, and then click New Network and enter the following parameter values:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mark as Optional Row</td>
<td>Click Optional 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. See Create a Template Variables Spreadsheet.</td>
</tr>
<tr>
<td>Network Prefix *</td>
<td>Enter the network prefix you want EIGRP to advertise in the format of prefix/mask.</td>
</tr>
</tbody>
</table>

Click Add to save the network prefix.

Equivalent CLI Commands

```
vpn  vpn-id
router
    eigrp
        address-family ipv4-unicast
```
Advanced Parameters

To configure advanced parameters for EIGRP, click the Advanced tab and configure the following parameter values:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hold Time (seconds)</td>
<td>Set the interval after which EIGRP considers a neighbor to be down. The local router then terminates the EIGRP session to that peer. This acts as the global hold time.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Range</strong>: 0 through 65,535</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default</strong>: 15 seconds</td>
</tr>
<tr>
<td>Hello Interval (seconds)</td>
<td>Set the interval at which the router sends EIGRP hello packets.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Range</strong>: 0 through 65,535</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default</strong>: 5 seconds</td>
</tr>
<tr>
<td>Route Policy Name</td>
<td>Enter the name of an EIGRP route policy.</td>
</tr>
</tbody>
</table>

Equivalent CLI Commands

```
vpn  vpn-id
router
eigrp name
  address-family ipv4 vrf vrf-name autonomous-system number
  af-interface intf-name
  hello-interval seconds
  hold-time seconds
```

Route Authentication Parameters

The IP Enhanced IGRP Route Authentication feature supports MD5 or HMAC-sha-256 authentication of routing updates from the EIGRP routing protocol. To configure authentication for EIGRP routes:

1. Click the Authentication tab.
2. Click Authentication to open the Authentication Type field.
3. Select global parameter scope.
4. From the drop-down list, select md5 or hmac-sha-256.
**Description**

Enter an MD5 key ID to compute an MD5 hash over the contents of the EIGRP packet using that value.

**Option**

MD5 Key ID

**Description**

Enter an MD5 key ID to compute an MD5 hash over the contents of the EIGRP packet using that value.

**Parameter**

MD5 Authentication Key

**Description**

Enter an MD5 authentication key to use an encoded MD5 checksum in the transmitted packet.

**Parameter**

Authentication Key

**Description**

A 256-byte unique piece of information that is used to compute the HMAC and is known both by the sender and the receiver of the message.

Click **Add** to save the authentication parameters.

---

**Note**

To use a preferred route map, specify both an MD5 key (ID or auth key) and a route map.

---

**Equivalent CLI Commands**

```
vpn vpn-id
router
eigrp name
  address-family ipv4 vrf vrf-name autonomous-system number
  af-interface intf-name
  authentication key-chain keychain-name
  authentication mode {hmac-sha-256 | md5}
```

---

**Interface Parameters**

To configure interface parameters for EIGRP routes, click **Interface**, and enter the following parameter values:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mark as Optional Row</td>
<td>Click <strong>Optional</strong> 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>Interface name</td>
<td>Enter the interface name(s) on which EIGRP should run.</td>
</tr>
<tr>
<td>Shutdown</td>
<td>No (the default) enables the interface to run EIGRP. Yes disables the interface.</td>
</tr>
</tbody>
</table>

Click **Add** to save the interfaces.

---

**Other Supported Protocols**

This topic lists all the other protocols supported in Cisco SD-WAN.

- DHCP Server: See the System and Interfaces guide for more information.
• BGP, OSPF, OMP: See the Unicast Overlay Routing chapter in this guide for more information.
• PIM, IGMP: See the Multicast Overlay Routing chapter in this guide for more information.
QoS on Subinterface

A physical interface may be treated as multiple interfaces by configuring one or more logical interfaces called subinterfaces. One use case is separating the traffic of different VLANs by using a separate subinterface for each VLAN.

Quality of Service (QoS) policies may be applied to individual subinterfaces. Configure QoS as usual, specifying the interface and subinterface using the interface.subinterface notation. For example, for GigabitEthernet interface 4, subinterface 100:

GigabitEthemet4.100

- Limitations, on page 145
- Configuration Example: QoS on Subinterface, on page 145

Limitations

- Do not configure a QoS policy on both a main interface and one of its subinterfaces. The exception is a class-default shape policy on the main interface.
- A QoS policy that is applied to a subinterface must have shaping defined. This configured with the shape command. Example:

```
policy-map shape_GigabitEthernet4.100
   class class-default
      service-policy xyz_QoS-model
      shape average 100000000
```

Configuration Example: QoS on Subinterface

This example applies a QoS policy to subinterface GigabitEthernet4.100 (shown in red in the figure below). This subinterface handles traffic for VLAN 100. The QoS policy affects only subinterface GigabitEthernet4.100, and not subinterface GigabitEthernet4.200, which is on the same physical interface.
Configuration by CLI

class-map match-any DATA
  match qos-group 1
class-map match-any Queue0
  match qos-group 0
class-map match-any Queue1
  match qos-group 1
class-map match-any Queue2
  match qos-group 2
class-map match-any Queue7
  match qos-group 7
class-map match-any WEB
  match qos-group 7

policy-map xyz_QoS-model
  class Queue0
    priority percent 37
  class Queue1
    bandwidth percent 33
  class Queue7
    random-detect
    bandwidth percent 10
  class class-default
    random-detect
    bandwidth percent 20

policy-map shape_GigabitEthernet4.100
  class class-default
    service-policy xyz_QoS-model
    shape average 100000000

interface GigabitEthernet4.100
  no shutdown
  encapsulation dot1Q 100
  ip address 173.10.0.2 255.255.255.0
  ip mtu 1496
  service-policy output shape_GigabitEthernet4.100
  exit

  exit
interface Tunnel3
  no shutdown
  ip unnumbered GigabitEthernet4.100
  tunnel source GigabitEthernet4.100
  tunnel mode sdwan
exit

sdwan
interface GigabitEthernet4.100
tunnel-interface	encapsulation ipsec
color private3 restrict
max-control-connections 0

policy
class-map
class Queue0 queue 0
class VOICE queue 0
class DATA queue 1
class Queue1 queue 1
class Queue2 queue 2
class Queue7 queue 7
class WEB queue 7
!

Configuration by vManage

To apply a QoS policy to a subinterface using vManage, the procedure is similar to that used for configuring policies on a main interface. Add a subinterface feature template to the device template for the target device. This enables loading the QoS policy onto the subinterface.

Preparation

• Configure a QoS Policy
  Configuration > Policies > Localized Policy > Custom Options > Forwarding Class/QoS

• Apply a QoS Policy to a Subinterface
  Apply a QoS policy and define shaping.
  1. Configuration > Feature > feature-name > ACL/QoS
  2. Configure the following fields:
     • Shaping Rate (Kbps)
     • QoS Map

Procedure

This procedure applies a QoS policy to a subinterface.
Prerequisite: One or more class maps have been defined. These assign classes of traffic (for example, VoIP traffic) to specific queues.
1. Create a QoS policy map.
   1. Configuration > Policies
   2. Click Localized Policy at the top.
   3. Click the Add Policy button to create a new policy map.
   4. Click Next.
   5. Click the Add QoS Map button and select Create New from the dropdown menu.
   6. (This step relies on class maps that have been defined. The class maps assign classes of traffic to specific queues. The queues then represent those classes of traffic. This step uses the queues to control how the traffic will be handled.)
      In the Add Queue dialog box, select queues that represent the types of traffic relevant to the QoS objectives. Configure parameters such as Bandwidth% and Buffer% for the queues. For example, to configure bandwidth for audio traffic, select a queue that represents audio traffic and configure the bandwidth parameter. Click the Save Queue button.
   7. Click the Save Policy button.

2. Create a QoS policy that uses the QoS policy map defined above.
   See the documentation for creating a QoS policy.

3. Use a device template to push the QoS policy to the target device.
   (Note: The device policy defines other parts of the device configuration also. This procedure only affects the QoS policy portion.)
QoS on Subinterface

Configuration by vManage

1. Configuration > Templates
2. In the list of templates, locate the device template for the target device.
3. In table row for that template, click the ... button at the right, and select Edit.
4. In the Additional Templates area, in the Policy field, click the dropdown menu and select the policy name.
5. Click Update.
6. Click Next.
7. In the left pane, select the target device. The configuration appears in the right pane.
8. Click the Configure Devices button to push the policy to the device. SD-WAN displays the Task View, showing the status of the update tasks.

4. Load the QoS policy onto the subinterface.

Prerequisite: The subinterface feature template must already have been added to the device template.

1. Configuration > Templates
2. Click Feature at the top.
3. In the list of templates, locate the feature template for the subinterface. (This is the subinterface to which you are assigning the QoS policy.)
4. In the Device Templates column, confirm that the feature template is assigned to a device template.
5. In the Devices Attached column, confirm that the feature template is assigned to a device.
6. In table row for the template, click the ... button at the right, and select Edit.
7. Click ACL/QoS to jump to the ACL/QoS section.
8. In the Shaping Rate field, use the dropdown menu to select Global or Device Specific, and enter a shaping rate value.
9. In the QoS Map field, use the dropdown menu to select Global and enter the QoS policy map name.
10. Click Update.
11. In the left pane, select the device to display the configuration in the right pane.
12. Click the Configure Devices button to push the policy map to the subinterface. SD-WAN displays the Task View, showing the status of the update tasks.