Cisco Catalyst SD-WAN Network Configuration Guide, Releases 26.x and Later

First Published: 2026-04-24

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

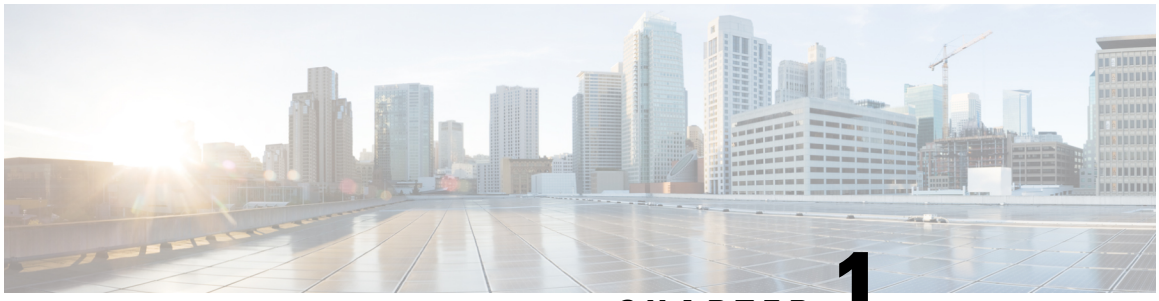
All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2026 Cisco Systems, Inc. All rights reserved.

CONTENTS



CHAPTER 1

Read Me First

Note To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**, and **Cisco Controllers** to **Cisco Catalyst SD-WAN Control Components**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

Related References

- Cisco Catalyst SD-WAN Control Components Compatibility Matrix and Server Recommendations
- Cisco Catalyst SD-WAN Device Compatibility

User Documentation

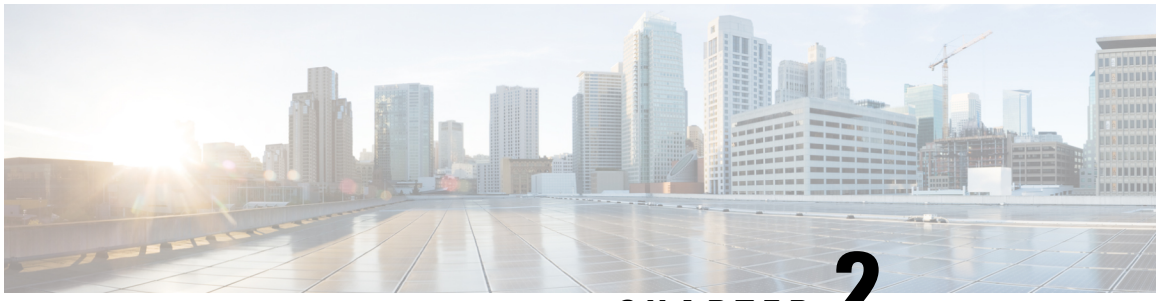
- User Documentation for Cisco IOS XE Catalyst SD-WAN Release 17

Communications, Services, and Additional Information

- Sign up for Cisco email newsletters and other communications at: Cisco Profile Manager.
- For information on the latest technical, advanced, and remote services to increase the operational reliability of your network visit Cisco Services.
- To browse and discover secure, validated enterprise-class apps, products, solutions, and services, visit Cisco Devnet.
- To obtain general networking, training, and certification titles from Cisco Press Publishers, visit Cisco Press.
- To find warranty information for a specific product or product family, visit Cisco Warranty Finder.
- To view open and resolved bugs for a release, access the Cisco Bug Search Tool.
- To submit a service request, visit Cisco Support.

Documentation Feedback

To provide feedback about Cisco technical documentation use the feedback form available in the right pane of every online document.



CHAPTER 2

Bandwidth Utilization Reference Values

Feature history for bandwidth utilization Reference Values

Table 1: Feature History

Feature Name	Release Information	Description
Upstream and Downstream Bandwidth Reference Values	Cisco IOS XE Catalyst SD-WAN Release 17.16.1a Cisco Catalyst SD-WAN Manager Release 20.16.1	Use the upstream and downstream bandwidth reference values to govern how Cisco SD-WAN Manager displays interface utilization percentages in charts. The values also act as configurable thresholds that trigger interface-bw events when a network interface's utilization exceeds a defined point.

Bandwidth Utilization Reference Values

A bandwidth utilization reference value is a configurable parameter that

- specifies the upstream (egress) bandwidth and downstream (ingress) bandwidth for each interface, and
- acts as a reference for calculating bandwidth utilization and generating notifications.

You can configure bandwidth utilization reference values for each interface. Cisco SD-WAN Manager uses these reference values to display bandwidth utilization percentages in charts.

Generating Notifications

To receive notifications when the traffic bandwidth exceeds 85% utilization, configure reference values for both upstream (transmitted) and downstream (received) traffic. These values act as thresholds for generating the interface bandwidth events. The range is from 1 and 2,147,483,647 kbps.

The device samples the interface traffic each 10 seconds. If the received or transmitted bandwidth exceeds 85 percent of the configured value in 85 percent of the sampled intervals in a continuous 5-minute period, the device generates an SNMP trap. After the first trap is generated, sampling continues at the same frequency, but notifications are rate-limited to once per hour. A second trap is sent, and subsequent traps are sent, if the bandwidth exceeds 85 percent of the value in 85 percent of the 10-second sampling intervals over the next 1-hour period. If, after 1 hour, the device does not send another SNMP trap, the notification interval reverts to 5 minutes.

The upstream bandwidth and downstream bandwidth settings are solely for monitoring purposes and do not impose a bandwidth limit on the traffic. For example, in some network configurations, the full bandwidth of an interface may not be available. To ensure that utilization values reflect the available network bandwidth, set bandwidth utilization reference values lower than an interface's full speed.

You can verify the configured upstream and downstream bandwidth values using the **show interface detail** command, which displays the upstream bandwidth (tx-kbps) and downstream bandwidth (rx-kbps) fields usage.

You can monitor transport circuit bandwidth on Cisco IOS XE Catalyst SD-WAN devices and on Cisco SD-WAN Manager.

Monitoring Bandwidth Utilization with Interface Charts

Cisco SD-WAN Manager provides a chart showing bandwidth utilization for each interface of a device. To view the chart, from the Cisco SD-WAN Manager menu, choose **Monitor** > **Devices**, click a device and click **Interface**.

Configuring upstream or downstream reference values affects how the chart displays the percentages:

- No bandwidth utilization reference values configured: By default, devices calculate the bandwidth utilization value according to the interface speed of the connection.
- Bandwidth utilization reference values configured: If you configure bandwidth utilization reference values, devices calculate the bandwidth utilization value as a percentage of the reference point.

  For example, if you configure upstream bandwidth and downstream bandwidth values of 500 megabits per second, and if the downstream utilization is 500 megabits per second, the device reports downstream utilization as 100%.

Devices limit the calculated utilization value to 100% even if the traffic utilization is more than 100% of the configured reference value.

Restrictions for bandwidth utilization reference values

- Cisco SD-WAN Manager supports bandwidth data rate and utilization statistics only for a primary network interface, not a subinterface or loopback interface. You can still configure upstream and downstream bandwidth under a subinterface or loopback interface as a reference value or for per-tunnel QoS functionality.
- Configure upstream and downstream bandwidth references for service-side VPN interfaces using only a CLI template.

Configure bandwidth utilization reference values

Use one of these methods to configure bandwidth utilization reference values:

- Configuration group
- CLI commands
- Feature template

Configure bandwidth utilization reference values using a configuration group

Follow these steps to configure bandwidth utilization reference values using a configuration group

Before you begin

Minimum supported releases: Cisco IOS XE Catalyst SD-WAN Release 17.16.1a and Cisco Catalyst SD-WAN Manager Release 20.16.1

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration Groups**.

Step 2 Under Transport & Management Profile, click **...** adjacent to the transport profile, and choose **Edit**.

Step 3 Click the edit icon adjacent to Ethernet Interface.

Step 4 In the **Basic Configuration** section, enter the upstream and downstream bandwidth reference values.

Step 5 Click **Save**.

Configure bandwidth utilization reference values using CLI commands

This section provides example CLI configurations to configure upstream and downstream bandwidth reference values in Cisco SD-WAN Manager. For more information about using CLI templates, see *CLI Add-On Feature Templates* and *CLI Templates*.

By default, CLI templates execute commands in global config mode.

Procedure

Step 1 Enter the SD-WAN configuration mode.

Example:

```
sdwan
```

Step 2 Enter interface configuration mode.

Example:

```
interface interface-name
```

Step 3 Define the upstream or downstream bandwidth reference value in kbps.

Example:

```
bandwidth-upstream upstream-value
bandwidth-downstream downstream-value
```

Here is a complete configuration example for configuring upstream and downstream bandwidth reference values:

```
sdwan
!
interface GigabitEthernet1
bandwidth-upstream 10000
bandwidth-downstream 10000
```

Configure bandwidth utilization reference values using templates

Follow these steps to configure bandwidth utilization reference values using feature templates.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.

Step 2 Click **Feature Templates**.

Step 3 Do one of these:

- Click **Add Template**, choose a device, and create a Cisco VPN Interface Ethernet template.
- Edit an existing Cisco VPN Interface Ethernet template.

Step 4 In the **Basic Configuration** section, enter the upstream and downstream bandwidth reference values.

Step 5 Click **Update** to save the template.

Monitor Upstream and Downstream Bandwidth Reference Values

Monitor bandwidth utilization events

Follow these steps to monitor upstream and downstream bandwidth events using the Cisco SD-WAN Manager.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Monitor** > **Logs** > **Events**.

Step 2 View the details of upstream and downstream bandwidth values in the device configuration section. The default name for interface bandwidth-related events is **interface-bw**.

Monitor bandwidth utilization reference values

Follow these steps to monitor bandwidth utilization reference values using Cisco SD-WAN Manager.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Monitor** > **Devices**.

Step 2 Click the **Device** tab.

Step 3 Click the hostname of the device you want to monitor.

Step 4 From the **Applications** section, choose **Interface**.

Step 5 Click the **Chart Options** drop-down list and choose **Utilization**.

Step 6 Hover over the real-time or historical data points to view the utilization percentage.

Verify bandwidth utilization reference values

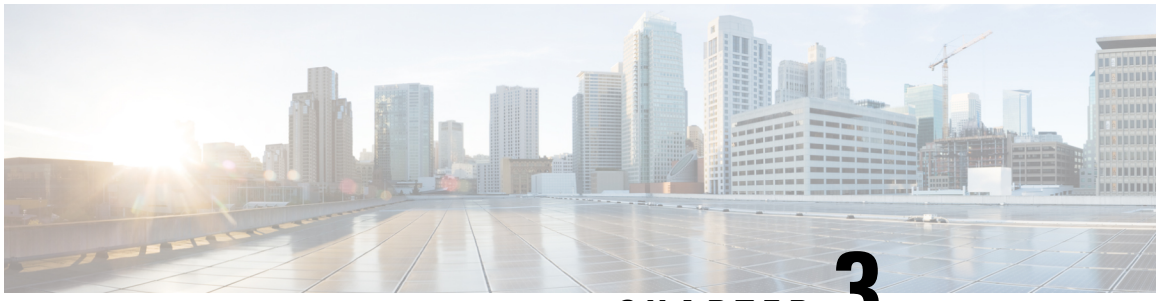
Monitor interface alarms

Use the **show platform software sdwan interface-alarm summary** command to display the bandwidth reference values for an interface in the alarm summary. This sample output shows the summary details of a GigabitEthernet interface alarm.

See Interface Alarm Summary, and Base and High Intervals for more information.

```
device# show platform software sdwan interface-alarm summary
```

```
============ Interface Alarm Summary ============
Interface Name      Upstream (kbps)  Downstream (kbps)   Base Interval(s)   High Interval(s)
GigabitEthernet1      10000               10000              300                3600
```



CHAPTER 3

Carrier Supporting Carrier

Feature history for carrier supporting carrier

Table 2: Feature history

Feature name	Release information	Description
Cisco Catalyst SD-WAN Support for Carrier Supporting Carrier Connectivity	Cisco IOS XE Catalyst SD-WAN Release 17.6.1a Cisco vManage Release 20.6.1	The feature adds support for carrier supporting carrier (CSC) connectivity on Cisco IOS XE Catalyst SD-WAN devices. CSC enables you to interconnect IP or multiprotocol label switching (MPLS) networks operating at different sites over an MPLS backbone network. Using CSC requires an edge router that supports CSC functionality, called a carrier edge (CE) device, at each site. This feature enables a Cisco IOS XE Catalyst SD-WAN device to serve as a CE device, making it unnecessary to have a separate dedicated CE device at each site managed by Cisco Catalyst SD-WAN.

Carrier supporting carrier

A carrier supporting carrier (CSC) is a hierarchical VPN model that

- allows organizations to interconnect their IP or MPLS networks located at different sites,
- operates over an MPLS backbone network, and
- eliminates the need for organizations to build and maintain their own MPLS backbone.

Components of carrier supporting carrier

- Backbone carrier: The service provider operates the backbone network. Typically, the backbone carrier network employs multiple segments to segregate the traffic of different customer carriers that share it. The same organization as the customer carriers or a different organization manages the backbone carrier.
- Customer carrier: An organization that uses the backbone network to route traffic from one site to another. The customer carrier may be part of the organization that operates the backbone network, or may be independent.
- CSC-CE: The customer edge (CE) device operates within a local site network and connects the site to the backbone carrier using an MPLS connection. It uses the backbone carrier to connect to other sites.
- CSC-PE: The provider edge (PE) device operates within the backbone carrier network and connects to CSC-CE devices at customer sites using an MPLS connection.

Benefits of carrier supporting carrier

A Cisco IOS XE Catalyst SD-WAN device functions as an customer edge device at a site requiring CSC, eliminating the need for a separate CE router.

Traffic flow configuration between CSC-CE and CSC-PE devices

The traffic flow configuration between CSC-CE and CSC-PE devices determines how service VPN, control, and BFD probe traffic is routed based on available MPLS and internet connections. This configuration optimizes network resources, ensures efficient traffic handling, and supports high availability in service provider networks.

Traffic flow

When a CSC-CE device has only an MPLS connection to its neighboring CSC-PE device, it sends all traffic, including service VPN traffic, control traffic, and Cisco Catalyst SD-WAN bidirectional forwarding detection (BFD) probe traffic, over the MPLS connection.

When a CSC-CE device has both an MPLS connection to the neighboring CSC-PE device and a separate internet connection, it routes traffic as follows:

- Based on the configured traffic policy, it can send control traffic and BFD probe traffic over either the internet or the MPLS connection.
- It sends service VPN traffic exclusively over the MPLS connection.

Label switching

When traffic uses an MPLS connection between a CSC device and the backbone carrier, the backbone carrier manages the traffic through label-switched paths and does not store any information about the customer carrier routes.

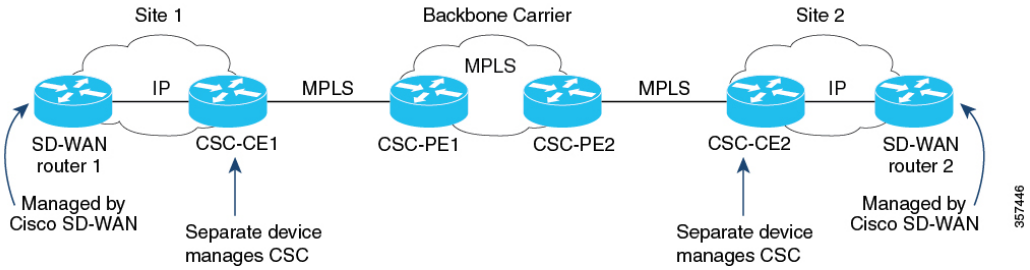
Edge device functioning as a CSC-CE device

The Cisco IOS XE Catalyst SD-WAN software simplifies carrier supporting carrier network topologies by enabling a single device to perform both SD-WAN edge and CSC-CE functions.

Before Cisco IOS XE Catalyst SD-WAN Release 17.6.1a

In releases earlier than 17.6.1a, each site in a CSC network topology used two separate devices: an edge device managed by Cisco Catalyst SD-WAN and a dedicated CSC-CE device. This is because the Cisco IOS XE Catalyst SD-WAN device could not function as a CSC-CE. See the illustration for the CSC topology before Cisco IOS XE Catalyst SD-WAN Release 17.6.1a.

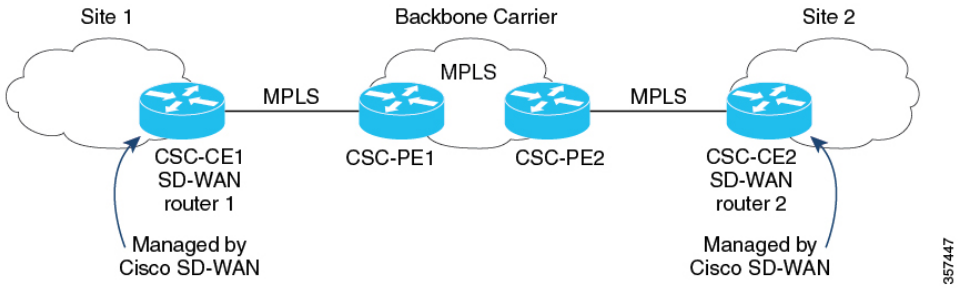
Figure 1: Carrier supporting carrier with Cisco Catalyst SD-WAN, before Cisco IOS XE Catalyst SD-WAN Release 17.6.1a



From Cisco IOS XE Catalyst SD-WAN Release 17.6.1a

From Cisco IOS XE Catalyst SD-WAN Release 17.6.1a, a Cisco IOS XE Catalyst SD-WAN device can function as a CSC-CE, removing the need for a separate dedicated CSC-CE device. See the illustration for the simplified CSC topology with Cisco IOS XE Catalyst SD-WAN devices providing CSC-CE functionality.

Figure 2: Carrier supporting carrier with Cisco Catalyst SD-WAN, Cisco IOS XE Catalyst SD-WAN Release 17.6.1a and later



Use cases for carrier supporting carrier

Carrier Supporting Carrier (CSC) enables secure, private transport of multiple customer networks over a shared backbone.

Global organizations

Global organizations can use Cisco Catalyst SD-WAN to support CSC with a backbone carrier, enabling multiple, separate divisions of an organization to maintain private traffic while sharing a common backbone carrier.

Service providers

Service providers that implement a CSC topology can benefit from Cisco Catalyst SD-WAN, as it allows carrier edge devices to handle CSC functionality without requiring a separate device.

Configure carrier supporting carrier

Use one of these methods to configure carrier supporting carrier.

- Feature template
- CLI commands

Configure carrier supporting carrier using a feature template

Follow these steps to configure a CE device for CSC using a new feature template in Cisco SD-WAN Manager.

Procedure

Step 1 Create a new device template.

a. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.

b. Choose **Device Templates**, and click **Create Template**.

c. From the drop-down list, select **From Feature Template**.

Note
In Cisco vManage Release 20.7.x and earlier releases, **Device Templates** is titled as **Device**.

Step 2 Enter device details.

- In the **Device Model** field, choose the correct device model.
- In the **Device Role** field, select **SDWAN Edge**.
- In the **Template Name** field, enter a name for the template.

Step 3 Configure VPN settings.

a. In the **Transport & Management VPN** section, under **Cisco VPN 0,** choose a template to configure VPN 0.

For more, see Configure Interfaces in the WAN Transport VPN (VPN 0).

b. In the **Cisco VPN Interface Ethernet** field, choose a template to configure the interface.

For more information, see Configure VPN Ethernet Interface.

Step 4 In the **Transport & Management VPN** section, click **Cisco BGP** to add the Cisco BGP field.

For more information, see see Configure BGP Using SD-WAN Manager Templates.

Step 5 In the **MPLS Interface** section, under **Interface Name 1**, enter the interface used to connect the device to the backbone carrier.

Step 6 In the **Neighbor** section, click **Advanced Options** to configure CSC options.

Table 3: Configure CSC options

Field	Description
Send label	Choose **On** to enable CSC support.
Explicit null	Choose **On** if the device uses a loopback WAN interface.
As override	Choose **On** if CE1 and CE2 use the same autonomous system (AS) number.
Allowas in	Choose **On** if the two CE sites use the same AS number.

Step 7 Click **Save** to save the BGP configuration.

Step 8 Click **Create** to create the feature template.

The **Configuration** > **Templates** page displays the available templates.

Step 9 Attach the template to a device.

a. On the **Configuration** > **Templates** page, click **Device Templates**.

b. For the new template, click **…** and choose **Attach Devices**.

c. Move a device to the **Selected Devices** column and click **Attach**.

Configure carrier supporting carrier using CLI

You can use the BGP feature template to configure CSC instead of CLI commands.

Before you begin

Apply a BGP configuration to a Cisco IOS XE Catalyst SD-WAN device before you configure it for CSC-CE functionality.

Follow these steps to configure a CE device for CSC using the CLI.

Procedure

Step 1 Configure CSC-CE1

a) Map MPLS labels to VRFs.

The device checks the MPLS label of incoming traffic and uses the IP lookup table of the VRF mapped to that label.For example, if the MPLS label 10 maps to VRF 1, the router uses the IP lookup table of VRF 1 for traffic with label 10.

```
device# config-transaction
device(config)# mpls label mode all-vrfs protocol bgp-vpnv4 per-vrf
device(config)# mpls label mode all-vrfs protocol bgp-vpnv6 per-vrf
device(config)# mpls label range min-label max-label static min-static-label max-static-label
```

b) Enable MPLS on the interface.

```
device(config)# interfaceinterface
device(config-if)# mpls bgp forwarding
```

c) Configure BGP.

```
device(config)# router bgp bgp-number
device(config-router)# neighbor neighbor-ip> allowas-in
```

d) Advertise MPLS labels when using a loopback WAN interface.

```
device(config-router)# neighbor <neighbor-ip> send-label explicit-null
```

Note
Using `send-label explicit-null` on non-loopback interfaces does not affect performance.

Step 2 Configure CSC-CE2

a. Map MPLS labels to VRFs.

```
device# config-transaction
device(config)# mpls label mode all-vrfs protocol bgp-vpnv4 per-vrf
device(config)# mpls label mode all-vrfs protocol bgp-vpnv6 per-vrf
device(config)# mpls label range min-label max-label static min-static-label max-static-label
```

b. Enable MPLS on the interface.

```
device(config)# interface interface
device(config-if)# mpls bgp forwarding
```

c. Configure BGP.

```
evice(config)# router bgp bgp-number
Device(config-router)# neighbor neighbor-ip as-override
Device(config-router)# neighbor neighbor-ip send-label explicit-null
```

The following example shows CSC-CE1 and CSC-CE2 configurations with BGP and MPLS:

- CSC-CE1: 10.1.1.10
- CSC-CE2: 10.1.1.20

- CSC-PE1 (neighbor of CSC-CE1): 10.2.2.10
- CSC-PE2 (neighbor of CSC-CE2): 10.2.2.20

CSC-CE1 Configuration

```
mpls label mode all-vrfs protocol bgp-vpnv4 per-vrf
mpls label mode all-vrfs protocol bgp-vpnv6 per-vrf
mpls label range 100000 1048575 static 16 99

interface GigabitEthernet2
 no shutdown
 mpls bgp forwarding
 ip address 10.1.1.15 255.255.255.0

router bgp 10
 bgp log-neighbor-changes
 bgp router-id 172.16.255.15
 neighbor 10.1.1.20 remote-as 100
 neighbor 10.1.1.20 fall-over bfd
 address-family ipv4 unicast
  maximum-paths 4
  neighbor 10.1.1.20 activate
  neighbor 10.1.1.20 advertisement-interval 30
  neighbor 10.2.2.10 allowas-in
  neighbor 10.2.2.10 send-label explicit-null
  neighbor 10.1.1.20 send-community both
 exit-address-family
 timers bgp 60 180
```

CSC-CE2 Configuration

```
mpls label mode all-vrfs protocol bgp-vpnv4 per-vrf
mpls label mode all-vrfs protocol bgp-vpnv6 per-vrf
mpls label range 100000 1048575 static 16 99

interface GigabitEthernet5
 ip address 10.0.6.11 255.255.255.0
 negotiation auto
 mpls bgp forwarding

router bgp 10
 bgp log-neighbor-changes
 bgp router-id 172.16.255.11
 neighbor 10.1.1.10 remote-as 200
 address-family ipv4 unicast
  neighbor 10.1.1.10 activate
  neighbor 10.1.1.10 advertisement-interval 30
  neighbor 10.2.2.20 as-override
  neighbor 10.2.2.20 send-label explicit-null
  network 10.0.7.0 mask 255.255.255.0
  redistribute connected
  redistribute static
 exit-address-family
```

Verify device configuration for carrier supporting carrier

To verify if a device is correctly configured to reach the remote CSC-CE device using MPLS-labeled routing.

Procedure

Step 1 Run the following command on the device:

Example:

```
show ip route remote-csc-ce-device-address
```

Step 2 Confirm if the output displays a routing entry for the remote site IP address.

Step 3 Check if the output includes one or more routing descriptor blocks that describe the next-hop addresses for the path to the remote CSC-CE device.

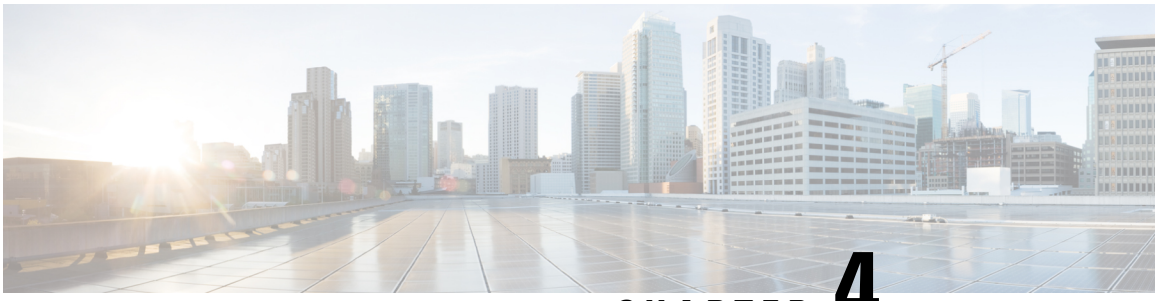
Step 4 Ensure that each descriptor block contains an MPLS label.

- If the device is configured correctly, the output shows:

```
Device# show ip route 10.0.1.100
Routing entry for 10.0.1.0/24
...
Routing Descriptor Blocks:
* 10.1.1.100, from 10.1.1.100, 00:00:50 ago
...
MPLS label: 26
```

- If the device is not configured correctly, the output shows:

```
% Subnet not in table
```



CHAPTER 4

DHCP Vendor Option Support

Feature history for DHCP vendor option support

Table 4: Feature History Table

Feature Name	Release Information	Description
DHCP Option Support	Cisco IOS XE Catalyst SD-WAN Release 16.12.1b	This feature allows DHCP server options, 43 and 191 to configure vendor-specific information in client-server exchanges.
DHCP vendor option support	Cisco IOS XE Catalyst SD-WAN Release 17.10.1a Cisco Catalyst SD-WAN Control Components Release 20.10.1	This feature allows DHCP client option 124 and option 125 to configure vendor-specific information in client-server exchanges. Configure this feature using the CLI Add-on feature template in Cisco SD-WAN Manager.

DHCP vendor option support

A configurable dynamic host configuration protocol (DHCP) client is a network client feature that:

- allows use of user-specified client or class identifiers,
- supports suggestion of lease time during address requests, and
- enables configuration of vendor-class and vendor-specific information with Option 124 and Option 125.

This functionality provides flexibility for DHCP clients when requesting addresses and enables differential services, device classification, and feature enablement.

DHCP client options

The DHCP client provides flexibility by allowing the following options to be configured for a DHCP client:

- Option 124—This option is used by DHCP clients and servers to exchange vendor-class information.
- Option 125—This option is used by DHCP clients and servers to exchange vendor-specific information.

These options are used by Zero-Touch Provisioning (ZTP), Cisco Plug-and-Play (PnP), and Identity Services Engine (ISE) to support multiple use cases.

For example, the content of Option 124 is used for device classification and enabling solution-specific features.

The DHCP Vendor Option Support feature introduces new CLI parameters to make Option 124 and Option 125 flexible. You can modify and customize enabling vendor specific options to carry different values for different customer features. The combination of Option 124 and Option 125 enables various features.

The **ip dhcp client vendor-class** command provides flexibility to pack either Device PID or MAC Address of the DHCP client or any user configurable string in option-124. The default behavior for this command is to continue to send device PID when you choose option 124.

This default behavior can be overridden to carry MAC Address in Day 1 configuration mode by explicitly requesting option-125 from the server using the **ip dhcp client vendor-class** command.

By default, Cisco IOS XE DHCP client sends the following data:

Attribute	IPv4 DHCP Option	Default Value
Vendor-Identifying Vendor Class Option	124	PID

Note The **ip dhcp client vendor-class [mac-address | ascii | disable | hex]** command overrides PID with MAC Address / user defined string / disable Option 124.

DHCPv6 Client Options

Configuration parameters and other control information are carried in tagged data items that are stored in the options field of the DHCP message.

The DHCP client provides flexibility by allowing the following options to be configured for a DHCP client:

- Option 16—This option is used by DHCP clients and servers to exchange vendor-class information.
- Option 17—This option is used by DHCP clients and servers to exchange vendor-specific information

In DHCPv6, option-16 and option-17 are used by DHCP clients and servers to exchange vendor-specific information.

The **ipv6 dhcp client vendor-class** command provides flexibility to pack either Device PID or MAC Address of the DHCP Client or any user configurable string in option-16.

The default behavior for this command is to continue to send device PID when you choose option-16 but it can be overridden to carry MAC Address in Day 1 configuration mode using the **ipv6 dhcp client vendor-class** command.

By default, Cisco IOS XE DHCP client sends the following data:

Attribute	IPv4 DHCP Option	Default Value
Vendor Class Option	16	PID

Note The **ipv6 dhcp client vendor-class [mac-address | hex | ascii | disable]** command can be used to override default value of PID with MAC Address or User defined string or disable the option.

Configure a DHCP vendor option using CLI command

This section provides a sample CLI configuration to configure a DHCP vendor option.

For more information about using CLI templates, see CLI Add-On Feature Templates.

Before you begin

By default, CLI templates execute commands in global config mode.

Procedure

Step 1 Configure an interface type and enter the interface configuration mode.

Example:

```
interface type number
```

Step 2 Acquire an IP address on an interface from DHCP.

Example:

```
ip address dhcp
```

Step 3 Configure the DHCP vendor-class option.

Example:

```
ip dhcp client vendor-class [mac-address | ascii | hex | disable]
```

Note
You must first configure the **no ip dhcp-client** command before configuring the IP address.

Override the device PID with MAC address:

The DHCP vendor-class option overrides the device PID with the MAC Address.

```
interface GigabitEthernet 0/0/0
    ip address dhcp
    ip dhcp client vendor-class mac-address
    !
```

Override the device PID with user defined string in hex or in ascii format:

```
interface GigabitEthernet 0/0/0
    ip address dhcp
    ip dhcp client vendor-class hex aabbcc
    !

interface GigabitEthernet 0/0/0
    ip address dhcp
    ip dhcp client vendor-class ascii cisco
    !
```

Disable option-124 in DHCP messages:

```
interface GigabitEthernet 0/0/0
    ip address dhcp
    ip dhcp client vendor-class disable
    !
```

Configure a DHCPv6 client option option using a CLI template

By default, CLI templates execute commands in global config mode.

For more information about using CLI templates, see *CLI Add-On Feature Templates* in the *Cisco Catalyst SD-WAN Systems and Interfaces Configuration Guide, Cisco IOS XE Catalyst SD-WAN Release 17.x*.

The section provides a sample CLI configuration to configure DHCPv6 client option.

Procedure

Step 1 Configure an interface type and enter the interface configuration mode.

Example:

```
interface type number
```

Step 2 Acquire an IPv6 address on an interface from DHCP.

Example:

```
ipv6 address dhcp
```

Step 3 Configure the DHCP vendor-class option.

Example:

```
ipv6 dhcp client vendor-class {mac-address | ascii | hex | disable}
```

Note
By default, the DHCPv6 client carries the device PID of the device in option-16.

This default behaviour can be overridden by configuring the **ipv6 dhcp client vendor-class** command.

Override the device PID with the MAC address:

The DHCP vendor-class option overrides the Device PID with the MAC Address.

```
interface GigabitEthernet 0/0/0
    ipv6 address dhcp
    ipv6 dhcp client vendor-class mac-address
    !
```

Override the device PID with user defined string in hex or in ascii format:

```
interface GigabitEthernet 0/0/0
    ipv6 address dhcp
    ipv6 dhcp client vendor-class hex aabbcc
    !

interface GigabitEthernet 0/0/0
    ipv6 address dhcp
    ipv6 dhcp client vendor-class ascii cisco
    !
```

Disable option-16 in DHCP messages:

```
interface GigabitEthernet 0/0/0
    ipv6 address dhcp
    ipv6 dhcp client vendor-class disable
    !
```

Configure DHCP server using templates

To configure a Cisco IOS XE Catalyst SD-WAN device to act as a DHCP server using Cisco SD-WAN Manager templates:

1. Create a DHCP-Server feature template to configure DHCP server parameters, as described in this topic.
2. Create one or more interface feature templates, as described in the VPN-Interface-Ethernet and the VPN-Interface-PPP-Ethernet sections.
3. Create a VPN feature template to configure VPN parameters.

To configure a Cisco IOS XE Catalyst SD-WAN device interface as a DHCP helper to broadcast DHCP requests from DHCP servers, in the **DHCP Helper** field of the applicable interfaces template, enter the addresses of the DHCP servers.

Use the DHCP-Server template for all Cisco Catalyst SD-WANs.

You enable DHCP server functionality on a Cisco IOS XE Catalyst SD-WAN device interface so it can assign IP addresses to hosts in the service-side network.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.

Step 2 Click **Device Templates**, and then click **Create Template**.

In Cisco vManage Release 20.7.1 and earlier releases, **Device Templates** is titled **Device**.

a) From the **Create Template** drop-down list, choose **From Feature Template**.
b) From the **Device Model** drop-down list, select the type of device for which you are creating the template.
c) Click **Service VPN** or scroll to the **Service VPN** section.
d) From **Additional VPN Templates**, click **VPN Interface**.
e) From the **Sub-Templates** drop-down list, choose DHCP Server.
f) From the **DHCP Server** drop-down list, click **Create Template**. The DHCP-Server template form is displayed.

This form contains fields for naming the template, and fields for defining the DHCP Server parameters.

g) In **Template Name**, enter a name for the template.

The name can be up to 128 characters and can contain only alphanumeric characters.

h) In **Template Description**, 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 list.

Step 3 Configure DHCP basic server functionality.

Table 5:

Parameter Name	Description
Address Pool*	Enter the IPv4 prefix range, in the format *prefix/length*, for the pool of addresses in the service-side network for which the router interface acts as DHCP server.
Exclude Addresses	Enter one or more IP addresses to exclude from the DHCP address pool. To specify multiple individual addresses, list them separated by a comma. To specify a range of addresses, separate them with a hyphen.
Maximum Leases	Specify the number of IP addresses that can be assigned on this interface. Range: 0 through 4294967295
Lease Time	Specify how long a DHCP-assigned IP address is valid. Range: 0 through 4294967295 seconds

Parameter Name	Description
Offer Time	Specify how long the IP address offered to a DHCP client is reserved for that client. By default, an offered IP address is reserved indefinitely, until the DHCP server runs out of addresses. At that point, the address is offered to another client. Range: 0 through 4294967295 seconds Default: 600 seconds
Administrative State	Select Up to enable or Down to disable the DHCP functionality on the interface. By default, DHCP server functionality is disabled on an interface.

Step 4 Configure a static lease to assign a static IP address to a client device on the service-side network.

Table 6:

Parameter Name	Description
MAC Address	Enter the MAC address of the client to which the static IP address is being assigned.
IP Address	Enter the static IP address to assign to the client.
Hostname	Enter the hostname of the client device.

To edit a static lease, click **pencil** icon.

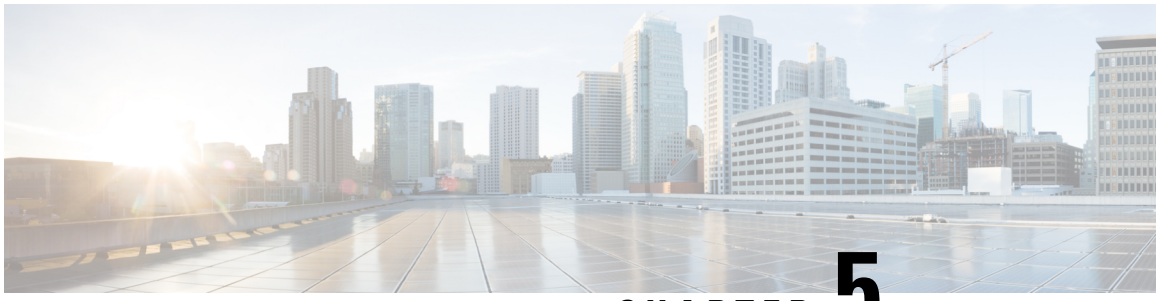
To remove a static lease, click **trash** icon.

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

Step 5 Configure advanced DHCP server options

Table 7:

Parameter Name	Description
Interface MTU	Specify the maximum MTU size of packets on the interface. Range: 68 to 65535 bytes
Domain Name	Specify the domain name that the DHCP client uses to resolve hostnames.
Default Gateway	Enter the IP address of a default gateway in the service-side network.
DNS Servers	Enter one or more IP address for a DNS server in the service-side network. Separate multiple entries with a comma. You can specify up to eight addresses.
TFTP Servers	Enter the IP address of a TFTP server in the service-side network. You can specify one or two addresses. If two, separate them with a comma.



CHAPTER 5

Dynamic On-Demand Tunnels

Feature history for dynamic on-demand tunnels

Table 8: Feature History Table

Feature Name	Release Number	Description
Dynamic On-Demand Tunnels	Cisco IOS XE Catalyst SD-WAN Release 17.3.1a Cisco SD-WAN Release 20.3.1	This feature enables you to configure an inactive state for tunnels between edge devices. This configuration reduces performance demands on devices and decreases network traffic.

Feature Name	Release Number	Description
Dynamic On-Demand Tunnels with Transport Gateways	Cisco IOS XE Catalyst SD-WAN Release 17.12.1a Cisco Catalyst SD-WAN Manager Release 20.12.1	A transport gateway can serve as the hub between two spoke devices. It provides the backup route that is necessary for spoke-to-spoke on-demand tunnels to operate. Using a transport gateway as the hub simplifies the process of enabling on-demand tunnels. This method does not require any changes to control policy on Cisco SD-WAN Controllers.

Dynamic on-demand tunnels

A Cisco Catalyst SD-WAN on-demand tunnel is a network feature that:

- automatically establishes secure tunnels between spoke devices when traffic starts,
- uses a configurable inactivity timer to remove tunnels after traffic stops, and
- conserves bandwidth and device performance when inactive.

On-demand tunnels enable efficient, temporary connections that optimize network resource usage.

Cisco Catalyst SD-WAN on-demand tunnels are triggered by device traffic and are removed after a set period of inactivity, ensuring inactive links do not consume bandwidth or affect performance.

How on-demand tunnels work

Summary

When you configure a site to use dynamic tunnels, the on-demand functionality is enabled. In this mode of operation, Cisco Catalyst SD-WAN edge routers do not bring up direct tunnels to other sites that are also enabled with on-demand functionality.

Cisco Catalyst SD-WAN selects one or more edge routers (typically centrally located routers) to act as backup forwarding node(s), providing a secondary path for traffic between two nodes. The backup node(s) are not enabled for on-demand. All on-demand sites form static tunnels with the backup node(s). The backup node(s) provide a static backup route for traffic between two nodes that have on-demand enabled.

The first packet of traffic between two nodes is routed through the static backup path, and triggers the on-demand tunnel to become active between the sites. The backup path continues to forward traffic until the direct path becomes active.

All on-demand sites learn the TLOCs and prefixes of all other on-demand remote sites. The prefixes also have a backup path set up through Cisco Catalyst SD-WAN Controller control policy. So in the control plane, the on-demand tunnel network has the same state as a full-mesh tunnel network, including a backup path.

The control plane downloads to the data plane, routes, with the backup path and remote TLOCs that represent a potential direct path between any two sites, but it does not set up a direct path tunnel to remote TLOCs.

All prefixes learned from remote sites must have a backup path. A less-specific aggregate route from a hub site is not a valid backup path. The specific prefix advertised from the remote branch must also be advertised as a backup from the hub.

If this specific prefix is missing as a backup, the on-demand tunnel setup will fail.

Therefore, the backup path must be a static tunnel that is always UP. This static tunnel should include both the specific prefixes from the remote branch and any necessary aggregate routes to ensure proper tunnel establishment and maintenance.

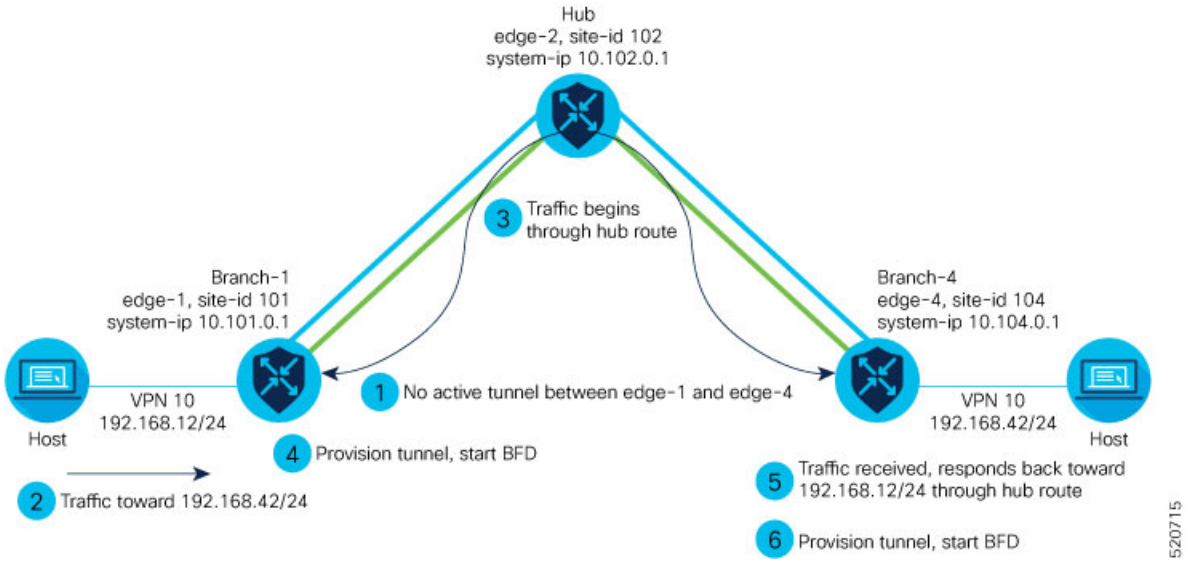
Traffic from either end of the on-demand tunnel triggers setting up the tunnel. This enables on-demand tunnels to accommodate network address translation (NAT) traversal.

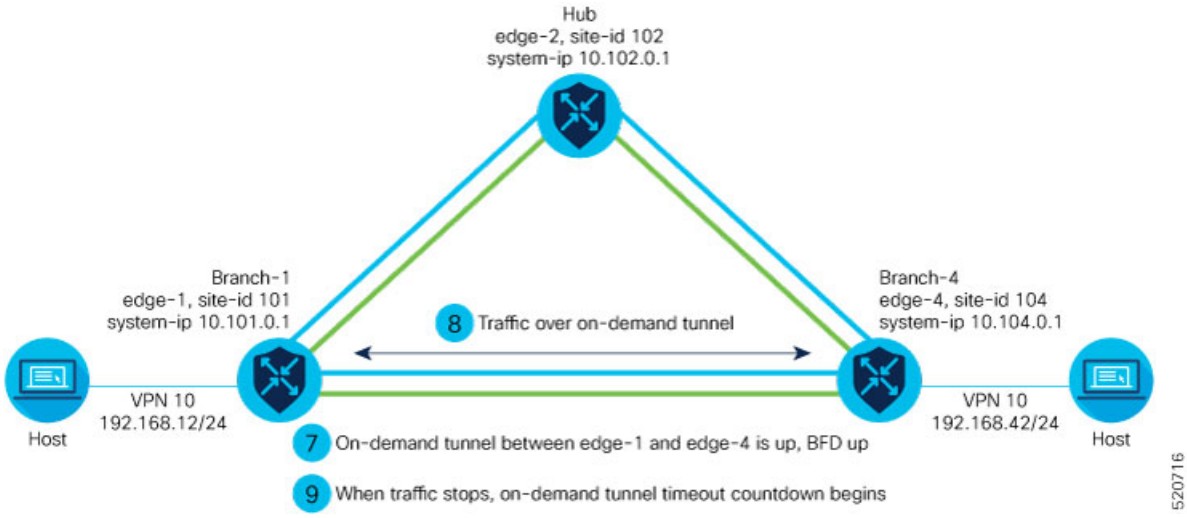
The on-demand tunnel feature introduces two states for the on-demand branch site:

- Inactive: The on-demand tunnel is not set up with the remote site. There is no active traffic to or from the remote site. Remote site TLOCs are inactive—no bidirectional forwarding detection (BFD) is set up, the prefix is installed with the inactive paths, and the backup path is set as the path to forward any traffic.
- Active: The on-demand direct site-to-site tunnel is set up to the remote site. There is active traffic to or from the remote site. This state is identical to the case of a typical tunnel, where the remote TLOCs have BFD set up, and the prefix is installed with the direct path tunnel. In this state, tunnel activity is tracked. If there is no traffic for the "idle time" duration (default 10 minutes), the direct site-to-site tunnel is removed and the state changes to Inactive.

Workflow

Figure 3: Process of On-Demand Tunnel Establishment Between Two Edge Routers





The steps below demonstrate what occurs between two edge routers with an on-demand tunnel configured.

1. There is no active tunnel between the two edge routers. edge-1 and edge-4 are in their inactive states.
2. The host behind edge-1 initiates traffic toward the host behind edge-4.
3. edge-1 forwards the traffic through the backup path using the hub or backup node to edge-4.
4. edge-1 provisions the on-demand tunnel and begins bidirectional forwarding detection (BFD). edge-4 is now in its active state on edge-1.
5. When edge-4 receives the return traffic for the host behind edge-1, it forwards the traffic through the backup path using the hub or backup node to edge-1.
6. edge-4 provisions the on-demand tunnel and begins BFD. edge-1 is now in active state on edge-4.
7. At this point, the on-demand tunnel between edge-1 and edge-4 is up, and BFD is up.
8. Traffic between the two edge devices takes the direct route through the on-demand tunnel.
9. Both edge-1 and edge-4 track the traffic activity on the on-demand tunnel in both directions.

   If there is no traffic for the idle timeout duration, the on-demand tunnel is deleted, and the edge-1 and edge-4 devices go back to the inactive state.

How on-demand tunnels work with a transport gateway

Summary

A transport gateway can serve as the hub between two spoke devices, providing the backup route that is necessary for spoke-to-spoke on-demand tunnels to operate. Using a transport gateway as the hub simplifies the process of enabling on-demand tunnels. This method does not require configuring control policy on Cisco SD-WAN Controllers.

Workflow

For information about configuration, see *Configure On-Demand Tunnels Using a Transport Gateway*.

Prerequisites for on-demand tunnels

There are several prerequisites for using on-demand tunnels:

- Configure a Centralized Control Policy for On-Demand Tunnels
- Prerequisites: OMP Settings
- Prerequisites: Hub Device Traffic Engineering Service
- Prerequisites: Spoke Device ECMP Limit
- Prerequisites: OMP Settings
- Prerequisites: Hub Device Traffic Engineering Service
- Prerequisites: Spoke Device ECMP Limit

Prerequisites: OMP settings

When on-demand tunnels are enabled, spokes use backup paths through the hub, so a higher path limit is necessary. The direct paths as well as the backup paths need to be advertised. To accommodate this, increase the Cisco Catalyst SD-WAN Controller send-path-limit to advertise all available paths. We recommend to use the maximum possible value.

Note If there are too many Hub TLOCs configured in the on-demand tunnel control policy, the recommended value for send-path-limit is not enough always. In such cases, the on-demand tunnel feature will not work at all.

Starting from Cisco vManage Release 20.8.1 and Cisco IOS XE Catalyst SD-WAN Release 17.8.1a, the maximum send-path-limit is 32. In Cisco vManage Release 20.7.x and earlier releases, the maximum send-path-limit is 16.

For information about configuring Cisco SD-WAN Controller send-path-limit, see the routing configuration guides on the Cisco Catalyst SD-WAN Configuration Guides page.

Before you begin

The Cisco Catalyst SD-WAN Controller send-path-limit must be more than the default 4.

Procedure

Step 1 Configure the OMP send path limit using a feature template

For a Cisco Catalyst SD-WAN Controller in Managed mode, use this procedure to configure the OMP send path limit.

To confirm that it is in Managed mode, from the Cisco SD-WAN Manager menu, choose **Configuration** > **Control Components**. For a **Controller** row, the **Managed By** column shows a template name.

a) From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
b) Click **Feature Templates.** In Cisco vManage Release 20.7.x and earlier releases, **Feature Templates** is titled **Feature**.
c) Edit or create an OMP template for device type Controller. In some earlier releases, the device type is called vSmart. Any functioning Cisco Catalyst SD-WAN deployment has at least one vSmart OMP template configured.
d) In **Basic Configuration**, set the **Number of Paths Advertised per Prefix** to 16 (recommended). From Cisco SD-WAN Release 20.8.1, the maximum is 32.

Step 2 Configure the OMP send path limit for an edge device using CLI commands

Use a CLI add-on profile, CLI add-on feature template, or CLI template to execute these CLI commands. By default, CLI templates and the CLI add-on profile execute commands in global configuration mode. For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates.

For an edge device, use these commands to configure an OMP send path limit.

a) Enter OMP configuration mode.
b) Set the send path limit. We recommend 16.

From Cisco SD-WAN Release 20.8.1, the maximum is 32.

Example:

```
sdwan
omp
 send-path-limit 16
```

Prerequisites: Hub device traffic engineering service

Before you begin

On the hub device, the Traffic Engineering service (service TE) must be enabled.

This ensures that the Cisco Catalyst SD-WAN Overlay Management Protocol (OMP) on the spoke devices accepts the backup path through the hub, which is being added as an intermediate path between the two spoke devices. Without this, the backup path through the hub would be considered invalid and unresolved by the spoke devices.

Procedure

Step 1 Enable the Traffic Engineering Service Using Cisco SD-WAN Manager

a) In Cisco SD-WAN Manager, open **Configuration** > **Templates**
b) Click **Feature Templates**.

In Cisco vManage Release 20.7.x and earlier releases, **Feature Templates** is titled **Feature**.

c) Click **Add Template**.
d) Select a platform.
e) From **VPN**, select **VPN**.

f) Ensure that in **Basic Configuration**, the **VPN** field is set to 0.
g) From **Service**, click **New Service** and select **TE**.
h) Click **Add**, and then click **Update**. The TE service appears in the table of services.
i) Apply the VPN-0 template to the hub

Step 2 Enable the Traffic Engineering Service, Using CLI Commands (Cisco IOS XE Catalyst SD-WAN Devices)

Use a CLI add-on profile, CLI add-on feature template, or CLI template to execute these CLI commands. By default, CLI templates and the CLI add-on profile execute commands in global configuration mode. For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates.

Example:

```
sdwan
 service TE vrf global
exit
```

Step 3 Enable the Traffic Engineering Service, Using CLI Commands (Cisco vEdge Devices)

Example:

```
vpn 0
 service TE
 exit
```

Prerequisites: Spoke Device ECMP Limit

Before you begin

On spoke devices, the ECMP limit must be more than the default 4. Recommended: 16

When on-demand tunnels are enabled, spoke devices create both direct and backup paths. To accommodate the need for more paths, increase the ECMP limit.

Procedure

Step 1 Configure the ECMP Limit Using a Configuration Group

On the **Configuration** > **Configuration Groups** page, choose the SD-WAN solution type.

a) From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration Groups**.
b) Do one of these:

- Edit a profile directly:

  In the System Profile tab, create (**Add New)** or edit a System profile.

- Edit a profile in a configuration group:

  Open a configuration group and edit the System profile.

c) In the system profile, create or edit an OMP feature.
d) In the **Basic Configuration** section, use the **ECMP Limit** field to configure the equal-cost multi-path (ECMP) limit as 16 (recommended).

Step 2 Configure the ECMP Limit Using a Feature Template

a) From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
b) Click **Feature Templates**.

In Cisco vManage Release 20.7.x and earlier releases, **Feature Templates** is titled **Feature**.

c) Click **Add Template**.
d) Select a device and click **Cisco OMP**.
e) In **Basic Configuration**, set the **ECMP Limit** field to 16 (recommended).

Step 3 Configure the ECMP Limit Using CLI Commands

Use a CLI add-on profile, CLI add-on feature template, or CLI template to execute these CLI commands. By default, CLI templates and the CLI add-on profile execute commands in global configuration mode. For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates.

Example:

```
sdwan
omp
 no shutdown
 ecmp-limit       16
```

You can view the current ecmp-limit using the `show running-config omp` command.

Restrictions for on-demand tunnels

- PfR statistics collection starts fresh for each on-demand tunnel setup and does not cache statistics for deleted tunnels after idle timeout.
- Out-of-order (OOO) packets may occur when the router switches traffic from the backup path to the on-demand tunnel; the router forwards packets as received.
- Unidirectional and multicast flows do not trigger on-demand tunnel setup and continue to use the backup path.
- Do not configure a data policy that applies a **set tloc-list** action to an on-demand site TLOC; doing so will result in dropped traffic.
- If the Pair Wise Key (PWK) IPSEc feature is enabled, on-demand tunnels are not supported.
- All TLOCs in the system are reset (disabled and then enabled) when you execute **on-demand enable** or **no on-demand enable**.
- When provisioning on-demand tunnels, the edge device provisions tunnels to all TLOCs on the remote edge device; for multi-home sites, you must enable on-demand mode on all systems.
- The system keeps all edge devices using on-demand tunnels active if service or user traffic exists on any on-demand tunnel in either direction; tunnels can be enabled between two sites only if both are in on-demand mode.
- You must configure a backup path for all prefixes from on-demand remote sites; the backup path must always be UP. The setup or removal of on-demand tunnels does not affect overlay route (OMP) updates or service/LAN-side route updates (such as OSPF or BGP). If either site is not in on-demand mode, the system sets up static tunnels between the sites.

Configure on-demand tunnels

The following procedures describe how to configure on-demand tunnels using different methods, including using control policy, or a simpler method using a transport gateway as a hub.

- Configure On-Demand Tunnels Using Control Policy
- Configure On-Demand Tunnels Using a Transport Gateway
- Enable On-Demand Tunnels on a Spoke Device Using a Configuration Group
- Enable On-Demand Tunnels on a Spoke Device Using a Template
- Enable On-Demand Tunnels Using a CLI Template

Configure on-demand tunnels using control policy

To configure on-demand tunnels using the control policy method, do the following:

Procedure

Step 1 Configure a control policy, as described in Configure a centralized control policy for on-demand tunnels.

Step 2 Enable on-demand tunnels n spoke devices, as described in Enable on-demand tunnels on a spoke device using a template and Enable on-demand tunnels using a CLI template.

Configure a centralized control policy for on-demand tunnels

Before you begin

This procedure configures a centralized control policy on a Cisco Catalyst SD-WAN Controller to enable on-demand tunnels.

- The Cisco Catalyst SD-WAN Controller centralized control policy must include the `tloc-action backup` action.

  This ensures that the backup path through the hub for communication between all of the spoke devices.

- The Cisco Catalyst SD-WAN Controllerr centralized control policy must accept all spoke prefix routes.
- The Cisco Catalyst SD-WAN Controller centralized control policy must accept TLOCs of all spokes.

  For information about configuring a Cisco Catalyst SD-WAN Controller **centralized control policy,** see the policies configuration guides on the Cisco Catalyst SD-WAN Configuration Guides page.

- When configuring on-demand tunnels using a transport gateway, do not use the control policy procedure described here. For information, see Configure On-Demand Tunnels Using a Transport Gateway.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.

a) Select **Centralized Policy**.
b) Click **Add Policy**.
c) In the left pane, click **Site**.
d) Click **Next**.
e) Click **Add Topology** and select **Custom Control (Route & TLOC)**.
f) Enter a name and description for the topology.
g) Click **Sequence Type**.
h) In the **Add Control Policy** pop-up window, choose **Route**.
i) Click **Sequence Rule** to create a sequence.

Step 2 Click **Match**.

a) Among the match conditions, click **Site**.
b) In the **Match Conditions** area, click the **Site List** menu and choose a site list.
c) Click **Actions**, and then **Accept**.

Step 3 Among the actions, click **TLOC Action**.

a) In the **Actions** area, click the **TLOC Action** menu and choose **Backup**.
b) Among the actions, click **TLOC**.
c) In the **Actions** area, click the **TLOC List** menu and choose or create a TLOC list.
d) Click **Save Match and Actions**.

Step 4 Click **Default Action**.

a) In the **Default Action** area, click the pencil icon to edit.
b) Near the **Actions** label, click **Accept**.
c) Click **Save Control Policy**.
d) Click **Next** twice.
e) In the Topology tab, click **New Site/WAN Region List**.
f) Click **Outbound Site List** and choose a site list that defines the sites at which you are enabling on-demand tunnels.
g) Adjacent to the site list, click **Add**.
h) Enter a name and description for the policy.
i) Click **Save Policy**.

Configure centralized control policy for on-demand tunnels using a CLI policy

Before you begin

The Cisco Catalyst SD-WAN Controller must be managed by Cisco SD-WAN Manager.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Classic** > **Policies**.

Step 2 Open **Centralized Policy**.

Step 3 From **Custom Options**, choose **Centralized Policy** > **CLI Policy**.

Step 4 Click **Add Policy**.

Step 5 Enter the CLI commands for the policy.

Example:

```
control-policy Dynamic-Tunnel-Control-Policy
    sequence 100
     match route
      site-list Branches
     !
     action accept
      set
       tloc-action backup
       tloc-list Hub-TLOCs
      !
     !
     sequence 200
      match tloc
     !
     action accept
    !
  default-action accept
 !
 lists
  site-list Branches
   site-id 200
   site-id 300
  !
  tloc-list Hub-TLOCs
   tloc 10.0.0.1 color mpls encap ipsec
   tloc 10.0.0.1 color public-internet encap ipsec
!
!
apply-policy
 site-list Branches
  control-policy Dynamic-Tunnel-Control-Policy out
 !
!
```

Configure on-demand tunnels using a transport gateway

Before you begin

- On Cisco SD-WAN Controllers, configure the send path limit, as described in Prerequisites: OMP settings.
- On spoke devices, configure the ECMP limit, as described in Prerequisites: Spoke Device ECMP Limit.
- When using a transport gateway as a hub to support on-demand tunnels, there is no need to create or modify a control policy.

  Do not use the procedure described in Configure a Centralized Control Policy for On-Demand Tunnels.

Procedure

Step 1 Enable transport gateway functionality on a router serving as the hub, providing a backup route between spokes, as described in the Transport Gateway section of the *Cisco Catalyst SD-WAN Routing Configuration Guide*.

Step 2 Enable on-demand tunnels and configure the idle timeout on spoke devices as described in Enable on-demand tunnels on a spoke device using a template.

Enable on-demand tunnels on a spoke device using a configuration group

Before you begin

On the **Configuration** > **Configuration Groups** page, choose the **SD-WAN** solution type.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration Groups**.

Step 2 Do one of these:

- Edit a profile directly:

  In the System Profile tab, create (**Add New**) or edit a System profile.

- Edit a profile in a configuration group:

  Open a configuration group and edit the System profile.

Step 3 In the System profile, create (**Add New**) or edit a Basic feature.

Step 4 In the **Advanced** section, use the **On Demand Tunnel** control to enable on-demand tunnels.

Enable on-demand tunnels on a spoke device using a template

Before you begin

- See the Prerequisites for On-Demand Tunnels.
- Do not enable on-demand on the hub device.
- On the spoke devices, enable on-demand at the system level. In the case of multi-homed sites, enable on-demand on all systems at the site.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**

Step 2 Click **Feature Templates**.

Note
In Cisco vManage Release 20.7.x and earlier releases, **Feature Templates** is titled **Feature**.

Step 3 Click **Add Template**.

Step 4 Select a device.

Step 5 From **Basic Information**, select **Cisco System**.

Step 6 Click **Advanced**.

Step 7 Enable **On-demand Tunnel.**

Step 8 (optional) Configure the **On-demand Tunnel Idle Timeout** time. The default idle timeout value is 10 minutes. Range: 1 to 65535 minutes

Step 9 Attach the System feature template to the device template for the spoke device.

Enable on-demand tunnels using a CLI template

For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates.

By default, CLI templates execute commands in global configuration mode.

Before you begin

- See Prerequisites for On-Demand Tunnels.
- Do not enable on-demand on the hub device

Procedure

On the spoke devices, enable on-demand tunnels at the system level. In the case of multi-homed sites, enable on-demand on all systems in the site.

The default idle timeout value is 10 minutes. Range: 1 to 65535 minutes

Example:

```
system
     on-demand enable
     on-demand idle-timeout 10
```

Monitor the status of on-demand tunnels

The following sections describe procedures for monitoring the status of on-demand tunnels.

- View the Current Status of On-Demand Tunnels Using Cisco SD-WAN Manager
- View a Chart of the On-Demand Tunnel Status Over Time in Cisco SD-WAN Manager
- View the Route to a Destination Device

View the current status of on-demand tunnels using Cisco SD-WAN Manager

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Monitor** > **Devices**.

For Cisco Catalyst SD-WAN Control Components Release 20.6.x and earlier:

From the Cisco SD-WAN Manager menu, choose **Monitor** > **Network**.

Step 2 Select a device.

Step 3 Select **Real Time**.

Step 4 For **Device Options**, select one of the following:

- **On Demand Local**: Displays the status of on-demand tunnels on the specified device.
- **On Demand Remote**: Displays the status of on-demand tunnels on the specified device, and on all connected devices.

The output is equivalent to executing the `show [sdwan] system on-demand [remote-system] [system-ip ip-address] CLI` command. It displays the status of on-demand tunnels.

View a chart of the on-demand tunnel status over time in Cisco SD-WAN Manager

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Monitor** > **Devices**.

Cisco Cisco Catalyst SD-WAN Control Components Release 20.6.x and earlier: From the Cisco SD-WAN Manager menu, choose **Monitor** > **Network** > > **.**

Step 2 Select a device.

Step 3 From WAN, choose Tunnel.

Step 4 From the **Chart Options** drop-down list, select **On-Demand Tunnel Status.** The chart shows the status of tunnels as `ACTIVE` or `INACTIVE`. `INACTIVE` indicates that an on-demand tunnel is in its inactive mode.

View the route to a destination device

Viewing the route between routers A and B can show whether the route is using an on-demand tunnel. On router A, use the `traceroute` command and enter router B as the destination. The command output shows whether the current route includes a hop at a hub device or whether the route is directly to the destination.

In the following examples, the router IP addresses are as follows:

- Router A: 10.1.1.1

- Router B: 10.1.1.2
- Hub device: 10.100.1.100

No Active On-Demand Tunnel

In the following example, there is no active on-demand tunnel between routers A and B, so the route includes the hub device. Note that it takes two hops to reach router B.

```
RouterA#traceroute vrf 1 10.1.1.2 numeric
Type escape sequence to abort.
Tracing the route to 10.1.1.2
VRF info: (vrf in name/id, vrf out name/id)
1 10.100.1.100 10 msec 8 msec 0 msec
2 10.1.1.2 2 msec * 1 msec
```

Active On-Demand Tunnel

```
RouterA#traceroute vrf 1 10.1.1.2 numeric
Type escape sequence to abort.
Tracing the route to 10.1.1.2
VRF info: (vrf in name/id, vrf out name/id)
1 10.1.1.2 1 msec
```

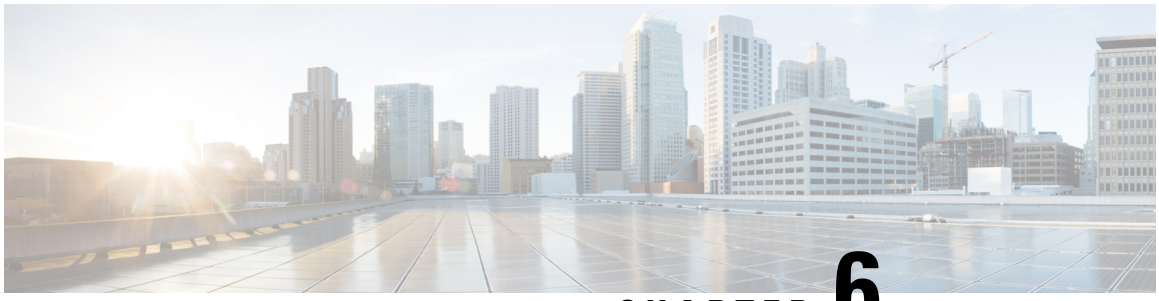
In the following example, there is an active on-demand tunnel between routers A and B, so the route from router A and to router B is direct.

View OMP routes

Viewing OMP routes can show the status of on-demand tunnels between two routers. Use the show `sdwan omp routes` command and view the **STATUS** column. The following table shows the possible values for this column, depending on whether an on-demand tunnel is active or not between two routers:

Table 9: Status of Routes, with or without an Active On-Demand Tunnel Between Two Routers

On-Demand Tunnel Between Routers A and B	STATUS for OMP Routes Between Routers A and B	STATUS for Backup Routes (through the Hub)
Not active	I, U, IA (installed, unresolved, and inactive)	C, I, R (chosen, installed, and resolved)
Active	C, I, R (chosen, installed, and resolved)	R (resolved)



CHAPTER 6

GRE-in-UDP

Feature history for GRE-in-UDP

Table 10: Feature History

Feature Name	Release Information	Description
GRE-in-UDP	Cisco IOS XE Catalyst SD-WAN Release 17.11.1a Cisco vManage Release 20.11.1	You can configure GRE encapsulation for UDP transport.

GRE-in-UDP

A GRE-in-UDP protocol is a network protocol that

- encapsulates Generic Routing Encapsulation (GRE) tunnels within a User Datagram Protocol (UDP) packet to facilitate load balancing and improve performance over networks, and
- enables a router to encapsulate GRE packets—including source and destination port information—within a UDP header, send the UDP packet through the tunnel, and allow the destination device to de-encapsulate the UDP packet.

Cisco Catalyst SD-WAN supports generic routing encapsulation (GRE) with UDP for IPv4 and IPv6 traffic.

Configure GRE-in-UDP using CLI commands

Follow these steps to configure GRE-in-UDP using a CLI template.

For more information about using CLI templates, see *CLI Add-On Feature Templates* and *CLI Templates*. By default, CLI templates execute commands in global config mode.

GRE-in-UDP is a protocol that encapsulates Generic Routing Encapsulation (GRE) tunnels within a User Datagram Protocol (UDP) packet to facilitate load balancing and improve performance over networks.

Procedure

Step 1 For the desired interface, enter interface configuration mode.

```
sdwan
interface interface
```

Step 2 Enter tunnel interface mode.

```
tunnel-interface
```

Step 3 Configure GRE encapsulation.

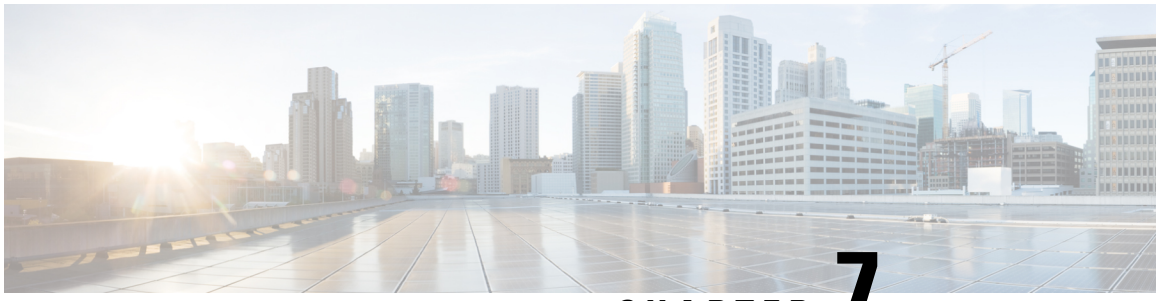
```
encapsulation gre
```

Step 4 Configure GRE-in-UDP as the encapsulation mode.

```
gre-in-udp
```

Here is a complete example of configuring GRE-in-UDP.

```
interface GigabitEthernet1
  tunnel-interface
   encapsulation gre
   color lte
   gre-in-udp
   no allow-service bgp
   allow-service dhcp
   allow-service dns
   allow-service icmp
   no allow-service sshd
   no allow-service netconf
   no allow-service ntp
   no allow-service ospf
   no allow-service stun
   allow-service https
   no allow-service snmp
   no allow-service bfd
  exit
```



CHAPTER 7

Hot Standby Router Protocol

Feature history table for hot standby router protocol

Table 11: Feature History

Feature Name	Release Information	Description
Support for HSRP and HSRP Authentication on Cisco IOS XE Catalyst SD-WAN Devices	Cisco IOS XE Catalyst SD-WAN Release 17.7.1a Cisco vManage Release 20.7.1	This feature allows you to configure HSRPv2 and HSRP authentication on Cisco IOS XE Catalyst SD-WAN platforms via CLI template. HSRP is a long-standing Cisco proprietary First Hop Redundancy Protocol (FHRP) to support version 2 of the protocol and authentication.

Hot standby router protocol

default

Hot Standby Router Protocol (HSRP) is a First Hop Redundancy Protocol (FHRP) that allows transparent failover of the first-hop IP device and provides high network availability. HSRP offers first-hop routing redundancy for IP hosts on networks configured with a default-gateway IP address. It identifies active and standby devices, supports multiple groups for load sharing, and uses virtual addresses for gateway redundancy. HSRP includes version 2 enhancements for stability and management, provides MD5 authentication for security, and enables dynamic-priority changes through object tracking.

HSRP version 2 support

Following are the HSRP version 2 (HSRPv2) features:

- HSRPv2 advertises and learns millisecond timer values. This change ensures stability of the HSRP groups in all cases.
- HSRPv2 expands the group number range from 0 to 4095.
- HSRPv2 provides improved management and troubleshooting. The HSRPv2 packet format includes a 6-byte identifier. This field is typically populated with the interface MAC address.
- HSRPv2 uses the IP multicast address 224.0.0.102 to send hello packets. This multicast address allows Cisco Group Management Protocol (CGMP) leave processing to be enabled concurrently with HSRP.
- HSRPv2 has a different packet format that uses a type–length–value (TLV) format.

HSRP MD5 authentication

HSRP supports two authentication schemes for protocol packets: simple plain-text strings and Message Digest 5 (MD5). HSRP MD5 authentication is an advanced authentication method that generates a Message Digest 5 (MD5) digest for the HSRP portion of the multicast HSRP protocol packet. This functionality provides added security and protects against the threat from HSRP-spoofing software.

MD5 authentication provides greater security than plain text authentication. MD5 authentication allows each HSRP group member to use a secret key to generate a keyed MD5 hash, which is part of the outgoing packet. A keyed hash of an incoming packet is generated; if the hash in the incoming packet does not match the generated hash, the packet is ignored.

You can provide the MD5 hash key directly in the configuration using a key string or supply it indirectly through a key chain.

HSRP packets will be rejected if one or more of the following conditions occur:

- Authentication schemes differ on the device and in the incoming packets.
- MD5 digests differ on the device and in the incoming packets.
- Text authentication strings differ on the device and in the incoming packets.

HSRP object tracking

Object tracking separates the tracking mechanism from HSRP and creates a stand-alone tracking process. Other processes and HSRP can use this tracking process. The priority of a device can change dynamically when it has been configured for object tracking, and the object that is being tracked goes down. Examples of objects that can be tracked are the line protocol state of an interface or the reachability of an IP route. If the specified object goes down, the HSRP priority is reduced.

How HSRP topologies work

Summary

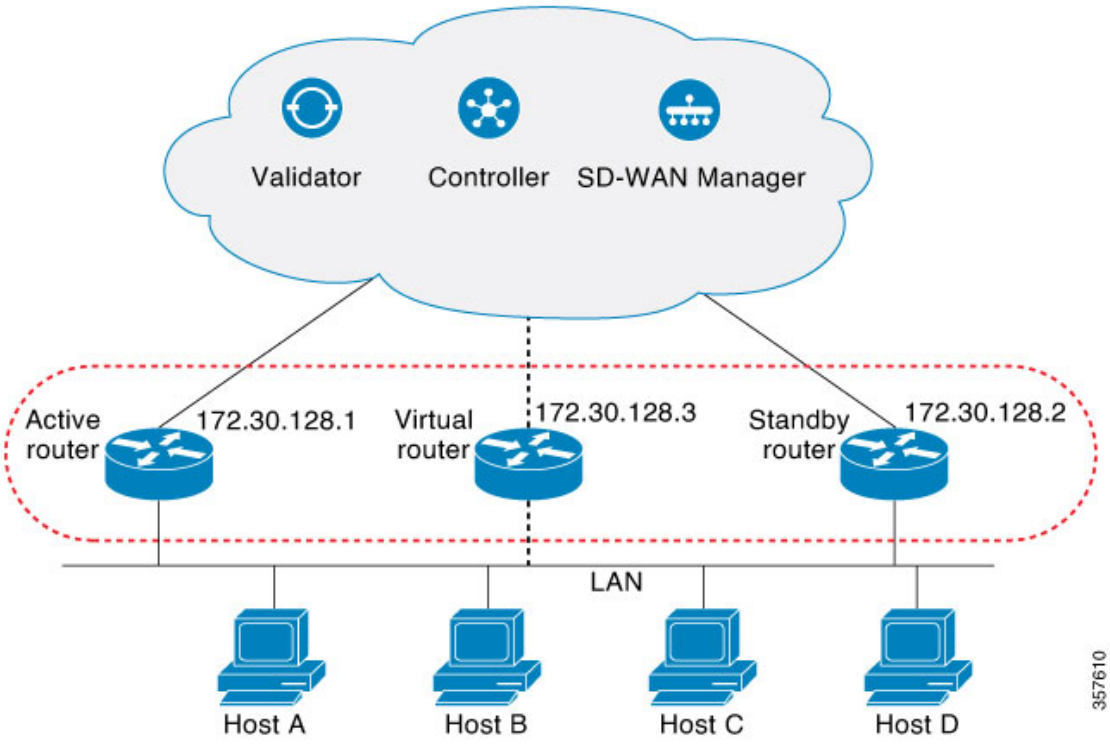
HSRP enables redundancy and reliability by allowing multiple routers to present as a single virtual gateway.

The key components involved in the process are:

- HSRP routers: Physical routers configured in a standby group.
- Virtual IP address: Shared address used by hosts as their default gateway.
- Virtual MAC address: Shared hardware address enabling failover.
- Hosts: End devices on the LAN that use the virtual gateway.

Workflow

Figure 4: HSRP Topology



The process involves the following stages:

1. The routers are configured into an HSRP standby group, sharing a virtual IP and MAC address.
2. One router assumes the active role, responding to traffic for the virtual IP.
3. Another router is in the standby role, monitoring the active router.
4. Hosts are configured with the virtual IP as their default gateway.
5. If the active router fails or does not send hello messages within a preset interval, the standby router becomes active and takes over packet forwarding.

Result

Hosts experience uninterrupted gateway service; redundancy is achieved, ensuring network reliability.

HSRP benefits

- Redundancy: HSRP employs a proven redundancy scheme and is widely deployed in large networks.
- Fast Failover: HSRP provides transparent, fast failover for the first-hop device.
- Preemption: Preemption allows a standby device to delay activation for a configurable time.
- Authentication: The HSRP Message Digest 5 (MD5) authentication algorithm safeguards against HSRP spoofing software and uses the MD5 standard to improve reliability and security.

Supported devices

- Cisco Catalyst 8500 Series Edge Platforms
- Cisco Catalyst 8300 Series Edge Platforms
- Cisco Catalyst 8200 Series Edge Platforms
- Cisco Catalyst 8200 uCPE Series Edge Platforms
- Cisco ASR 1000 Series Aggregation Services Routers
- Cisco ISR 1000 and ISR 4000 Series Integrated Services Routers (ISRs)
- Cisco ISR 1100 and ISR 1100X Series Integrated Services Routers (ISRs)
- Cisco IR1101 Integrated Services Router Rugged
- Cisco Catalyst 8000v Series Cloud Services Router

For details on supported models for each of these device families, refer to Cisco Catalyst SD-WAN Device Compatibility page.

Configure HSRP using the CLI

Before you begin

You can configure HSRP using the Cisco SD-WAN Manager CLI Add-on feature templates and CLI device templates. For more information on configuring using CLI templates, see CLI Templates. The following commands can be used in any order.

Procedure

Step 1 Enable HSRP on the interface.

Create (or enable) the HSRP group in IPv4 using its number and virtual IP address:

```
Device(config)# interface interface-type
Device(config-if)# standby group-number ip [ip-address [secondary]]
```

Activate HSRP in IPv6:

```
Device(config)# interface interface-type
Device(config-if)# standby group-number ipv6 {link-local-address | autoconfig }
```

Step 2 Set the HSRP version.

Note that the **nostandby** or **nostandby version 2** commands are rejected when the interface has IPv6 groups.

```
Device(config)# interface interface-type
Device(config-if)# standby version {1|2}
```

Step 3 Configure group priority and preemption.

Set the priority value used in choosing the active router, and configure HSRP preemption and preemption delay:

```
Device(config)# interface interface-type
Device(config-if)# standby group-number ip [ip-address [secondary]]
Device(config-if)# standby group-number priority [priority]
Device(config-if)# standby group-number preempt [ delay{ [ minimum seconds] [ reload seconds] [ sync
 seconds]}]
```

Step 4 Enable HSRP authentication (MD5 or text authentication).

- Configure HSRP MD5 authentication using a key chain.

  Key chains allow a different key string to be used at different times according to the key chain configuration. HSRP queries the appropriate key chain to obtain the current live key and key ID for the specified key chain.

  ```
  Device(config)# interface interface-type
  Device(config-if)# ip address ip-address mask [secondary ]
  Device(config-if)# standby group-number priority [priority]
  Device(config-if)# standby group-number preempt [ delay{ [ minimum seconds] [ reload seconds] [
   sync seconds]}]
  Device(config-if)# standby group-number authentication md5 key-chain key-chain-name
  Device(config-if)# standby group-number ip [ip-address [secondary]]
  ```

- Configure HSRP text authentication.

  The authentication string can be up to eight characters in length; the default string is Cisco.

  ```
  Device(config)# interface interface-type
  Device(config-if)# ip address ip-address mask [secondary ]
  Device(config-if)# standby group-number priority [priority]
  Device(config-if)# standby group-number preempt [ delay{ [ minimum seconds] [ reload seconds] [
   sync seconds]}]
  Device(config-if)# standby group-number authentication text string
  Device(config-if)# standby group-number ip [ip-address [secondary]]
  ```

Step 5 Adjust HSRP timers.

Configure the time between the hello packets and the time before other routers declare the active router to be inactive:

```
Device(config)# interface interface-type
Device(config-if)# standby group-number ip [ip-address [secondary]]
Device(config-if)# standby group-number timers hellotime holdtime
```

Step 6 Adjust HSRP object tracking.

Configure HSRP to track an object and change the HSRP priority based on the state of the object:

```
Device(config)# interface interface-type
Device(config-if)# standby group-number track object-number [decrement priority-decrement] [shutdown]
```

Step 7 Optimize CPU and network performance with HSRP multiple group optimization.

- Configure an HSRP group as a client group:

```
Device(config)# interface interface-type
Device(config-if)# standby group-number follow group-name
```

- Configure the HSRP client group refresh interval:

```
Device(config)# interface interface-type
Device(config-if)# standby group-number mac-refresh seconds
```

Step 8 Configure a specific virtual MAC address.

Specify a virtual MAC address for HSRP:

```
Device(config)# interface interface-type
Device(config-if)# standby group-number mac-address mac-address
```

Step 9 Link IP redundancy clients to HSRP groups.

Configure the name of a standby group:

```
Device(config)# interface interface-type
Device(config-if)# standby group-number name [redundancy-name]
```

The configured interface participates as a member of the specified HSRP group and provides high-availability failover with other routers in the group.

Example

The following is a complete HSRP configuration example on Cisco IOS XE Catalyst SD-WAN devices through CLI:

```
config-transaction
!
    interface GigabitEthernet0/0/1.94
    encapsulation dot1Q 94
    vrf forwarding 509
    ip address 10.96.194.2 255.255.255.0
    ip directed-broadcast
    ip mtu 1500
    ip nbar protocol-discovery
    standby version 2
    standby 1 preempt
    standby 94 ip 10.96.194.1
    standby 94 timers 1 4
    standby 94 priority 110
    standby 94 preempt delay minimum 180
    standby 94 authentication md5 key-string 7 094F471A1A0A
    standby 94 track 8 shutdown
    standby 194 ipv6 2001:10:96:194::1/64
    standby 194 timers 1 4
    standby 194 priority 110
    standby 194 preempt delay minimum 180
    standby 194 authentication md5 key-string 7 094F471A1A0A
    standby 194 track 80 shutdown
    ip policy route-map clear-df
    ipv6 address 2001:10:96:194::2/64
    ipv6 mtu 1500
```

```
  arp timeout 1200
  end
```

What to do next

- Verify HSRP operation and monitor for proper failover behavior.
- Adjust settings based on observed performance as needed.

Verify hot standby router protocol

The following is a sample output from the **show standby** command displaying the standby router information:

```
Device# show standby
GigabitEthernet0/0/1.94 - Group 94 (version 2)
  State is Standby
    1 state change, last state change 01:06:09
    Track object 8 state Up
  Virtual IP address is 10.96.194.1
  Active virtual MAC address is 0000.0c9f.f05e (MAC Not In Use)
    Local virtual MAC address is 0000.0c9f.f05e (v2 default)
  Hello time 1 sec, hold time 4 sec
    Next hello sent in 0.688 secs
  Authentication MD5, key-string
  Preemption enabled, delay min 180 secs
  Active router is 10.96.194.2, priority 110 (expires in 4.272 sec)
    MAC address is cc16.7e8c.6dd1
  Standby router is local
  Priority 105 (configured 105)
  Group name is "hsrp-Gi0/0/1.94-94" (default)
  FLAGS: 0/1
GigabitEthernet0/0/1.94 - Group 194 (version 2)
  State is Standby
    1 state change, last state change 01:06:07
    Track object 80 state Up
  Link-Local Virtual IPv6 address is FE80::5:73FF:FEA0:C2 (impl auto EUI64)
    Virtual IPv6 address 2001:10:96:194::1/64
  Active virtual MAC address is 0005.73a0.00c2 (MAC Not In Use)
    Local virtual MAC address is 0005.73a0.00c2 (v2 IPv6 default)
  Hello time 1 sec, hold time 4 sec
    Next hello sent in 0.480 secs
  Authentication MD5, key-string
  Preemption enabled, delay min 180 secs
  Active router is FE80::CE16:7EFF:FE8C:6DD1, priority 110 (expires in 4.032 sec)
    MAC address is cc16.7e8c.6dd1
  Standby router is local
  Priority 105 (configured 105)
  Group name is "hsrp-Gi0/0/1.94-194" (default)
  FLAGS: 0/1
```

The following is a sample output from the **show standby** command displaying HSRP Version 2 information if HSRP Version 2 is configured:

```
Device# show standby
Ethernet0/1 - Group 1 (version 2)
  State is Speak
  Virtual IP address is 10.21.0.10
  Active virtual MAC address is unknown
   Local virtual MAC address is 0000.0c9f.f001 (v2 default)
  Hello time 3 sec, hold time 10 sec
```

```
   Next hello sent in 1.804 secs
  Preemption enabled
  Active router is unknown
  Standby router is unknown
  Priority 20 (configured 20)
  Group name is "hsrp-Et0/1-1" (default)
Ethernet0/2 - Group 1
  State is Speak
  Virtual IP address is 10.22.0.10
  Active virtual MAC address is unknown
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.804 secs
  Preemption disabled
  Active router is unknown
  Standby router is unknown
  Priority 90 (default 100)
    Track interface Serial2/0 state Down decrement 10
  Group name is "hsrp-Et0/2-1" (default)
```

The following is a sample output from the **show standby** command displaying HSRP authentication information if HSRP MD5 authentication is configured:

```
Device# show standby
Ethernet0/1 - Group 1
  State is Active
    5 state changes, last state change 00:17:27
  Virtual IP address is 10.21.0.10
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 2.276 secs
  Authentication MD5, key-string, timeout 30 secs
  Preemption enabled
  Active router is local
  Standby router is unknown
  Priority 110 (configured 110)
  Group name is "hsrp-Et0/1-1" (default)
```

The following is a sample output from the **show standby brief** command displaying HSRP information for a specific interface:

```
Device# show standby brief
Interface   Grp  Pri P State   Active                  Standby   Virtual IP
Gi0/0/1.94  94   105 P Standby 10.96.194.2               local    10.96.194.1
Gi0/0/1.94  194  105 P Standby FE80::CE16:7EFF:FE8C:6DD1   local    FE80::5:73FF:FEA0:C2
```

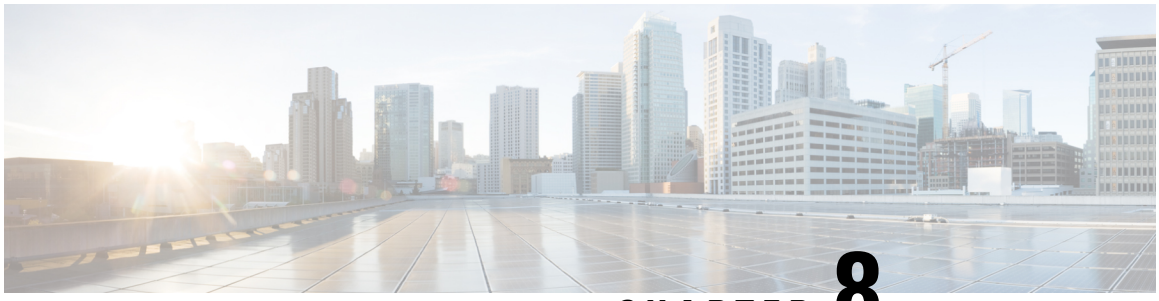
The following is a sample output from the **show standby neighbors** command displaying the HSRP neighbors on Ethernet interface 0/0. Neighbor 10.0.0.250 is active for group 2 and standby for groups 1 and 8, and is registered with BFD:

```
Device# show standby neighbors Ethernet0/0
HSRP neighbors on Ethernet0/0
  10.0.0.250
    Active groups: 2
    Standby groups: 1, 8
    BFD enabled
  10.0.0.251
    Active groups: 5, 8
    Standby groups: 2
    BFD enabled
  10.0.0.253
    No Active groups
    No Standby groups
```

```
    BFD enabled
```

The following is a sample output from the **show standby neighbors** command displaying information for all HSRP neighbors:

```
Device# show standby neighbors
HSRP neighbors on FastEthernet2/0
  10.0.0.2
    No active groups
    Standby groups: 1
    BFD enabled
HSRP neighbors on FastEthernet2/0
  10.0.0.1
    Active groups: 1
    No standby groups
    BFD enabled
```



CHAPTER 8

IP DHCP Smart-Relay

Feature history for IP DHCP smart-relay

Table 12: Feature History Table

Feature Name	Release Information	Feature Description
IP DHCP Smart-Relay	Cisco IOS XE Catalyst SD-WAN Release 17.13.1a Cisco Catalyst SD-WAN Manager Release 20.13.1	With this feature, you can set the gateway address to the secondary IP address using the DHCP relay agent, when there is no IP address and DHCP service information from the DHCP server. A DHCP relay agent is any host or IP router that forwards DHCP packets between clients and servers. This functionality is useful when the DHCP server cannot be configured to use secondary pools.

IP DHCP smart-relay

A Dynamic Host Configuration Protocol (DHCP) relay agent is a network host that

- forwards DHCP packets between clients and servers on different physical subnets,
- generates new DHCP messages on another interface, setting the gateway IP address and optionally adding relay agent information (option 82), and
- supports unnumbered interfaces by automatically managing static host routes for DHCP clients.

Relay-agent forwarding differs from standard IP routing; instead of transparent packet switching, the relay agent receives DHCP messages, creates new messages with updated information, and forwards these between interfaces.

When a DHCP reply is received from the server, the relay agent removes option 82 (if present) before forwarding it to the client. With unnumbered interfaces, the relay agent borrows an IP address from another configured interface and dynamically adds or removes static host routes for DHCP clients as leases are assigned and released.

Supporting reference information

- DHCP relay agents are essential when DHCP clients and servers are not on the same subnet, as they enable message forwarding across networks.
- Relay agents specifically set the gateway IP address and can insert option 82 for added information in packets sent to the DHCP server.
- The Cisco IOS XE DHCP relay agent supports unnumbered interfaces, which helps conserve address space and network resources. Static host routes for DHCP clients on unnumbered interfaces are created and removed automatically based on the DHCP lease status.

Benefits of IP DHCP smart-relay

- Automatically assigning IP addresses at each remote site substantially reduces internet access costs. Purchasing static IP addresses is considerably more expensive than using automatically allocated IP addresses.
- Simplifies configuration, reduces operational overhead and costs related to device configuration, and makes deployment easier for nontechnical users.
- The DHCP server maintains configurations for several subnets, so an administrator needs to update only a single, central server when configuration parameters change.

Configure IP DHCP smart-relay agent using a CLI template

To forward UDP broadcasts to the DHCP server, configure helper addresses on the interface. If you have configured the secondary addresses on that interface and you want the router to step through each IP network when forwarding DHCP requests, use the `ip dhcp smart-relay` command.

If the smart relay agent forwarding is not configured, all requests are forwarded using only the primary IP address on the interface.

If the `ip dhcp smart-relay` command is configured, the relay agent counts the number of times that the client retries sending a request to the DHCP server when there is no DHCPOFFER message from the DHCP server. After three retries, the relay agent sets the gateway address to the secondary address. If the DHCP server still does not respond after three more retries, then the next secondary address is used as the gateway address.

For more information about using CLI templates, see *CLI Add-On Feature Templates* and *CLI Templates*.

Before you begin

- To configure the IP DHCP smart-relay feature, configure the IP helper address on desired interfaces using `ip helper-address` command. You can use the `service dhcp` command to enable the DHCP service or the `no service dhcp` command to disable it, depending on the requirement.
- The Cisco DHCP relay agent is enabled on an interface only when the `ip helper-address` command is configured. This command enables the DHCP broadcast to be forwarded to the configured DHCP server.

Procedure

Step 1 Enter SD-WAN configuration mode.

Example:

```
sdwan
```

Step 2 Enable DHCP server.

Example:

```
service dhcp
```

Step 3 In the SD-WAN configuration mode, configure an interface type such as **Gigabit Ethernet.**

Example:

```
interface Gigabitethernet0/0
```

Step 4 Enable the DHCP broadcast to be forwarded to the configured DHCP server.

Example:

```
 ip helper-address
```

Step 5 Configure the DHCP relay agent to switch the gateway address to a secondary address when there is no DHCPOFFER message from a DHCP server.

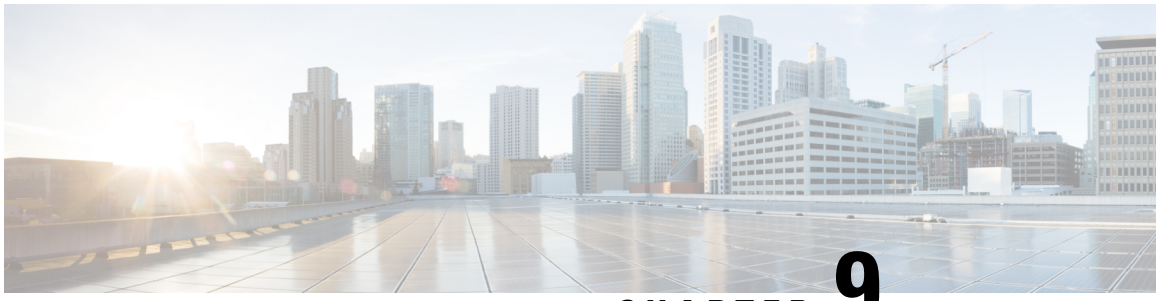
Example:

```
ip dhcp smart-relay
```

This is a DHCP smart-relay CLI configuration. In the example, the device forwards the DHCP broadcast received on Gigabitethernet interface 0/0 to the DHCP server (10.0.0.1), by inserting 192.168.255.254 in the gateway address field of the DHCP packet.

```
service dhcp
ip address 172.16.0.1 255.255.0.0
secondary ip address 192.168.255.254 255.255.0.0

interface GigabitEthernet0/0
ip helper-address 10.0.0.1
ip dhcp smart-relay
end
```



CHAPTER 9

Layer 2 VPN

Feature history for L2VPN

This table describes the developments of this feature, by release.

Table 13: Feature History

Feature Name	Release Information	Description
Layer 2 (L2) VPN	Cisco Catalyst SD-WAN Manager Release 20.14.1 Cisco IOS XE Catalyst SD-WAN Release 17.14.1a	The feature adds Layer 2 VPN support on the Cisco Catalyst SD-WAN overlay network. It allows you to configure Layer 2 point-to-point and point-to-multipoint connections within the Cisco Catalyst SD-WAN fabric.

Feature Name	Release Information	Description
Layer 2 (L2) VPN Multihoming and Hub-and-Spoke Support	Cisco Catalyst SD-WAN Manager Release 20.15.1 Cisco IOS XE Catalyst SD-WAN Release 17.15.1a	With this feature, you can configure Layer 2 VPN on multiple devices on the same site in an active-standby configuration. This feature also enables Layer 2 connections using an indirect path, such as a hub, for point-to-multipoint connections within the Cisco Catalyst SD-WAN fabric.

Layer 2 VPNs within the SD-WAN overlay network

Layer 2 VPN within the Cisco Catalyst SD-WAN overlay network is a network feature that

- enables Layer 2 connectivity across the SD-WAN fabric for legacy systems and non-IP applications,
- supports point-to-point (P2P) and point-to-multipoint (P2MP) L2VPN services with options for single homing, multihoming, and topologies including full mesh and hub-and-spoke, and
- provides MAC learning through OMP protocol (Control Plane), along with features such as ingress replication for broadcast, unknown-unicast, and multicast (BUM) traffic.

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.14.1a, the following L2VPN features are supported:

- Point-to-point L2VPN Service (P2P)
- Point-to-Multipoint L2VPN Service (P2MP)
- Single homing
- Flood and Learn in WAN and LAN
- Ingress replication for Broadcast, Unknown-unicast and Multicast (BUM)
- Full mesh topology only

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, the following L2VPN features are supported:

- Multihoming for P2P and P2MP
- Hub-and-spoke topology support for L2VPN services
- The MAC learning mode (previously the Flood and Learn in WAN and LAN) is changed to learning through OMP protocol (that is, Control Plane).

Note From Cisco IOS XE Catalyst SD-WAN Release 17.18.1a, you can adjust the TCP Maximum Segment Size (MSS) even for a TCP packet encapsulated in an MPLS label. You can set the TCP MSS per the Path Maximum Transmission Unit (PMTU) with 30 bytes to account for Layer 2 headers, such as Ethernet, VLAN tags and MPLS headers.

For IPv4, the TCP MSS is set per PMTU with 80 bytes for IPv4 and TCP headers and an additional 30 bytes for Layer 2 headers. For example, if the PMTU is 1438, the TCP MSS is set as 1328 (1438 - 80 - 30).

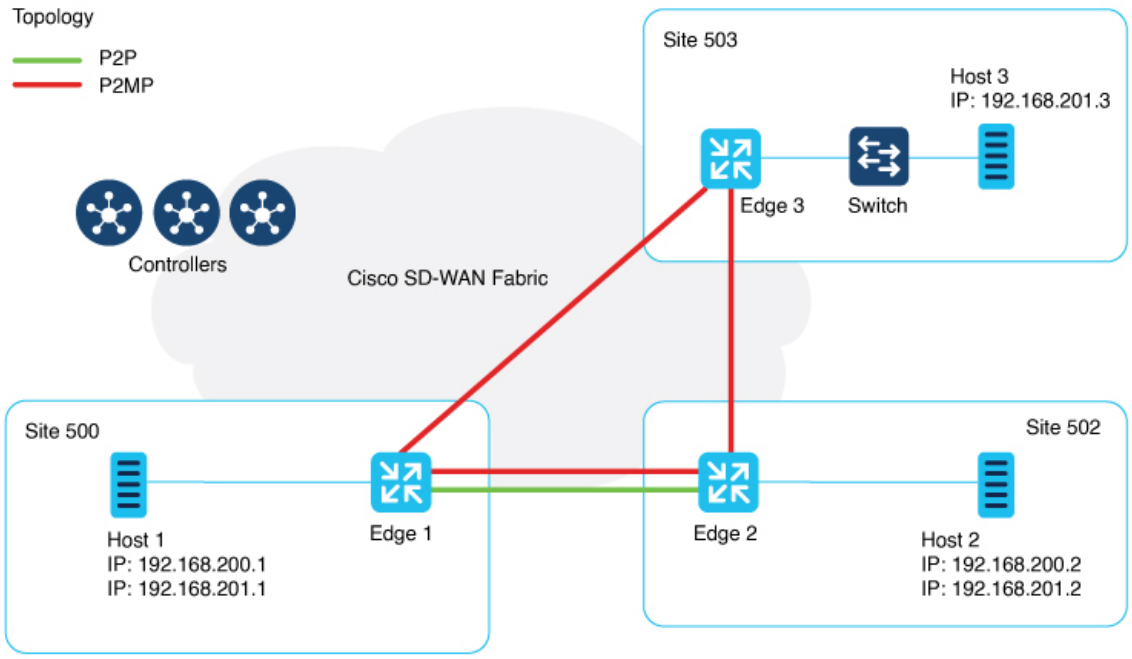
For IPv6, the TCP MSS is set per PMTU with 100 bytes for IPv6 and TCP headers and an additional 30 bytes for Layer 2 headers. For example, if the PMTU is 1438, the TCP MSS is set as 1308 (1438-100-30).

For more information about configuring TCP MSS, see Configure TCP MSS Using CLI

This change helps prevent drop in TCP sessions, improving overall network performance and reliability.

Network Topology for Layer 2 Connections

Figure 5: Topology

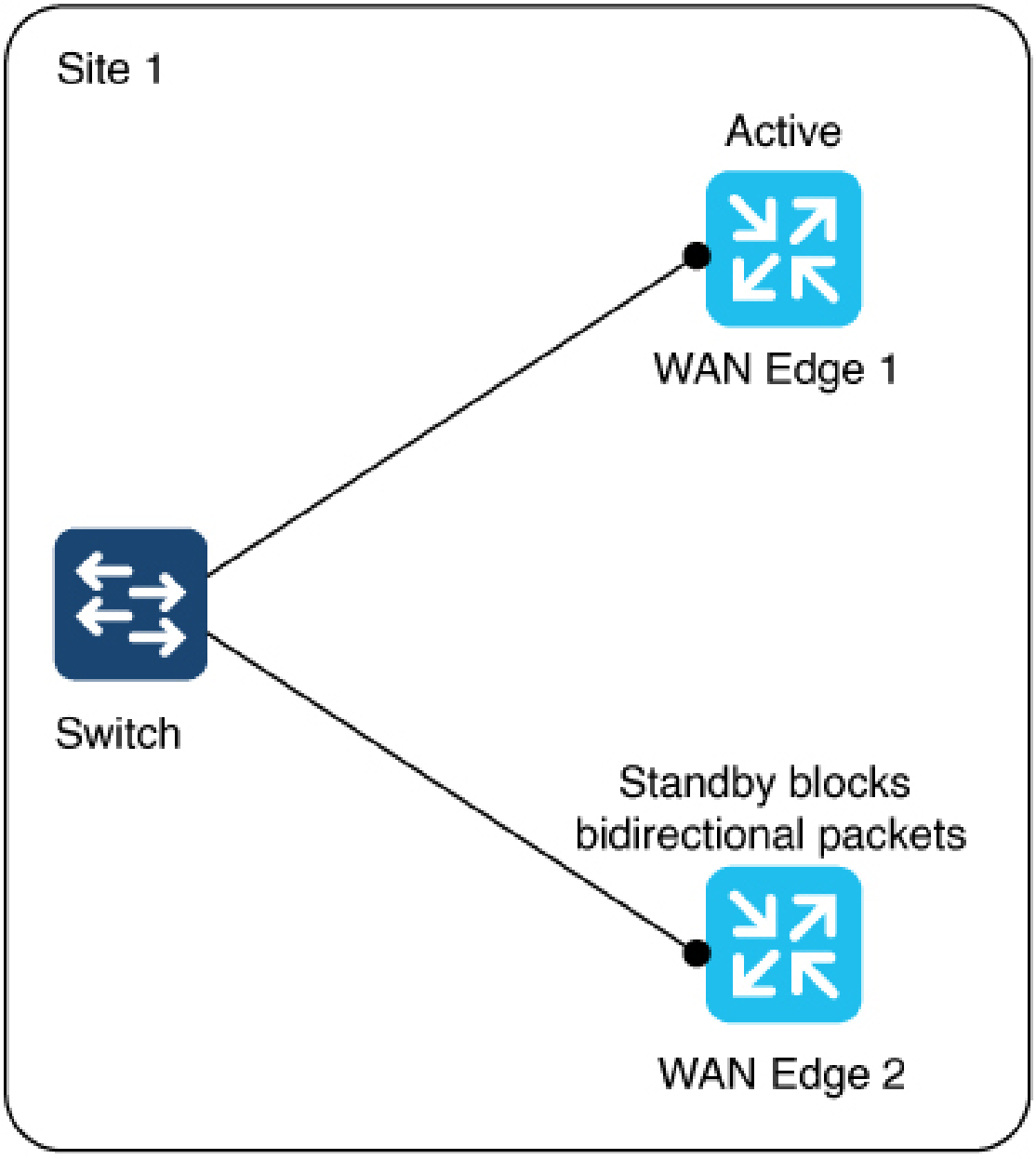


This illustration shows three sites and shows P2P (green line) and P2MP (red lines) connections between edge routers at the sites.

- Point-to-Point (P2P): Connects sites 500 and 502 with a dedicated Layer 2 VPN. The L2VPN connection between the two sites allows Host 1 and Host 2 to interact.
- Point-to-Multipoint (P2MP): Connects sites 500, 502, and 503 with Layer 2 VPN. Host 1 communicates with both Host 2 and Host 3 across a Layer 2 multipoint network.

The L2VPN connections use existing Cisco Catalyst SD-WAN tunnels.

Figure 6: Multihoming



The illustration shows two edge routers on the same site connected to a switch. For an (instance-id + vc), one router is active and the other is on standby. (instance-id +vc) maps to a bridge domain and a bridge-domain maps to a VLAN (or a VLAN range).

The router on standby blocks bidirectional traffic for that VLAN.

Multihoming supports L2VPN configuration on up to two edge devices on the same site, thereby providing redundancy for L2VPN service over SD-WAN.

Multihoming allows an active-standby scenario where one device is chosen as active and the other as standby. This provides automated failover. It determines which of the two edge devices should be active and which one should be on standby. When the OMP timer expires on the controller, it marks the L2VPN status route as stale, and notifies other edges.

Role determination for an Active and Standby device

The active and standby role between edge devices is automatically determined by with this algorithm: (SDWAN-Instance-ID + VC-ID) modular 2.

If the modular result is 0, the edge with lower system-ip is selected as the active device. The edge with the higher system-ip is selected as the standby device.

If the modular result is 1, the edge with higher system-ip is selected as the active device. The edge with the lower system-ip is selected as the standby device.

Example:

There are two WAN edge devices. WAN edge 1 has a system-ip of 172.16.255.10. WAN edge 2 has a system-ip of 172.16.255.11.

For sdwan-instance-id 100, vc-id 2, WAN edge 1 with the lower system-ip is selected as the active device. WAN edge 2 is the standby device.

For sdwan-instance-id 100, vc-id 1, WAN edge 2 with the higher system-ip is selected as the active device. WAN edge 1 is the standby device.

If a failure occurs on the service side of one of the edge devices, the controller is notified about a change to the L2VPN status route, and other edge routers can switchover traffic to the new active device.

Supported platforms for Layer 2 VPN

All Cisco IOS XE Catalyst SD-WAN devices.

Restrictions for Layer 2 VPN

Supported configuration CLI templates for Layer 2 VPN

Only CLI template or CLI add-on template configuration is supported for Layer 2 VPN.

LAN side interface limitation in single homing and multihoming

For both single homing and multihoming, only one LAN side interface is supported in a bridge-domain.

Point-to-Multipoint (P2MP) requirements for hub and spoke connectivity

P2P configuration between two spokes is not supported. In such cases, use P2MP instead of P2P.

Note P2P configuration between hub and spoke is supported.

Multihoming support only for dual homing

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, multihoming only supports dual homing.

Layer 2 VPN limitations for hub-and-spoke topology

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, hub-and-spoke topology is supported for Layer 2 VPN. It is limited by:

- No support for point-to-point Layer 2 VPN service between spokes.
- Support for up to 6000 spokes and 6000 sites within the same Layer 2 VPN in hub-and-spoke topology, and
- Support for only 256 sites within the same Layer 2 VPN in a non-hub-and-spoke design.

Methods to configure Layer 2 VPN using CLI template

Using these procedures, configure a Layer 2 VPN on a Cisco Catalyst SD-WAN overlay network.

- Configure a Layer 2 VPN on a Cisco IOS XE Catalyst SD-WAN device using CLI template, on page 62
- Configure a point-to-point Layer 2 VPN using CLI template, on page 63
- Configure a point-to-multipoint Layer 2 VPN using CLI template, on page 66
- Configure a Layer 2 VPN Switchport using CLI template, on page 70

Configure a Layer 2 VPN on a Cisco IOS XE Catalyst SD-WAN device using CLI template

Before you begin

For information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates.

Follow these steps to configure an L2VPN on a Cisco IOS XE Catalyst SD-WAN Device Using CLI Template

Procedure

Step 1 Configure an L2VPN instance for P2P and P2MP connections.

```
l2vpn sdwan instance instance-id point-to-point
l2vpn sdwan instance instance-id multipoint
```

The instance ID is a unique identifier for each L2VPN connection, and must not overlap or be shared with any Layer 3 VRFs in the SD-WAN fabric. For example, you cannot use L2VPN instance 10 and vrf definition 10.

Step 2 Configure a bridge-domain.

```
bridge-domain bridge-id
```

Step 3 Configure a Layer 2 interface on a Cisco IOS XE Catalyst SD-WAN device.

```
interface vlan-id
 service instance instance-id ethernet
  encapsulation dot1q vlan-id
  no shutdown
```

Note
A rewrite is used to modify the default VLAN tag. If you have not configured rewrite under service instance, dot1q must be the same at all sites participating in the Layer 2 network. The rewrite option in a Layer 2 configuration modifies the VLAN tags of packets as they ingress or egress an interface. To use the rewrite option, you need to configure Ethernet Virtual Connections (EVCs) on edge routers (Cisco ASR 1000 Series). For more information about configuring an EVC, see Configuring Ethernet Virtual Connections on a Cisco Router.

Configure a point-to-point Layer 2 VPN using CLI template

Before you begin

- You can use one L2VPN instance ID for one or more bridge domains. It must be the same at both ends of the circuit.

  To identify a particular bridge-domain, use Virtual Circuit (VC) ID. This ID is the identifier of the virtual circuit between the Cisco IOS XE Catalyst SD-WAN devices.

- To create a P2P pseudowire, L2VPN instance ID, and VC ID must be the same on different Cisco IOS XE Catalyst SD-WAN devices.
- Remote-site-id is only supported for P2P configuration.
- For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates for Cisco IOS XE Catalyst SD-WAN Devices

Note By default, CLI templates execute commands in Global Configuration mode.

Follow these steps to configure P2P L2VPN services between two sites (Site A and Site B) on the Cisco Catalyst SD-WAN overlay network.

Procedure

Step 1 Configure an edge router at Site A for point-to-point Layer 2 VPN using CLI template, on page 63

Step 2 Configure an edge router at Site B for point-to-point Layer 2 VPN using CLI temple, on page 64

Configure an edge router at Site A for point-to-point Layer 2 VPN using CLI template

Site A uses an edge router and connects the Ethernet interface to the L2 network that bridges to Site B.

Follow these steps to configure a an Edge Router at Site A for Point-to-Point Layer 2 VPN Using CLI Template

Procedure

Step 1 Define the L2VPN instance for point-to-point service:

```
l2vpn sdwan instance instance-id point-to-point
```

Step 2 Configure the Ethernet interface:

```
interface interface-name
 service instance instance-id ethernet
  encapsulation dot1q vlan-id
```

Step 3 Define the bridge domain and associate it with the interface and L2VPN instance:

```
bridge-domain bridge-id
 member vlan-name service-instance instance-id
 member sdwan instance  instance-id  remote-site remote-site-id vc-id  virtual-circuit-id single-homing
```

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, you can specify the homing type as dual homing to enable multihoming.

```
bridge-domain bridge-id
member vlan-name service-instance instance-id
member sdwan instance instance-id remote-site remote-site-id vc-id virtual-circuit-id
dual-homing
```

Example

The following configures Site A using Cisco Catalyst 8000V Edge Software to manage traffic through GigabitEthernet5, which is linked to the Layer 2 network that provides connectivity to Site B.

```
l2vpn sdwan instance 100 point-to-point

interface GigabitEthernet5
 service instance 100 ethernet
  encapsulation dot1q 2002
  !
bridge-domain 100
 member GigabitEthernet5 service-instance 100
 member sdwan-instance 100 remote-site 502 vc-id 100 single-homing
```

Configure an edge router at Site B for point-to-point Layer 2 VPN using CLI temple

Site B uses an edge router and Switchport Ethernet interface.

Follow these steps to configure an edge router at Site B for point-to-point Layer 2 VPN.

Procedure

Step 1 Define the L2VPN instance for point-to-point service.

```
l2vpn sdwan instance instance-id point-to-point
```

Step 2 Define the VLAN for the L2VPN.

```
vlan vlan-id
 name l2vpn
```

Step 3 Configure the VLAN interface.

```
interface interface-name
 service instance instance-id ethernet
  encapsulation dot1q vlan-id
  no shutdown
```

Step 4 Configure the Ethernet interface as an access port for VLAN.

```
interface interface-name
 switchport access vlan vlan-id
```

Step 5 Define the bridge-domain for site B and associate it with the VLAN and L2VPN instance.

```
bridge-domain bridge-id
 member vlan-name service-instance instance-id
 member sdwan instance  instance-id  remote-site remote-site-id vc-id  virtual-circuit-id single-homing
```

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, specify the homing type as dual homing to enable multihoming.

```
bridge-domain bridge-id
member vlan-name service-instance instance-id
member sdwan instance instance-id remote-site remote-site-id vc-id virtual-circuit-id
dual-homing
```

Example

The following configures Switchport GigabitEthernet 0/1/7 at Site B to connect to the interface with a Cisco ISR1100-8P device.

```
l2vpn sdwan instance 100 point-to-point
vlan 2002
 name L2vpn
interface Vlan2002
 service instance 100 ethernet
  encapsulation dot1q 2002
  no shutdown
  !
interface GigabitEthernet 0/1/7
```

```
 switchport access vlan 2002
bridge-domain 100
 member Vlan2002 service-instance 100
 member sdwan-instance 100 remote-site 500 vc-id 100 single-homing
```

After configuring the point-to-point L2VPN service on both sites, you can integrate these configuration blocks into your CLI Template or CLI Add-On Feature Template. This template can then be used to deploy the configuration across the relevant devices in the Cisco Catalyst SD-WAN fabric. Verify the connectivity and functionality of the L2VPN service following the deployment to confirm that the bridge between site A and site B is operational.

Configure a point-to-multipoint Layer 2 VPN using CLI template

Before you begin

- For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates.

  By default, CLI templates execute commands in global config mode.
- One L2VPN instance ID can be used by one or more bridge domains. VC ID is used to identify a particular bridge-domain.
- L2VPN instance ID and VC ID must be the same on different edge devices.

Follow these steps to configure P2MP Layer 2 VPN over Cisco Catalyst SD-WAN overlay, connecting a local Layer 2 network at site A to multiple remote sites (B and C). Site A uses Gigabit Ethernet interface to connect to the Layer 2 network for bridging.

Procedure

Step 1 Configure an edge router at sites A, B and C, on page 66

Step 2 Configure an edge router at Site B for point-to-point Layer 2 VPN using CLI temple, on page 64

Step 3 Configure an edge router at Site C for point-to-point Layer 2 VPN using CLI template, on page 68

Configure an edge router at sites A, B and C

Site A is using an edge router, where an Ethernet interface is connected to the Layer 2 network that bridges to Site B and Site C.

Follow these steps to configure an Edge Router at Sites A, B, and C.

Procedure

Step 1 Define the L2VPN instance for the multipoint service on the data center router:

`l2vpn sdwan instance` *`instance-id`* **`multipoint`**

Step 2 Configure the Ethernet interface on the data center router:

```
interface interface-name
service instance instance-id ethernet
encapsulation dot1q vlan-id
```

Step 3 Define the bridge-domain on the data center route and associate it with the interface and L2VPN instance:

```
bridge-domain bridge-id
member vlan-name service-instance instance-id
member sdwan instance instance-id remote-site remote-site-id vc-id virtual-circuit-id
single-homing
```

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, specify the homing type as dual homing to enable multihoming.

```
bridge-domain bridge-id
member vlan-name service-instance instance-id
member sdwan instance instance-id remote-site remote-site-id vc-id virtual-circuit-id
dual-homing
```

What to do next

See the topic, Configure an edge router at Site B for point-to-point Layer 2 VPN using CLI Temple.

Configure an edge router at Site B for point-to-point Layer 2 VPN using CLI temple

Site B uses an edge router and Switchport Ethernet interface.

Follow these steps to configure an edge router at Site B for point-to-point Layer 2 VPN.

Procedure

Step 1 Define the L2VPN instance for point-to-point service.

```
l2vpn sdwan instance instance-id point-to-point
```

Step 2 Define the VLAN for the L2VPN.

```
vlan vlan-id
 name l2vpn
```

Step 3 Configure the VLAN interface.

```
interface interface-name
 service instance instance-id ethernet
  encapsulation dot1q vlan-id
  no shutdown
```

Step 4 Configure the Ethernet interface as an access port for VLAN.

```
interface interface-name
 switchport access vlan vlan-id
```

Step 5 Define the bridge-domain for site B and associate it with the VLAN and L2VPN instance.

```
bridge-domain bridge-id
 member vlan-name service-instance instance-id
 member sdwan instance  instance-id  remote-site remote-site-id vc-id  virtual-circuit-id single-homing
```

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, specify the homing type as dual homing to enable multihoming.

```
bridge-domain bridge-id
member vlan-name service-instance instance-id
member sdwan instance instance-id remote-site remote-site-id vc-id virtual-circuit-id
dual-homing
```

Example

The following configures Switchport GigabitEthernet 0/1/7 at Site B to connect to the interface with a Cisco ISR1100-8P device.

```
l2vpn sdwan instance 100 point-to-point
vlan 2002
 name L2vpn
interface Vlan2002
 service instance 100 ethernet
  encapsulation dot1q 2002
  no shutdown
  !
interface GigabitEthernet 0/1/7
 switchport access vlan 2002
bridge-domain 100
 member Vlan2002 service-instance 100
 member sdwan-instance 100 remote-site 500 vc-id 100 single-homing
```

After configuring the point-to-point L2VPN service on both sites, you can integrate these configuration blocks into your CLI Template or CLI Add-On Feature Template. This template can then be used to deploy the configuration across the relevant devices in the Cisco Catalyst SD-WAN fabric. Verify the connectivity and functionality of the L2VPN service following the deployment to confirm that the bridge between site A and site B is operational.

Configure an edge router at Site C for point-to-point Layer 2 VPN using CLI template

Before you begin

Repeat the same steps as for branch router C, substituting the specific interface used on site B.

Follow these steps to configure an edge router at Site C for point-to-point Layer 2 VPN.

Procedure

Step 1 Define the L2VPN instance for multipoint service on the branch router:

```
l2vpn sdwan instance instance-id multipoint
```

Step 2 Define the VLAN for the L2VPN on the branch router:

```
vlan vlan-id
name L2vpn
```

Step 3 Configure the VLAN interface on the branch router:

```
interface interface-name
service instance instance-id ethernet
encapsulation dot1q vlan-id
no shutdown
```

Step 4 Configure the Ethernet interface on the branch router as an access port for VLAN:

```
interface interface-name
switchport access vlan vlan-id
```

Step 5 Define the bridge-domain on the branch router and associate it with the VLAN and L2VPN instance:

```
bridge-domain bridge-id
member vlan-name service-instance instance-id
member sdwan instance instance-id  vc-id virtual-circuit-id single-homing
```

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, specify the homing type as dual homing to enable multihoming.

```
bridge-domain bridge-id
member vlan-name service-instance instance-id
member sdwan instance instance-id remote-site remote-site-id vc-id virtual-circuit-id
dual-homing
```

Example

This section provides an example configuration for P2MP L2VPN service within the Cisco Catalyst SD-WAN overlay network, connecting a local Layer 2 network at site A to multiple remote sites (B and C). Site A uses GigabitEthernet6 interface to connect to the L2 network for bridging.

Verify the connectivity and functionality of the P2MP L2VPN service and ensure that all sites are correctly bridged.

Site A is using a Cisco Catalyst 8000V edge router, where GigabitEthernet6 is connected to the Layer 2 network that bridges to site B and site C.

```
l2vpn sdwan instance 200 multipoint

vlan 2001
 name L2MPvpn
```

```
interface Vlan2001
 service instance 200 ethernet
  encapsulation dot1q 2001
  no shutdown
  !
interface GigabitEthernet 0/1/6
 switchport access vlan 2001

bridge-domain 200
 member Vlan2001 service-instance 200
 member sdwan-instance 200 vc-id 200 single-homing
```

Configure branch router C:

Repeat the same steps as for branch router B, substituting the specific interface used on router 503. In this example, we have used the GigabitEthernet 0/1/6 interface.

```
l2vpn sdwan instance 200 multipoint

vlan 2001
 name L2MPvpn

interface Vlan2001
 service instance 200 ethernet
  encapsulation dot1q 2001
  no shutdown
  !
bridge-domain 200
 member Vlan2001 service-instance 200
 member sdwan-instance 200 vc-id 200 single-homing
```

Configure a Layer 2 VPN Switchport using CLI template

If your device such as Cisco ISR1121-8P or similar has embedded switchports and you want to use one of them for the L2VPN services, configure a VLAN interface first and then assign that VLAN to your switchport as described in this section.

To support a Layer 2 switchport, configure a service instance in the VLAN interface. In the VLAN interface, a packet always has the dot1q tag even when the Layer 2 switchport is configured with switchport mode access. Therefore, the dot1q tag is mandatory in the service instance of the VLAN interface.

This following section provides steps to configure a Layer 2 switchport for P2MP (applicable for devices with embedded switchports). You can also configure a Layer 2 switchport for P2P by updating the Layer 2 VPN instance command.

Site A is using an edge router, where the Ethernet interface is connected to the Layer 2 network that bridges to Site B and Site C.

Follow these steps to configure a Layer 2 VPN Switchport using CLI template.

Procedure

Step 1 Define the Layer 2 VPN instance for multipoint service on the branch routers:

`l2vpn sdwan instance` *`instance-id`* **`multipoint`**

Step 2 Define the VLAN for the Layer 2 VPN on the branch routers:

```
vlan vlan-id
name l2vpn
```

Step 3 Configure the Ethernet interface on the routers:

```
interface interface-name
```

Step 4 Set the switch port access VLAN and switchport mode to access to accept traffic only from the specified VLAN:

```
switchport access Vlan vlan-id
```

Step 5 Configure the VLAN interface on a router and disable the IP address assignment

```
interface interface-name
no ip address
service instance instance-id ethernet
encapsulation dot1q vlan-id
```

Step 6 Define the bridge-domain on the data center router and associate it with the interface and L2VPN instance:

```
bridge domain bridge-id
member vlan-name service-instance instance-id
member sdwan instance instance-id vc-id virtual-circuit-id single homing
```

The following configures a Layer 2 VPN Switchport to integrate a multipoint SD-WAN instance and bridge-domain. This configuration sets up GigabitEthernet0/1/2 as an access port for VLAN 201.

```
l2vpn sdwan instance 200 multipoint

interface GigabitEthernet0/1/2
 switchport access Vlan 201
 switchport mode access

interface Vlan201
 no ip address
 service instance 200 ethernet
  encapsulation dot1q 201
  !

bridge-domain 201
 member Vlan201 service-instance 200
 member sdwan-instance 200 vc-id 201 single-homing
```

Methods to verify Layer 2 VPN using CLI

To verify Layer 2 VPN using CLI, use these methods.

1. View a Layer 2 VPN status, on page 72
2. View L2VPN information learned through OMP route on a Cisco SD-WAN Controller, on page 72
3. View Bridge-domain information, on page 73

4. View Cisco Catalyst SD-WAN Flood List Information and Packet Counters in Data Plane, on page 74
5. View packet counters in data plane, on page 74

View a Layer 2 VPN status

To view the remote peer information, system IP, status, and related information, use the **show l2vpn sdwan [instance** *instance-id]***[vc-id** *vc-id]* command.

This is an example for a Cisco IOS XE Catalyst SD-WAN device.

```
Device# show l2vpn sdwan instance 13 vc-id 13
VC_ID: 13 Bridge-domain: 13
Local l2vpn status: UP
Local Pseudoports: GigabitEthernet7 service instance 13
```

View L2VPN information learned through OMP route on a Cisco SD-WAN Controller

To view the specific L2-route or path learned in the specific VPN and virtual circuit, use the **show sdwan omp l2-routes**[**vpn** *vpn-id*] [**vc-id** *vc-id*] command. If the **vpn** and **vc-id** are not included, the command shows Layer 2 routes learned through OMP from all VPNs across the Cisco Catalyst SD-WAN fabric.

This is a sample output from the **show omp l2-routes** command displaying Layer 2 routes learned through OMP for Cisco SD-WAN Controllers.

```
Device# show omp l2-routes | tab
C   -> chosen
I   -> installed
Red -> redistributed
Rej -> rejected
L   -> looped
R   -> resolved
S   -> stale
Ext -> extranet
Inv -> invalid
Stg -> staged
IA  -> On-demand inactive
U   -> TLOC unresolved


                                        REMOTE
                                ROUTE                   IP                  SITE
                PATH                    SITE
VPN     VC ID         ORIGINATOR        TYPE    MAC ADDRESS      ADDRESS  VPN TYPE    ID
FROM PEER         ID      LABEL     STATUS    ID
-------------------------------------------------------------------------------------------
12      12            172.16.255.15     vpn     0000.0000.0000   ::       p2p         500
172.16.255.15     66      1004      C,R       501

172.16.255.15     69      1004      C,R       501

172.16.255.20     1       1004      C,R       501
```

```
172.16.255.20    2      1004     C,R       501
12     12           172.16.255.27    vpn    0000.0000.0000   ::       p2p         501
172.16.255.20    1      1014     C,R       500

172.16.255.27    70     1014     C,R       500
13     13           172.16.255.15    vpn    0000.0000.0000   ::       multipoint  500
172.16.255.15    66     1006     C,R       -

172.16.255.15    69     1006     C,R       -

172.16.255.20    1      1006     C,R       -

172.16.255.20    2      1006     C,R       -
13     13           172.16.255.27    vpn    0000.0000.0000   ::       multipoint  501
172.16.255.20    1      1016     C,R       -

172.16.255.27    70     1016     C,R       -
13     13           172.16.255.32    vpn    0000.0000.0000   ::       multipoint  503
172.16.255.20    1      1007     C,R       -

172.16.255.32    71     1007     C,R       -
14     1            172.16.255.27    vpn    0000.0000.0000   ::       multipoint  501
172.16.255.20    1      1018     C,R       -

172.16.255.27    70     1018     C,R       -
15     1            172.16.255.15    vpn    0000.0000.0000   ::       p2p         500
172.16.255.15    66     1020     C,R       501

172.16.255.15    69     1020     C,R       501

172.16.255.20    1      1020     C,R       501

172.16.255.20    2      1020     C,R       501
15     1            172.16.255.27    vpn    0000.0000.0000   ::       p2p         501
172.16.255.20    1      1020     C,R       500

172.16.255.27    70     1020     C,R       500
```

View Bridge-domain information

To verify information related to bridge domains within the context of Forwarding Table Management Daemon (FTMD), use the **show platform software sdwan ftmd bridge-domain** command on a device.

This is a sample output from the **show platform software sdwan ftmd bridge-domain** command that displays information related to bridge domains within the context of Forwarding Table Management Daemon (FTMD).

```
Device# show platform software sdwan ftmd bridge-domain
L2vpn Bridge-domain 12 Table:
  sdwan efp dpidx: 4210708(0x404014)
  Label: 1004 lbl-nhop-id: 196611 (binosId=0xf830003f)
  Bum Label: 1005 bum-lbl-nhop-id: 196612 (binosId=0xf830004f)
  Remote Site Table(1 entries in total):
    remote-site-id: 501 sla-nhop-id: 29 (binosId=0xf80001df)

L2vpn Bridge-domain 13 Table:
  sdwan efp dpidx: 4210709(0x404015)
  Label: 1006 lbl-nhop-id: 196613 (binosId=0xf830005f)
```

```
  Bum Label: 1007 bum-lbl-nhop-id: 196614 (binosId=0xf830006f)
  Remote Site Table(2 entries in total):
    remote-site-id: 501 sla-nhop-id: 30 (binosId=0xf80001ef)
remote-site-id: 503 sla-nhop-id: 33 (binosId=0xf800021f)
```

View Cisco Catalyst SD-WAN Flood List Information and Packet Counters in Data Plane

To verify information related to Cisco Catalyst SD-WAN flood list information, use the **show platform hardware qfp active feature bridge-domain datapath** *bridge-domain-id* **sdwan-flood-list** command.

This is a sample output from the **show platform hardware qfp active feature bridge-domain datapath** *bridge-domain-id* **sdwan-flood-list** command that displays the Cisco Catalyst SD-WAN flood list information.

```
Device#show platform software sdwan ftmd bridge-domain
L2vpn Bridge-domain 12 Table:
  sdwan efp dpidx: 4210708(0x404014)
  Label: 1004 lbl-nhop-id: 196611 (binosId=0xf830003f)
  Bum Label: 1005 bum-lbl-nhop-id: 196612 (binosId=0xf830004f)
  Remote Site Table(1 entries in total):
    remote-site-id: 501 sla-nhop-id: 29 (binosId=0xf80001df)

L2vpn Bridge-domain 13 Table:
  sdwan efp dpidx: 4210709(0x404015)
  Label: 1006 lbl-nhop-id: 196613 (binosId=0xf830005f)
  Bum Label: 1007 bum-lbl-nhop-id: 196614 (binosId=0xf830006f)
  Remote Site Table(2 entries in total):
    remote-site-id: 501 sla-nhop-id: 30 (binosId=0xf80001ef)
remote-site-id: 503 sla-nhop-id: 33 (binosId=0xf800021f)
```

View packet counters in data plane

To verify information related to a QuantumFlow Processor (QFP) hardware module packet counters for a specific bridge domain within the data path, use the **show platform hardware qfp active feature bridge-domain datapath** *bridge-id* command.

This is a sample output from the **show platform hardware qfp active feature bridge-domain datapath** *bridge-id* command to display a QFP hardware module packet counters for a specific bridge domain within the data path.

```
Device# show platform hardware qfp active feature bridge-domain datapath 200
QFP L2BD Bridge Domain information



BD id                  : 200

State enabled          : Yes
```

```
Aging timeout (sec)    : 300

Aging active entry     : Yes

Max mac limit          : 65536

Unkwn mac limit flood  : Yes

mac_learn_enabled      : Yes

mac_learn_controled    : No

Unknown unicast olist  : Yes

otv_aed_enabled : No

otv_enabled : No

mcast_snooping_enabled : No

Feature : sdwan

SISF snoop protocols   : None

Sdwan instance id      : 200

Mac learned            : 0

BDI outer vtag         : 00000000

BDI inner vtag         : 00000000


Replication tree info:

  Global replication    : depth encode 0X1000001, (head 0XE4E90000)

  Split-horizon-group 0 : depth encode 00000000, (head 00000000)

  Split-horizon-group 1 : depth encode 00000000, (head 00000000)
Bridge Domain statistics


Total bridged               pkts : 0          bytes: 0

Total unknown unicast       pkts : 0          bytes: 0

Total broadcasted           pkts : 0          bytes: 0

Total to BDI                pkts : 0          bytes: 0

Total injected              pkts : 0          bytes: 0

Total mac-sec violation drop pkts : 0         bytes: 0

Total mac-sec move drop     pkts : 0          bytes: 0

Total mac-sec unknown drop  pkts : 0          bytes: 0

Total source filter drop    pkts : 0          bytes: 0

Total bfib policy drop      pkts : 0          bytes: 0
```

```
Total replication start drop pkts : 0          bytes: 0

Total recycle tail drop      pkts : 0          bytes: 0

Total static MAC move drop   pkts : 0          bytes: 0

Total BD disabled drop       pkts : 0          bytes: 0

Total STP state drop         pkts : 0          bytes: 0

Total UUF suppression drop   pkts : 0          bytes: 0

Total sisf ctrl punt         pkts : 0          bytes: 0

Total sisf ctrl drop         pkts : 0          bytes: 0

Total p2p lan to wan         pkts : 0          bytes: 0

Total p2p wan to lan         pkts : 0          bytes: 0
```

Monitor configured layer 2 VPN using CLI

This is a sample output from the **show l2vpn sdwan all** command. The following examples show the configuration and status information for Layer 2 VPN instances within a Cisco Catalyst SD-WAN overlay network. The output includes details for both point-to-point (P2P) and point-to-multipoint (P2MP) topologies.

Example 1: The example shows the L2VPN SD-WAN instance for instance 100 for point-to-point connectivity.

```
Device# show l2vpn sdwan all
L2VPN sdwan Instance : 100
VPN Type : point-to-point
  VC_ID: 100 Bridge-domain: 100 UP
    Local l2vpn status: UP
    Local Pseudoports: GigabitEthernet5 service instance 100
    Remote Site: 53
      System IP          status       up/down    color           encap    label  DF
      10.100.31.53       DOWN         00:15:04   public-internet ipsec    1023   N/A
```

Example 2: The example shows all the Layer 2 VPN SD-WAN instance for instance 200 for point-to-point connectivity.

```
Device# show l2vpn sdwan all
L2VPN sdwan Instance : 200
VPN Type : multipoint
IP Local-learning    : Disabled
Flooding Suppression : Disabled
  VC_ID: 200 Bridge-domain: 200 UP
    Local l2vpn status: UP
    Local Pseudoports: GigabitEthernet5 service instance 200
    Remote Site: 50
      System IP          status       up/down    color           encap    label  DF
      10.100.31.50       UP           00:04:14   public-internet ipsec    1008   N/A


    Remote Site: 53
      System IP          status       up/down    color           encap    label  DF
      10.100.31.53       UP           00:15:00   public-internet ipsec    1025   N/A
```

This is a sample output from the **show l2vpn sdwan instance** *instance-id* **vc-id** *vc-id***peers** command. The following examples show information about a specific Cisco Catalyst SD-WAN Layer 2 VPN instance (instance 200) and its associated virtual circuit (vc-id 200), including details about its peer connections.

`show l2vpn sdwan` **`instance`** *`instance-id`* **`vc-id`** *`vc-id`* **`peers`**

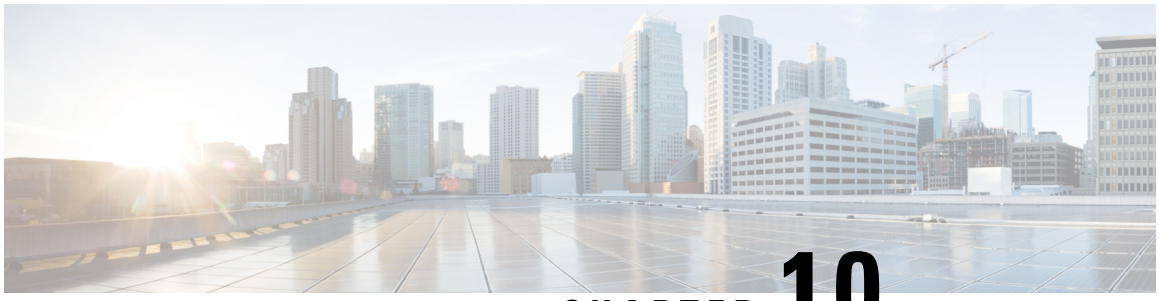
Example 1

```
Device1# show l2vpn sdwan instance 200 vc-id 200 peers
    Remote Site: 50   MACs Learn: 0
      System IP          status       up/down    color           encap    label  DF
      10.100.31.50       UP           00:19:54   public-internet ipsec    1008   N/A

    Remote Site: 53   MACs Learn: 0
      System IP          status       up/down    color           encap    label  DF
      10.100.31.53       UP           00:30:40   public-internet ipsec    1025   N/A
```

Example 2

```
Device# show l2vpn sdwan instance 200 vc-id 200 peers
     Remote Site: 1   MACs Learn: 0
      System IP          status       up/down    color           encap    label  DF
      10.100.31.1        UP           00:30:13   public-internet ipsec    1014   N/A
```



CHAPTER 10

IPv6 Functionality

Configure IPv6 functionality for an Interface or Subinterface using templates

Before you begin

Perform these steps to configure IPv6 functionality for an interface or subinterface template.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.

Step 2 Click **Feature Templates**, and click **Add Template** to select an appropriate device model.

Step 3 Select **Cisco VPN Interface Ethernet** from the list of templates.

Step 4 From **Basic Configuration**, click **IPv6** and configure these parameters.

Field	Description
Static	Selected by default because IPv6 addresses are static.
IPv6 Address	IPv6 address of the interface or subinterface.

Configure IPv6 functionality for an interface or subinterface using CLI commands

Procedure

Step 1 Create a CLI add-on profile or CLI add-on template.

Step 2 Configure according to this example.

```
interface GigabitEthernet1
  no shutdown
  ipv6 address 2001:DB8:1::1/64
  ipv6 enable
```

Configure IPv6 functionality for OMP using templates

Before you begin

Perform these steps to configure IPv6 functionality for OMP.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.

Step 2 Click **Feature Templates**, and click **Add Template** to select an appropriate device model.

Step 3 Select **Cisco OMP** from the list of templates.

Step 4 In the **Advertise** section, select **IPv6** and configure these parameters.

Field	Description
Connected	Click **Off** to disable advertising connected routes to OMP. By default, connected routes are advertised to OMP.
Static	Click **Off** to disable advertising static routes to OMP. By default static routes are advertised to OMP.
BGP	Click **Off** to advertise BGP routes to OMP. By default, BGP routes are not advertised to OMP.

Configure IPv6 functionality for OMP using CLI commands

Procedure

Step 1 Create a CLI add-on profile or CLI add-on template.

Step 2 Enable service VRF for IPv6, according to this example.

```
config-transaction
vrf definition 1
  rd 1:1
  address-family ipv6
```

Step 3 Enable OMP, according to this example.

OMP supports global IPv6 configuration. In addition, per VRF level configuration is allowed. Per VRF level configuration overrides global configuration.

```
config-transaction
sdwan
  omp
  !
  address-family ipv6
   advertise bgp
   advertise connected

   address-family ipv6 vrf 1
    advertise static
```

Step 4 Global configuration is the default configuration, so IPv6 is enabled by default for OMP. To disable IPv6 OMP route redistribution for a particular VRF, configure the redistribution protocol to **no**.

```
config-transaction
sdwan
  omp
  !
  address-family ipv6
   advertise bgp
   advertise connected
```

```
address-family ipv6 vrf 1
 no advertise connected
 no advertise static
 no advertise bgp
```

Configure IPv6 functionality for BGP using templates

Before you begin

Perform these steps to configure IPv6 functionality for BGP.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.

Step 2 Click **Feature Templates**, and click **Add Template** to select an appropriate device model.

Step 3 Select **Cisco BGP** from the list of templates.

Step 4 In the **Unicast Address Family** section, select **IPv6** and configure these parameters.

Tab	Field	Description
	Maximum Paths	Maximum number of parallel IBGP paths that can be installed into a route table to enable IBGP multipath load sharing. Range: 0 to 32
	Address Family	BGP IPv6 unicast address family.
RE-DISTRIBUTE		Click the **Redistribute** tab, and then click **Add New Redistribute**.
	Protocol	Select the protocols from which to redistribute routes into BGP, for all BGP sessions. Options are Connected, NAT, OMP, OSPF, and Static. At a minimum, select these: • For service-side BGP routing, select OMP. By default, OMP routes are not redistributed into BGP. • For transport-side BGP routing, select Connected, and then under Route Policy, specify a route policy that has BGP advertise the loopback interface address to its neighbors.
	Route Policy	Name of the route policy to apply to redistributed routes.
		Click **Add** to save the redistribution information.
NETWORK		Click the **Network** tab, and then click **Add New Network**.
	Network Prefix	Network prefix in the format of prefix/length, for BGP to advertise.
		Click **Add** to save the network prefix.

Tab	Field	Description
AGGREGATE ADDRESS		Click the **Aggregate Address** tab, and then click **Add New Aggregate Address**.
	Aggregate Prefix	Prefix of the addresses to aggregate for all BGP sessions, in the format prefix/length.
	AS Set Path	Click **On** to generate set path information for the aggregated prefixes.
	Summary Only	Click **On** to filter out more specific routes from BGP updates.
		Click **Add** to save the aggregate address.

Step 5 In the **Neighbor** section, select **IPv6**, create a new neighbor or edit an existing one, and then configure these parameters.

Field	Description
IPv6 Address*	IPv6 address of the BGP neighbor.
Description	Description of the BGP neighbor.
Remote AS*	AS number of the remote BGP peer.
Address Family	Select **Global** from the drop-down list, click **On** and select the address family. Enter the address family information.
Shutdown	To shut down a BGP neighbor when you push the template, select **Global** from the drop-down list and then click **Yes**. Default: Off

Configure IPv6 functionality for BGP using CLI commands

Procedure

Step 1 Create a CLI add-on profile or CLI add-on template.

Step 2 Configure according to this example.

```
config-transaction
router bgp 1
 bgp log-neighbor-changes
  address-family ipv6 unicast vrf 1
  neighbor 2001:DB8:19::1 remote-as 2
  neighbor 2001:DB8:19::1 activate
  neighbor 2001:DB8:19::1 advertisement-interval 1
  neighbor 2001:DB8:19::1 password cisco
  redistribute omp
```

```
redistribute static
exit-address-family
```

Configure IPv6 functionality for VRRP using templates

Before you begin

Perform these steps to configure IPv6 functionality for Virtual Router Redundancy Protocol (VRRP).

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.

Step 2 Click **Feature Templates**, and click **Add Template** to select an appropriate device model.

Step 3 Select **Cisco VPN Interface Ethernet** from the list of templates.

Step 4 In the **VRRP** section, select **IPv6**.

Step 5 Click **New VRRP** and configure these parameters.

Field	Description
Group ID	Virtual router ID, which represents a group of routers. Range: 1 to 255
Priority	Priority level of the router within a VRRP group. Range: 1 to 254 Default: 100
Timer	Not used.
Track OMP	Select **On** to track the Overlay Management Protocol (OMP) session running on the WAN connection when determining the primary VRRP virtual router. Default: Off
Track Prefix List	Value to track a list of IPv6 remote prefixes. This value is an alphanumeric string that is configured under Policy.
Link Local IPv6 Address	Virtual link local IPv6 address, which represents the link local address of the group. The address should be in standard link local address format. For example, FE80::AB8.

Field	Description
Global IPv6 Address	Virtual global unicast IPv6 address, which represents the global address of the group. The address should be an IPv6 global prefix address that has the same mask as the interface forwarding address on which the VRRP group is configured. Example: 2001::2/124 Maximum: 3 global IPv6 addresses

Configure IPv6 functionality for VRRP using CLI commands

Procedure

Step 1 Create a CLI add-on profile or CLI add-on template.

Step 2 Configure VRRP, according to this example.

```
config-transaction
interface GigabitEthernet1

 vrrp 10 address-family ipv6
   priority 20
   track omp shutdown
   address FE80::10:100:1 primary
   address 2001:10:100::1/64

Prefix-list tracking
track 1 ipv6 route 1:1::1/128
 reachability
 ipv6 vrf 1

 track 2 ipv6 route 2:2::2/128
 reachability
 ipv6 vrf 2

 track 20 list boolean or
  object 1
  object 2

vrrp 10 address-family ipv6
  track 20 shutdown
```

Configure IPv6 functionality for SNMP using templates

Before you begin

Configure the SNMP community and trap target group.

Perform these steps to configure IPv6 functionality for SNMP.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.

Step 2 Click **Feature Templates**, and click **Add Template** to select an appropriate device model.

Step 3 Select **Cisco SNMP** from the list of templates.

Step 4 In the **Trap** section, create or edit an SNMP trap target.

Note the prerequisites for this procedure.

Step 5 Configure these parameters.

Field	Description
VPN ID	Number of the VPN to use to reach the trap server. Range: 0 to 65530
IP Address	IP address of the SNMP server.
UDP Port	UDP port number for connecting to the SNMP server. Range: 1 to 65535
Trap Group Name	Name of a trap group configured in the **Group** tab.
User Name	Name of a community configured in the **Community** tab.
Source Interface	Interface to use to send traps to the SNMP server that is receiving the trap information.

Configure IPv6 functionality for SNMP using CLI commands

Procedure

Step 1 Create a CLI add-on profile or CLI add-on template.

Step 2 Configure SNMP, according to these examples.

This example permits any SNMP to access all objects with read-only permission using the community string named public. The device also sends Border Gateway Protocol (BGP) traps IPv6 host 3ffe:b00:c18:1::3/127 using SNMP v1.The community string named public is sent with the traps.

```
Device# config-transaction
Device(config)# snmp-server community public
Device(config)# snmp-server enable traps bgp
Device(config)# snmp-server host 3ffe:b00:c18:1::3/127 public
```

In this example, the SNMP context A is associated with the views in SNMPv2c group GROUP1 and the IPv6 named access list public2.

```
Device# config-transaction
Device(config)# snmp-server context A
Device(config)# snmp mib community-map commA context A target-list comm AVpn
Device(config)# snmp mib target list commAVpn vrf CustomerA
Device(config)# snmp-server view viewA ciscoPingMIB included
Device(config)# snmp-server view viewA ipForward included
Device(config)# snmp-server group GROUP1 v2c contextA read viewA write viewA notify access ipv6
public2
```

This example configures the IPv6 host as the notification server.

```
Device> enable
Device# config-transaction
Device(config)# snmp-server community mgr view restricted rw ipv6 mgr2
Device(config)# snmp-server engineID remote 3ffe:b00:c18:1::3/127 remotev6
Device(config)# snmp-server group publicv2c access ipv6 public2
Device(config)# snmp-server hosthost1.com2c vrf trap-vrf mgr
Device(config)# snmp-server user user1 bldg1 remote3ffe:b00:c18:1::3/127 v2c access ipv6 public2
Device(config)# snmp-server enable traps bgp
Device(config)# exit
```

Configure IPv6 functionality for a DHCP relay agent using templates

Before you begin

Perform these steps to configure IPv6 functionality for a DHCP relay agent.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.

Step 2 Click **Feature Templates**, and click **Add Template** to select an appropriate device model.

Step 3 Select **Cisco VPN Interface Ethernet** from the list of templates.

Step 4 In the **Basic Configuration** section, select **IPv6**.

Step 5 In the **DHCP Helper** area, click **Add** and configure these parameters.

Field	Description
DHCPv6 Helper #	IP address of the DHCP helper
DHCPv6 Helper VPN	VPN ID of the VPN source interface for the DHCP helper.

Configure IPv6 functionality for a DHCP relay agent using CLI commands

Procedure

Step 1 Create a CLI add-on profile or CLI add-on template.

Step 2 Configure a DHCP relay agent, according to this example.

```
device-configuration
interface GigabitEthernet8
 vrf forwarding 2
 no ip address
 ipv6 address 2001:A14:99::F/64
 ipv6 dhcp relay destination vrf 1 2001:A14:19::12 GigabitEthernet2
```

Configure IPv6 functionality for ACL and QoS using templates

Before you begin

Perform these steps to configure IPv6 functionality for access control lists (ACL) and quality of service (QoS).

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.

Step 2 Click **Feature Templates**, and click **Add Template** to select an appropriate device model.

Step 3 Select **Cisco VPN Interface Ethernet** from the list of templates.

Step 4 In the **ACL/QoS** section, configure these parameters.

Parameter Name	Description
Ingress ACL – IPv6	Click **On** to enable the IPv6 ingress access list.
IPv6 Ingress Access List	Enter the name of the IPv6 ingress access list.
Egress ACL – IPv6	Click **On** to enable the IPv6 egress access list.
IPv6 Egress Access List	Enter the name of the IPv6 egress access list.

Configure IPv6 functionality for ACL and QoS using CLI commands

Before you begin

Perform these steps to configure IPv6 functionality for access control lists (ACL) and quality of service (QoS).

Procedure

Step 1 Create a CLI add-on profile or CLI add-on template.

Step 2 Configure ACL according to this example.

```
Device(config)# policy
Device(config-policy)# ipv6
Device(config-ipv6)# access-list ipv6_acl
Device(config-access-list-ipv6_acl)#    sequence 11
Device(config-sequence-11)#     match
Device(config-match)#       source-ip 2001:380:1::64/128
Device(config-match)#      destination-ip 2001:3c0:1::64/128
Device(config-match)#      source-port      4000
Device(config-match)#      destination-port 3000
Device(config-match)#      traffic-class    6
Device(config-match)#      next-header      6
Device(config-match)#      packet-length    1000
Device(config-match)#     action accept
Device(config-action)#

Device(config)# sdwan interface GigabitEthernet6 ipv6 access-list ipv6_acl in
Device(config-interface-GigabitEthernet6)#
Device(config-interface-GigabitEthernet6)#

Device(config)# policy lists data-ipv6-prefix-list source_ipv6_list
Device(config-data-ipv6-prefix-list-source_ipv6_list)# ipv6-prefix 2001:380:1::/64

Device(config)# policy
Device(config-policy)# ipv6
Device(config-ipv6)# access-list ipv_ipv6_prefix
Device(config-access-list-ipv_ipv6_prefix)# sequence 11
Device(config-sequence-11)#     match
Device(config-match)#       source-data-prefix-list data-ipv6-prefix-list
Device(config-match)#       destination-data-prefix-list source_ipv6_list
Device(config-match)#      destination-ip 2001:3c0:1::64/128
Device(config-match)#      source-port      4000
Device(config-match)#      destination-port 3000
Device(config-match)#      traffic-class    6
Device(config-match)#      next-header      6
Device(config-match)#      packet-length    1000
Device(config-match)#     !
Device(config-match)#     action accept
```

Step 3 Configure QoS according to this example.

```
class-map match-any class0
match qos-group 0
class-map match-any class1
match qos-group 1
!
```

```
policy-map qos_map_for_data_policy
class class0
  bandwidth percent 10
  random-detect
class class1
  bandwidth percent 10
  random-detect

policy
no app-visibility
class-map
  class class0 queue 0
  class class1 queue 1
!
ipv6
  access-list fwd_class_data_policy
   sequence 5
    match
     traffic-class 0
    !
    action accept
     count fwd_class_data_policycnt_5
     class class0
    !
   sequence 6
    match
     traffic-class 1
    !
    action accept
     count fwd_class_data_policycnt_6
     class class1
!
default-action drop
```

Configure IPv6 functionality for a logging host using templates

Before you begin

Perform these steps to configure IPv6 functionality for a logging host.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.

Step 2 Click **Feature Templates**, and click **Add Template** to select an appropriate device model.

Step 3 Select **Cisco Logging** from the list of templates.

Step 4 From **Server**, click **IPv6** and configure these parameters.

Field	Description
IPv6 Hostname/IPv6 Address	Host name or IP address of the server to direct the logging information.
VPN ID	VPN ID of the VPN source interface.

Field	Description
Source Interface	Name of the source interface.
Priority	Maximum severity of messages that are logged.

Configure IPv6 functionality for a logging host using CLI commands

Before you begin

Perform these steps to configure IPv6 functionality for a logging host.

Procedure

Step 1 Create a CLI add-on profile or CLI add-on template.

Step 2 Configure a logging host, according to this example.

```
config-transaction
Device(config)# logging host ipv6
AAAA:BBBB:CCCC:DDDD::FFFF
```

Note
Creating and deleting the logging host configurations in same transaction causes unexpected behavior. For example, deleting **logging host** *ipv6-address* and creating **logging host** *ipv6-address* **vrf** *vrf-name* configuration in same transaction causes both configurations to disappear from the device. Send the two requests in separate transactions.

Configure IPv6 functionality for a prefix list using templates

Before you begin

Perform these steps to configure IPv6 functionality for a prefix list.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.

Step 2 From the **Custom Options** drop-down list, select **Lists**. You can make this selection for a centralized policy or a localized policy.

Step 3 Select **Prefix** from the list on the left and then select **New Prefix List**.

Step 4 Click **IPv6** and enter the IPv6 address in **Add Prefix**.

Configure IPv6 functionality for a prefix list using CLI commands

Before you begin

Perform these steps to configure IPv6 functionality for a prefix list.

Procedure

Step 1 Create a CLI add-on profile or CLI add-on template.

Step 2 Configure a prefix list, according to this example.

```
config-transaction
Device(config)# policy
Device(config-policy)# ipv6
Device(config-ipv6)# access-list ipv6_acl
Device(config-access-list-ipv6_acl)#  sequence 11
Device(config-sequence-11)#  match
Device(config-match)# source-ip 2001:DB8:1::64/128
Device(config-match)# destination-ip 2001:DB8:1::64/128
```

Configure IPv6 functionality for a data prefix using templates

Before you begin

Perform these steps to configure IPv6 functionality for a data prefix.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.

Step 2 From the **Custom Options** drop-down list, select **Lists**. You can make this selection for a centralized policy or a localized policy.

Step 3 Select **Data Prefix** from the list on the left and then select **New Data Prefix List**.

Step 4 From **Internet Protocol**, click **IPv6** and enter the IPv6 address in **Add Prefix**.

Configure IPv6 functionality for a data prefix using CLI commands

Before you begin

Perform these steps to configure IPv6 functionality for a data prefix.

Procedure

Step 1 Create a CLI add-on profile or CLI add-on template.

Step 2 Configure a data prefix, according to this example.

```
Device(config)# policy lists data-ipv6-prefix-list source_ipv6_list
Device(config-data-ipv6-prefix-list-source_ipv6_list)# ipv6-prefix 2001:DB8:1::/64
```

Configure IPv6 functionality for a centralized policy using templates

Before you begin

Perform these steps to configure IPv6 functionality for a centralized policy.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.

Step 2 From the **Custom Options** drop-down list, select **Traffic Policy** under **Centralized Policy**.

Step 3 Select **Traffic Data**.

Step 4 Select **Add Policy** and click **Create New**.

Step 5 Click **Sequence Type** and then select **Traffic Engineering**.

Step 6 Click **Sequence Rule**.

Step 7 From the **Protocol** drop-down list, select **IPv6** to apply the policy only to IPv6 address families, or select **Both** to apply the policy IPv4 and IPv6 address families.

Step 8 Click **Sequence Type** and then select **QoS**.

Step 9 Click **Sequence Rule**.

Step 10 From the **Protocol** drop-down list, click **IPv6** to apply the policy only to IPv6 address families, or select **Both** to apply the policy IPv4 and IPv6 address families.

Configure IPv6 functionality for a centralized policy using CLI commands

Before you begin

Perform these steps to configure IPv6 functionality for a centralized policy.

Procedure

Step 1 Create a CLI add-on profile or CLI add-on template.

Step 2 Configure a IPv6 in a centralized policy, according to this example.

```
config-transaction
(config)# policy
(config-policy)# lists ipv6-prefix-list foo ipv6-prefix 1::1/64
                  ipv6-prefix-list ipv6-1
                  ipv6-prefix 1::1/128
```

Configure IPv6 functionality for a localized policy using templates

Before you begin

Perform these steps to configure IPv6 functionality for a localized policy.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.

Step 2 From the **Custom Options** drop-down list, select **Access Control Lists** under **Localized Policy**.

Step 3 Click **Add Access Control List Policy** and choose **Add IPv6 ACL Policy**. The policy you create applies only to IPv6 address families.

Configure IPv6 functionality for a localized policy using CLI commands

Before you begin

Perform these steps to configure IPv6 functionality for a localized policy.

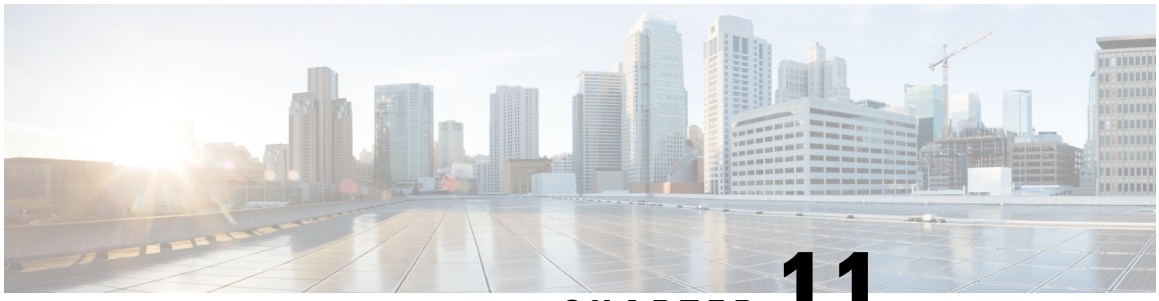
Procedure

Step 1 Create a CLI add-on profile or CLI add-on template.

Step 2 Configure a IPv6 in a localized policy, according to this example.

The example matches IPv6 routes that have addresses specified by the prefix list called marketing.

```
config-transaction
Device(config)# route-map name
Device(config-route-map)# match ipv6 address prefix-list marketing
```



CHAPTER 11

IPv6 in a Dual Stack Environment

Dual stack environment with IPv6 as the preferred address family

This table shows the history of the feature.

Table 14: Feature History

Feature Name	Release Information	Description
IPv6 as Preferred Address Family in a Dual Stack Environment	Cisco IOS XE Catalyst SD-WAN Release 17.10.1a Cisco Catalyst SD-WAN Control Components Release 20.10.1	You can select IPv6 as the preferred address family for control and data connections in a dual stack network environment. For Cisco SD-WAN Manager and Cisco Catalyst SD-WAN Controller, configure IPv6 as the preferred address family by using the feature template or the CLI template. For Cisco IOS XE Catalyst SD-WAN devices, configure IPv6 as the preferred address family using the configuration groups, the Quick Connect workflow, or by CLI commands executed through a CLI add-on profile or template.

Dual stack environment with IPv6 as the preferred address family

A dual stack environment is a network that supports both IPv4 and IPv6 addressing, and where both can be used simultaneously. In a dual stack Cisco Catalyst SD-WAN environment, you can select a preferred address family, either IPv4 or IPv6, for establishing control and data connections.

Control connections

You can select a preferred address family, IPv4 or IPv6, to establish control and data connections in a dual stack network environment. Use the **Dual Stack IPv6 Default** drop-down list in Cisco SD-WAN Manager to set IPv6 or IPv4.

- **Dual Stack IPv6 Default**: **True**:

  The device establishes IPv6 control connections with SD-WAN Manager and the SD-WAN Controller that the device is connected to.

- **Dual Stack IPv6 Default**: **False**:

  The device establishes IPv4 control connections with SD-WAN Manager and the SD-WAN Controller that the device is connected to.

Data connections

Data connections or Bidirectional Forwarding Detection (BFD) sessions are established based on the IPv6 option set in local, remote devices. In a dual stack network environment, if you choose **True** for a local or remote device, the BFD session is an IPv6 connection. Otherwise, it is IPv4.

- **Dual Stack IPv6 Default**: **True**:

  Establishes IPv6 BFD sessions.

- **Dual Stack IPv6 Default**: **False**:

  Establishes IPv4 BFD sessions.

SD-WAN Validator

The **Dual Stack IPv6 Default** drop-down list options applies to devices, SD-WAN Manager, and SD-WAN Controller, but not to SD-WAN Validator.

The connections from SD-WAN Manager, SD-WAN Controller, and devices to the SD-WAN Validator are always dual (IPv4 and IPv6) in a dual stack network environment, regardless of the **Dual Stack IPv6 Default** setting.

NAT44 and NAT66

You can configure an IPv6 connection on devices that are at sites behind NAT44 and NAT66.

Benefits of setting IPv6 as the preferred address family

You have the option to migrate from IPv4 to IPv6, which allows you to have more IP addresses compared to IPv4. With IPv6, there can be no depletion of IP addresses.

Methods for configuring a dual stack environment with IPv6 as the preferred address family

These are methods of configuring devices and SD-WAN Control Components for IPv6 connections.

Configuring devices for IPv6 connectivity

- Configure IPv6 as the preferred address family for devices, using a configuration group, on page 99
- Configure IPv6 as the preferred address family for devices, using the Quick Connect workflow, on page 100
- Configure IPv6 as the preferred address family for devices, using CLI commands, on page 101

Configuring SD-WAN Control Components for IPv6 connectivity

- Configure IPv6 as the preferred address family for SD-WAN Manager and SD-WAN Controller, using CLI commands, on page 101
- Configure IPv6 as the preferred address family for SD-WAN Manager and SD-WAN Controller, using templates, on page 102

Configure IPv6 as the preferred address family for devices, using a configuration group

Before you begin

Follow these steps to configure IPv6 as the preferred address family for Cisco IOS XE Catalyst SD-WAN devices in a dual stack environment, using a configuration group.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration Groups**.

Step 2 Click **Configuration Groups**.

Step 3 Create a new configuration group or open an existing one to display the full details of the configuration group.

Step 4 In the **Deployment** area, view the devices associated with the configuration group. You can add devices if desired.

Step 5 Click **Deploy** to deploy the configuration to the associated devices.

Step 6 In **Process Overview**, click **Next**.

Step 7 On the **Selected Devices to Deploy** page, select or deselect devices and click **Next**.

Step 8 For **Dual Stack IPv6 Default**, select **True** to set IPv6 as the default connection type, and click **Next**.

Here's how **Dual Stack IPv6 Default** affects control connections and BFD sessions:

- **True**: Devices establish an IPv6 control connection with SD-WAN Manager and the SD-WAN Controller they are connected to. BFD sessions use IPv6.
- **False**: Devices establish an IPv4 control connection with SD-WAN Manager and the SD-WAN Controller they are connected to. BFD sessions use IPv4.

The connections from devices to SD-WAN Validator are always dual (IPv4 and IPv6) in a dual IP stack environment, regardless of how**Dual Stack IPv6 Default** is configured.

Step 9 On the summary page, click **Deploy**.

What to do next

For more information on using configuration groups, see Configuration Groups and Feature Profiles.

Configure IPv6 as the preferred address family for devices, using the Quick Connect workflow

For information about the workflow, see Quick Connect Workflow.

Before you begin

Follow these steps to configure IPv6 as the preferred address family for Cisco IOS XE Catalyst SD-WAN devices in a dual stack environment, using the Quick Connect workflow.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Workflows** > **Quick Connect**.

Step 2 On the **Process Overview** page, click **Next**.

Step 3 Select an option to sync your devices, and click **Next**.

Step 4 On the **Selected devices to bring up** page, select devices, and click **Next**.

Step 5 In the **Dual Stack IPv6 Default** drop-down list, select **True** to set IPv6 as a default connection. Click **Apply**, and then click **Next**.

Here's how **Dual Stack IPv6 Default** affects control connections and BFD sessions:

- **True**: Devices establish an IPv6 control connection with SD-WAN Manager and the SD-WAN Controller they are connected to. BFD sessions use IPv6.
- **False**: Devices establish an IPv4 control connection with SD-WAN Manager and the SD-WAN Controller they are connected to. BFD sessions use IPv4.

The connections from devices to SD-WAN Validator are always dual (IPv4 and IPv6) in a dual IP stack environment, regardless of how**Dual Stack IPv6 Default** is configured.

Step 6 On the **Summary** page, click **Deploy**.

Configure IPv6 as the preferred address family for devices, using CLI commands

Before you begin

Follow these steps to configure IPv6 as the preferred address family for Cisco IOS XE Catalyst SD-WAN devices in a dual stack environment, using CLI commands.

Procedure

Step 1 Create a CLI add-on profile or CLI add-on template.

Step 2 Enable IPv6 on the tunnel interface.

```
interface tunnel1
no shutdown
ipv6 enable
```

Step 3 Enable IPv6.

```
system
ipv6-strict-control true
```

This example configures IPv6 as the preferred address family for devices.

```
interface Tunnel1
 no shutdown
 ip unnumbered GigabitEthernet1
 tunnel source GigabitEthernet1
 tunnel mode sdwan
 ipv6 enable
exit

system
gps-location latitude 32.0
gps-location longitude -100.0
system-ip 10.16.255.14
domain-id 1
site-id 400
ipv6-strict-control true
admin-tech-on-failure
organization-name "Cisco"
vbond vbond
```

Configure IPv6 as the preferred address family for SD-WAN Manager and SD-WAN Controller, using CLI commands

Before you begin

Follow these steps to configure IPv6 as the preferred address family for SD-WAN Manager and SD-WAN Controller in a dual stack environment, using CLI commands.

Procedure

Step 1 Create a CLI add-on profile or CLI add-on template.

Step 2 Enable IPv6.

```
system
ipv6-strict-control true
```

This example configures IPv6 as the preferred address family for SD-WAN Manager and SD-WAN Controllers.

```
system
host-name vm9
system-ip 10.16.255.19
site-id 400
ipv6-strict-control true
port-offset 9
no daemon-restart
admin-tech-on-failure
no vrrp-advt-with-phymac
organization-name "Cisco"
vbond vbond
```

Configure IPv6 as the preferred address family for SD-WAN Manager and SD-WAN Controller, using templates

Before you begin

Follow these steps to configure IPv6 as the preferred address family for SD-WAN Manager and SD-WAN Controller in a dual stack environment, using templates.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.

Step 2 Click **Feature Templates**, and click **Add Template**.

Step 3 In the list of devices, select **Controller**.

Step 4 Select **System** from the list of templates.

Step 5 In the **Basic Information** section, for the **Dual Stack IPv6 Default** field, click **On** to enable.

This sets IPv6 as a default connection.

Here's how **Dual Stack IPv6 Default** affects connections:

- **On**: SD-WAN Manager and SD-WAN Controllers establish an IPv6 control connection with other SD-WAN Manager and SD-WAN Controller instances.
- **Off**: SD-WAN Manager and SD-WAN Controllers establish IPv4 control connections.

The connections from SD-WAN Manager and SD-WAN Controller instances to SD-WAN Validator are always dual (IPv4 and IPv6) in a dual IP stack environment, regardless of how**Dual Stack IPv6 Default** is configured.

Step 6 Click **Save**.

Monitor the use of IPv6 as the preferred address family in a dual stack environment, in SD-WAN Manager

After you configure IPv6 as the preferred address family for connections, the BFD connections will be up and running in SD-WAN Manager.

Before you begin

Follow these steps to view the BFD connections in SD-WAN Manager.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Monitor** > **Devices**.

Step 2 Verify the status of the connection in the **BFD** column.

Monitoring IPv6 as the preferred address family in a dual stack environment

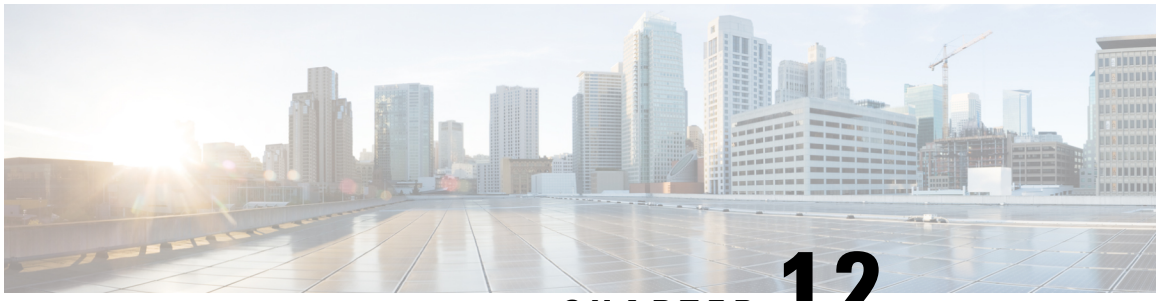
These commands are useful for monitoring the use of IPv6 addressing in a dual stack environment.

Cisco IOS XE Catalyst SD-WAN devices

- **show sdwan control connections**
- **show sdwan control local-properties**
- **show sdwan bfd sessions**
- **show sdwan omp tlocs**
- **show sdwan bfd tloc-summary-list**

SD-WAN Manager and SD-WAN Controller

- **show control connections**
- **show control local-properties**



CHAPTER 12

DHCP for IPv6

This table describes the feature history of DHCP for IPv6.

Table 15: Feature History

Feature Name	Release Information	Description
DHCP for IPv6	Cisco IOS XE Catalyst SD-WAN Release 17.7.1a Cisco vManage Release 20.7.1	This feature allows you to configure DHCP for IPv6 (DHCPv6) on Cisco IOS XE Catalyst SD-WAN devices to assign IPv6 addresses to hosts on an IPv6-enabled network. Assigning of IPv6 addresses is accomplished using SLAAC, DHCPv6, DHCPv6 Prefix Delegation, or DHCPv6 Relay. A Cisco IOS XE Catalyst SD-WAN device can be configured for DHCPv6 as a DHCP server, DHCP client, or as a DHCP relay agent.

Prerequisites for DHCP for IPv6

Ensure basic IPv6 connectivity for assigning IPv6 addresses to hosts connected to the devices.

Restrictions for DHCP for IPv6

These are restrictions that apply to DHCP for IPv6.

- DHCP for IPv6 is supported only through CLI configuration.
- A unique DHCPv6 pool name must be provided for each VRF.

DHCP for IPv6

DHCP for IPv6 (DHCPv6) is a protocol used by IPv6 devices to obtain IP addresses and configuration parameters from DHCPv6 servers. It operates similarly to DHCP for IPv4 but uses IPv6-specific mechanisms and messages.

You can configure Dynamic Host Configuration Protocol (DHCP) for IPv6 to assign addresses on an IPv6-enabled network. Alternatively, you can also configure Stateless Address Autoconfiguration (SLAAC) to assign addresses on an IPv6-enabled network.

SLAAC

The most common method for IPv6 client address assignment is SLAAC. SLAAC provides simple plug-and-play connectivity where hosts self-assign an address based on the IPv6 prefix.

SLAAC is configured like this:

- Host sends a router solicitation message.
- Hosts waits for a Router Advertisement (RA) message.
- Hosts take the first 64 bits of the IPv6 prefix from the RAmessage and combines it with the 64 bit EUI-64 address (in the case of ethernet, this is created from the MAC Address) to create a global unicast message. The host also uses the source IP address, in the IP header, of the RA message, as its default gateway.
- Duplicate Address Detection (DAD) is performed by IPv6 clients in order to ensure that random addresses that are picked do not collide with other clients.
- The choice of algorithm is up to the client and is often configurable.

The last 64 bits of the IP v6 address can be learned based on these two algorithms:

- EUI-64 which is based on the MAC address of the interface, or
- Private addresses that are randomly generated.

SLAAC and DHCPv6

DHCPv6	IPv6 devices use multicast to acquire IP addresses and to find DHCPv6 servers. The basic DHCPv6 client-server concept is similar to DHCP for IPv4. If a client wants to receive configuration parameters, it sends out a request on the attached local network to detect available DHCPv6 servers. The server responds with the requested information in a Reply message. The DHCPv6 client knows whether to use DHCPv6 based upon the instruction from a router on its link-local network. The default gateway has two configurable bits in its RA available for this purpose: • O bit—When this bit is set, the client can use DHCPv6 to retrieve other configuration parameters (for example, TFTP server addressor DNS server address) but not the client's IP address. • M bit—When this bit is set, the client can use DHCPv6 to retrieve a managed IPv6 address and other configuration parameters from a DHCPv6 server.
Stateless DHCP	Stateless DHCPv6 is a combination of SLAAC and DHCPv6. With this option SLAAC is still used to retrieve an IP address while DHCP is used to obtain additional information such as TFTP server addressor, DNS server address. In this case, the device sends an RA with the O bit set but does not set the M bit. This is known as Stateless DHCPv6 because the DHCPv6 server does not have to track the client address bindings.
Stateful DHCP	Stateful DCHPv6 functions exactly the same as DHCP IPv4 in which hosts receive both their IPv6 address and additional parameters from the DHCP server. When a device sends an RA with the M bit set, this indicates that clients must use DHCP to obtain their IP addresses. When the M bit is set, the setting of the O bit is irrelevant because the DHCP server also returns other configuration information together with the addresses. This is known as Stateful DHCPv6 because the DHCPv6 server tracks the client address bindings.

DHCPv6 prefix delegation

The DHCPv6 prefix delegation feature is a stateful mode of operation for simple delegation of prefixes from a delegating edge device (DHCP server) to requesting edge device (DHCP clients).

DHCPv6 prefix delegation feature is ideal for situations where:

- A delegating edge device that does not have the information about the topology of the networks to which the requesting edge device is attached to.
- A delegating edge device does not require other information apart from the identity of the requesting edge device to choose a prefix for delegation. This mechanism is appropriate for use by an ISP to delegate a prefix to a subscriber. After the ISP has delegated prefixes to a subscriber, the subscriber may further subnet and assign prefixes to the links within the subscriber's network.

DHCPv6 relay

A DHCPv6 relay agent is an edge device, residing on the client's network, is used to relay messages between the client and the server when a DHCPv6 server is not in the same network as the DHCPv6 clients.

Benefits

Configuring DHCP for IPv6 allows you to have more IP address compared to IPv4. With IPv6, there can be no depletion of IP addresses.

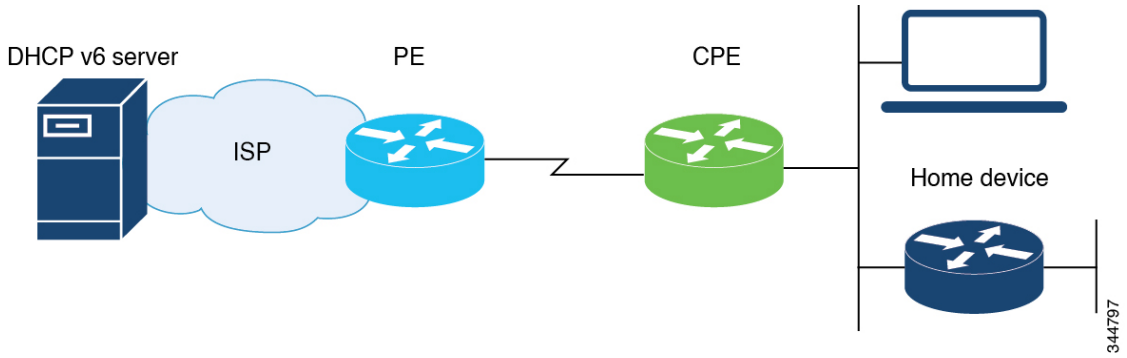
Use cases for DHCP for IPv6

These are use cases for DHCP for IPv6.

Cisco IOS XE Catalyst SD-WAN devices can be configured for DHCPv6 as a server, client, or a relay agent. As a server, a device can be configured for SLAAC, stateless DHCP, or for prefix delegation.

SLAAC with DHCP

The figure below shows a typical broadband deployment.



A Cisco IOS XE Catalyst SD-WAN device deployed on a customer premises (CPE) and connected to a ISP edge (PE) device can be a stateless or stateful DHCPv6 client. In either case, the ISP-side DHCPv6 server might provide configuration parameters such as Domain Name System (DNS) server addresses, domain names, and Simple Network Time Protocol (SNTP) servers to the DHCP client on the CPE. Such information can be specific to ISPs.

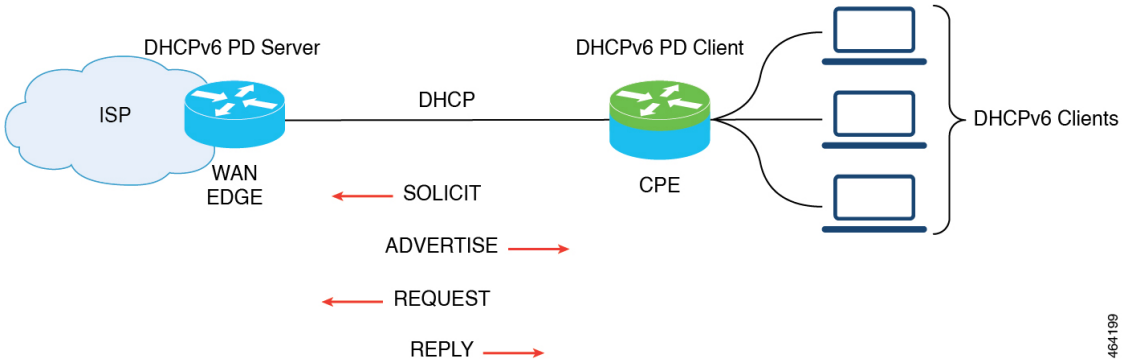
In addition to being a DHCPv6 client (toward the ISP), the CPE can act as a DHCPv6 server to the home network. For example, neighbor discovery followed by a stateless or stateful DHCPv6 client can occur on the link between the CPE and the home devices. In some cases, the information to be provided to the home network is the same as that obtained from the ISP-side DHCPv6 server. Therefore, the DHCPv6 component on the CPE allows automatic importing of configuration parameters from the DHCPv6 client to the DHCPv6 server pool.

DHCPv6 Prefix Delegation

The model of operation for prefix delegation is as follows. In this sample topology, an edge device is configured as a DHCP server which is provisioned with prefixes to be delegated to a DHCP client. A Cisco IOS XE Catalyst SD-WAN device is configured as a DHCP client and requests prefix(es) from the server. The server chooses prefix(es) for delegation and responds with prefix(es) to the DHCP client. The DHCP client is then responsible for the delegated prefix(es).

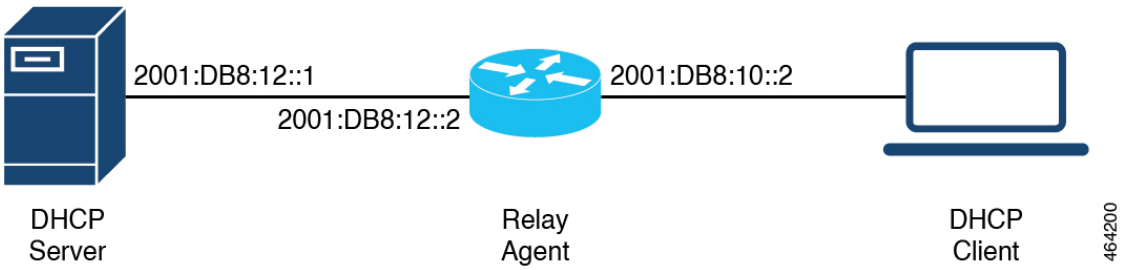
For example, the client might assign a subnet from a delegated prefix to one of its interfaces and begin sending Router Advertisements for the prefix on that link. Each prefix has an associated preferred lifetime and valid lifetime, which constitute an agreement about the length of time over which the client is allowed to use the

prefix. A client can request an extension of the lifetimes on a delegated prefix and is required to terminate the use of a delegated prefix if the valid lifetime of the prefix expires.



DHCPv6 Relay

In this sample topology, the DHCP server is not in the same network as DHCP client. A Cisco IOS XE Catalyst SD-WAN device residing on the client's network acts as a relay agent to relay messages between the client and the server.



Methods for configuring DHCP for IPv6 using CLI commands

These are methods for configuring DHCP for IPv6 (DHCPv6) using CLI commands.

Use a CLI add-on profile or template to configure devices from Cisco SD-WAN Manager using CLI commands. The sections here provide examples of various DHCP configurations.

Configure SLAAC

This example shows how to configure SLAAC on the client side.

```
device(config)# interface GigabitEthernet0/0/2
device(config-if)# ipv6 address autoconfig
device(config-if)# ipv6 enable
device(config-if)# end
```

This example shows how to configure SLAAC on the server side.

```
device(config)# interface GigabitEthernet1
device(config-if)# ipv6 address 2010:AB8:0:1::1/64
device(config-if)# ipv6 enable
device(config-if)# end
```

Configure SLAAC and DHCPv6 pool for options

This example shows how to configure SLAAC and DHCPv6 pool on the client side.

```
device(config)# interface GigabitEthernet0/0/2
device(config-if)# ipv6 address autoconfig
device(config-if)# ipv6 enable
device(config-if)# ipv6 nd autoconfig default-route
device(config-if)# ipv6 dhcp client request vendor
device(config-if)# end
```

This example shows how to configure SLAAC and DHCPv6 pool on the server side.

```
device(config)# interface GigabitEthernet1
device(config-if)# ipv6 address 2010:AB8:0:1::1/64
device(config-if)# ipv6 enable
device(config-if)# ipv6 nd autoconfig default-route
device(config-if)# ipv6 nd other-config-flag
device(config-if)# ipv6 dhcp server dhcpv6
device(config-if)# end

device(config)# ipv6 dhcp pool dhcpv6
device(config-dhcpv6)# dns-server 2001:DB8:3000:3000::42
device(config-dhcpv6)# domain-name example.com
device(config-dhcpv6)# vendor-specific 100
device(config-dhcpv6)# suboption 1 address 2001:CC:1234:44::10
device(config-dhcpv6)# suboption 2 ascii "ip phone"
```

Configure DHCPv6 (stateful) address assignment

This example shows how to configure DHCPv6 address assignment on the client side.

```
device(config)# interface GigabitEthernet0/0/2
device(config-if)# ipv6 address dhcp
device(config-if)# ipv6 enable
device(config-if)# ipv6 nd autoconfig default-route
device(config-if)# ipv6 dhcp client request vendor
device(config-if)# end
```

This example shows how to configure DHCPv6 address assignment on the server side.

```
device(config)# interface GigabitEthernet1
device(config-if)# ipv6 address 2010:AB8:0:1::1/64
device(config-if)# ipv6 enable
device(config-if)# ipv6 nd autoconfig default-route
device(config-if)# ipv6 nd managed-config-flag
device(config-if)# ipv6 dhcp server dhcpv6
device(config-if)# end

device(config)# ipv6 dhcp pool dhcpv6
device(config-dhcpv6)# address prefix 2010:AB8:0:1::1/64 lifetime 200 200
device(config-dhcpv6)# dns-server 2001:DB8:3000:3000::42
device(config-dhcpv6)# domain-name example.com
device(config-dhcpv6)# vendor-specific 100
device(config-dhcpv6)# suboption 1 address 2001:CC:1234:44::10
device(config-dhcpv6)# suboption 2 ascii "ip phone"
```

Configure DHCPv6 with prefix delegation (stateful)

This example shows how to configure DHCPv6 with prefix delegation on the client side.

```
device(config)# interface GigabitEthernet0/0/2
device(config-if)# ipv6 enable
device(config-if)# ipv6 nd autoconfig default-route
device(config-if)# ipv6 dhcp client pd prefix_from_provider
device(config-if)# ipv6 dhcp client request vendor
device(config-if)# end
```

This example shows how to configure DHCPv6 with prefix delegation on the server side.

```
device(config)# interface GigabitEthernet1
device(config-if)# ipv6 address 2010:AB8:0:1::1/64
device(config-if)# ipv6 enable
device(config-if)# ipv6 nd autoconfig default-route
device(config-if)# ipv6 nd managed-config-flag
device(config-if)# ipv6 nd ra interval 20
device(config-if)# ipv6 dhcp server dhcpv6
device(config-if)# end

device(config)# ipv6 dhcp pool dhcpv6
device(config-dhcpv6)# prefix-delegation pool dhcpv6-pool1 lifetime 200 200
device(config-dhcpv6)# dns-server 2001:DB8:3000:3000::42
device(config-dhcpv6)# domain-name example.com
device(config-dhcpv6)# vendor-specific 100
device(config-dhcpv6)# suboption 1 address 2001:CC:1234:44::10
device(config-dhcpv6)# suboption 2 ascii "ip phone"
device(config)# ipv6 local pool dhcpv6-pool1 2001:DB8:1200::/40 48
```

Configure DHCPv6 with relay

This example shows how to configure DHCPv6 with relay on the client side.

```
device(config)# interface GigabitEthernet3
device(config-if)# ipv6 address dhcp
device(config-if)# ipv6 enable
device(config-if)# ipv6 dhcp client pd pr-from-pd
device(config-if)# ipv6 dhcp client request vendor
device(config-if)# no mop enabled
device(config-if)# no mop sysid
device(config-if)# end
```

This example shows the configurations on the client facing a WAN edge device that acts as the relay agent.

```
device(config)# interface TenGigabitEthernet0/0/5
device(config-if)# vrf forwarding 10
device(config-if)# load-interval 30
device(config-if)# ipv6 address 2001:BB:1000::10/64
device(config-if)# ipv6 enable
device(config-if)# ipv6 dhcp relay destination 2001:BB8:1200::2
device(config-if)# ipv6 dhcp relay option vpn
device(config-if)# end
```

This example shows the configurations on the server facing WAN edge device.

```
device(config)# interface GigabitEthernet0/0/3
device(config-if)# vrf forwarding 10
device(config-if)# no ip address
device(config-if)# negotiation auto
```

```
device(config-if)# ipv6 address 2001:BB8:1200::1/64
device(config-if)# ipv6 enable
device(config-if)# end
```

This example shows how to configure DHCPv6 with relay on the server side.

```
device(config)# interface GigabitEthernet2
device(config-if)# ipv6 address 2001:BB8:1200::2/64
device(config-if)# ipv6 enable
device(config-if)# ipv6 dhcp server dhcpv6
device(config-if)# end

device(config)# ipv6 dhcp pool dhcpv6
device(config-dhcpv6)# prefix-delegation pool dhcpv6-pool10 lifetime infinite infinite
device(config-dhcpv6)# address prefix 2001:BB:1000::/64 lifetime 200 200
device(config-dhcpv6)# dns-server 2001:BB:1200::42
device(config-dhcpv6)# domain-name relay.com
device(config)# ipv6 local pool dhcpv6-pool10 8001:ABCD::/40 48
```

Methods for verifying a DHCPv6 client and server configuration using CLI commands

These are methods of verifying DHCP for IPv6 (DHCPv6) configurations.

Verify DHCPv6 interface information

This is a sample output from the **show ipv6 dhcp interface** command that provides details about DHCPv6 address allocation.

```
Device# show ipv6 dhcp interface GigabitEthernet0/0/2
GigabitEthernet0/0/2 is in client mode
  Prefix State is IDLE
  Address State is OPEN
  Renew for address will be sent in 00:01:09
  List of known servers:
    Reachable via address: FE80::250:56FF:FEBD:DBD1
    DUID: 00030001001EBD43F800
    Preference: 0
    Configuration parameters:
      IA NA: IA ID 0x00080001, T1 100, T2 160
        Address: 2010:AB8:0:1:95D1:CFC:F227:23FB/128
                 preferred lifetime 200, valid lifetime 200
                 expires at Oct 26 2021 07:28 AM (170 seconds)
      DNS server: 2001:DB8:3000:3000::42
      Domain name: example.com
      Information refresh time: 0
      Vendor-specific Information options:
          Enterprise-ID: 100
  Prefix Rapid-Commit: disabled
  Address Rapid-Commit: disabled
```

This is a sample output from the **show ipv6 dhcp interface** command that provides details about DHCPv6 prefix delegation.

```
Device# show ipv6 dhcp interface GigabitEthernet0/0/2
GigabitEthernet0/0/2 is in client mode
  Prefix State is OPEN
  Renew will be sent in 00:01:34
  Address State is IDLE
```

```
  List of known servers:
    Reachable via address: FE80::250:56FF:FEBD:DBD1
    DUID: 00030001001EBD43F800
    Preference: 0
    Configuration parameters:
      IA PD: IA ID 0x00080001, T1 100, T2 160
        Prefix: 2001:DB8:1202::/48
                preferred lifetime 200, valid lifetime 200
                expires at Oct 26 2021 07:30 AM (194 seconds)
      DNS server: 2001:DB8:3000:3000::42
      Domain name: example.com
      Information refresh time: 0
  Prefix name: prefix_from_server
  Prefix Rapid-Commit: disabled
  Address Rapid-Commit: disabled
```

This is a sample output from the **show ipv6 dhcp interface** command that provides details about SLAAC with DHCP.

```
Device# show ipv6 dhcp interface GigabitEthernet0/0/2
GigabitEthernet0/0/2 is in client mode
  Prefix State is IDLE (0)
  Information refresh timer expires in 23:59:49
  Address State is IDLE
  List of known servers:
    Reachable via address: FE80::250:56FF:FEBD:DBD1
    DUID: 00030001001EBD43F800
    Preference: 0
    Configuration parameters:
      DNS server: 2001:DB8:3000:3000::42
      Domain name: example.com
      Information refresh time: 0
      Vendor-specific Information options:
          Enterprise-ID: 100
  Prefix Rapid-Commit: disabled
  Address Rapid-Commit: disabled
```

View DHCPv6 Pool information

This is a sample output from the **show ipv6 dhcp pool** command that provides details about DHCPv6 address allocation.

```
Device# show ipv6 dhcp pool
DHCPv6 pool: relay_server
  VRF 10
  Prefix pool: dhcpv6-pool2
  Address allocation prefix: 5001:DB8:1234:42::/64 valid 20000 preferred 20000 (1 in use,
0 conflicts)
               preferred lifetime 200, valid lifetime 200
  DNS server: 2001:BB8:3000:3000::42
  Domain name: relay.com
  Information refresh: 60
  Vendor-specific Information options:
  Enterprise-ID: 10
    suboption 1 address 2001:DB8:1234:42::10
    suboption 2 ascii 'ip phone'
  Active clients: 1
  Pool is configured to include all configuration options in REPLY
```

This is a sample output from the **show ipv6 dhcp pool** command that provides details about DHCPv6 prefix delegation.

```
Device# show ipv6 dhcp pool
DHCPv6 pool: relay_server
```

```
  VRF 10
  Prefix pool: dhcpv6-pool2
  Address allocation prefix: 5001:DB8:1234:42::/64 valid 20000 preferred 20000 (0 in use,
0 conflicts)
              preferred lifetime 200, valid lifetime 200
  DNS server: 2001:BB8:3000:3000::42
  Domain name: relay.com
  Information refresh: 60
  Vendor-specific Information options:
  Enterprise-ID: 10
    suboption 1 address 2001:DB8:1234:42::10
    suboption 2 ascii 'ip phone'
  Active clients: 1
  Pool is configured to include all configuration options in REPLY
```

View DHCPv6 bindings

This is a sample output from the **show ipv6 dhcp binding** command that provides details about DHCPv6 address allocation.

```
Device# show ipv6 dhcp binding
Client: FE80::250:56FF:FEBD:8261
  DUID: 00030001001EE6DBF500
  Username : unassigned
  VRF : 10
  IA NA: IA ID 0x00080001, T1 10000, T2 16000
    Address: 5001:DB8:1234:42:500C:B3FA:54A7:F63D
             preferred lifetime 20000, valid lifetime 20000
             expires at Oct 26 2021 01:17 PM (19925 seconds)
```

This is a sample output from the **show ipv6 dhcp binding** command that provides details about DHCPv6 prefix delegation.

```
Device# show ipv6 dhcp binding
Client: FE80::250:56FF:FEBD:8261
  DUID: 00030001001EE6DBF500
  Username : unassigned
  VRF : 10
  Interface : GigabitEthernet0/0/3
  IA PD: IA ID 0x00080001, T1 100, T2 160
    Prefix: 2001:BB8:1602::/48
             preferred lifetime 200, valid lifetime 200
             expires at Oct 26 2021 08:01 AM (173 seconds)
```

View DHCPv6 database

This is a sample output from the **show ipv6 dhcp database** command.

```
Device# show ipv6 dhcp database
Database agent bootflash:
  write delay: 300 seconds, transfer timeout: 300 seconds
  last written at Oct 26 2021 08:01 AM, write timer expires in 250 seconds
  last read at never
  successful read times 0
  failed read times 0
  successful write times 2
  failed write times 0
```

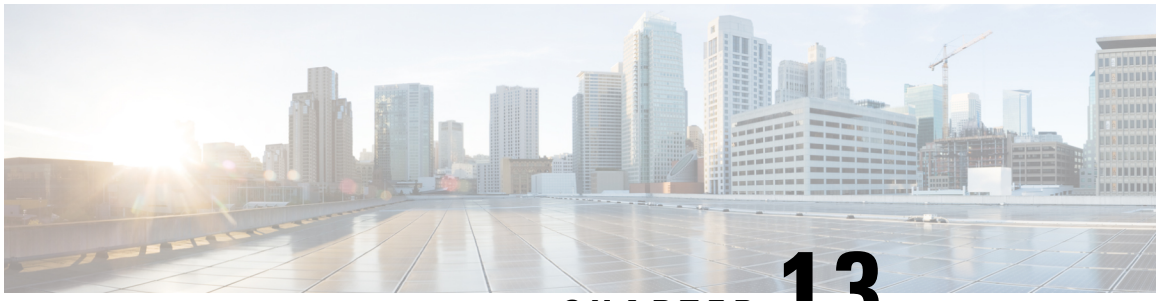
View DHCPv6 relay bindings

This is a sample output from the **show ipv6 dhcp relay bindings** command that provides details about DHCPv6 relay.

```
Device# show ipv6 dhcp relay binding

Relay Bindings associated with default vrf:

Relay Bindings associated with vrf 10:
Prefix: 2001:AA8:1100::/48 (GigabitEthernet3)
  DUID: 00030001001E49674C00
  IAID: 851969
  lifetime: INFINITE
  expiration: INFINITE
Summary:
  Total number of Relay bindings = 1
  Total number of IAPD bindings = 1
  Total number of IANA bindings = 0
  Total number of Relay bindings added by Bulk lease = 0
```



CHAPTER 13

Per Packet Load Balancing

Feature history for per packet load balancing

This table describes the developments of this feature, by release.

Table 16: Feature history

Feature name	Release information	Description
Per packet load balancing	Cisco IOS XE Catalyst SD-WAN Release 26.1.x Cisco Catalyst SD-WAN Manager Release 26.1.x	With this feature, Cisco SD-WAN sends packets from a single flow across multiple WAN links, maximizing bandwidth and maintaining performance by reordering packets at the destination.

Per packet load balancing

Per-packet load balancing (PPL) or adaptive single flow distribution is a bandwidth aggregation feature that

- distributes packets from a single flow across multiple WAN links,
- maximizes the use of all available paths, and
- maintains data integrity by reordering packets at the destination.

Traditional load balancing vs PPL

Traditional load balancing uses one network link for each data stream, which can limit overall bandwidth use. PPL shares traffic across multiple available links, allowing you to get the most out of your network and improve performance.

Reorder packets

When a device distributes packets over multiple links, packets may arrive at the destination out of order. The PPL feature automatically reorders these packets on a best-effort basis and releases any buffered packets based on either a time threshold or a memory threshold.

Key concepts for understanding PPL

Explains how PPL efficiently distributes traffic across multiple WAN tunnels by dynamically selecting the best paths based on latency, flowlets, and network policies.

Sender and receiver

The sender is the WAN edge device that sends PPL traffic. The receiver is the WAN edge device that receives this traffic and may reorder packets if they arrive out of order.

Traffic selection

Defines which specific traffic from the service VPNs use PPL, enabling detailed control through network policies. You can configure this using data policies in the "from-service" direction, and any traffic matched in the policy’s match clause becomes eligible for PPL.

Inter-tunnel latency measurement

PPL measures the delay across all available tunnels and selects the tunnel with the lowest latency as a reference point. It compares other tunnels to this reference and includes only those within a set latency range for traffic distribution. This method is a Cisco proprietary algorithm. It ensures efficient and reliable path selection for your network traffic.

Flow splitting

Instead of splitting traffic into individual packets, PPL groups packets into flowlets, which are small sets of packets. You create flowlets either after sending a set number of packets or when you detect natural breaks (packet gaps) in the traffic.

Packet gap

Packet gap is a method used in PPL to detect natural breaks in a flow of traffic. When there is an idle period between bursts of packets that exceeds the maximum latency difference among available tunnels, a new flowlet is started and sent on a different tunnel.

Candidate tunnel group (CTG)

The group of tunnels available for PPL, selected first by “local colors” and then filtered using metrics like latency and loss. A maximum of eight tunnels can be part of the CTG.

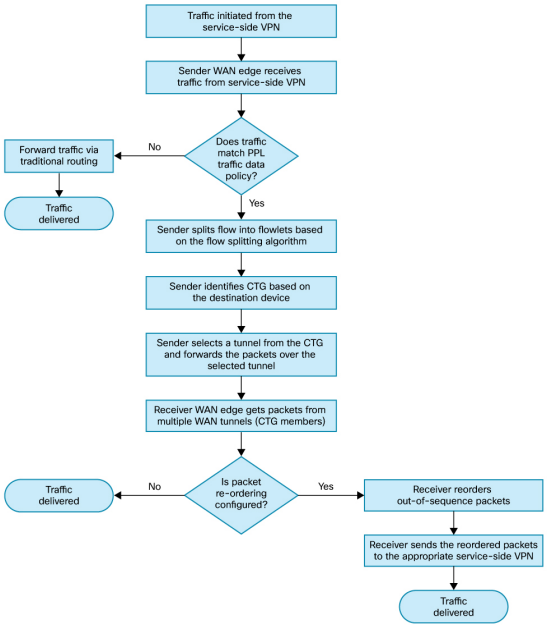
Path selection

Determines how each flowlet is sent across the reasonably available tunnels in the CTG, using round-robin distribution.

Sequence of events for PPL

The sender device receives traffic from service-side VPNs and checks it against the traffic data policy for PPL rules. If matched, the sender splits the flow into flowlets and forwards them over reasonably available tunnels in the Candidate Tunnel Group. The receiver device collects and re-orders the flowlets as needed, then forwards the reordered flows into the appropriate service VPNs.

See the illustration to understand the sequence of events.



Benefits of PPL in Cisco SD-WAN

Benefits of PPL:

- Adaptibility: Adjusts quickly to changes in network conditions to keep performance steady.
- Bandwidth and resource utilization: Combines the capacity of multiple WAN links for higher data transfer rates.

  Uses all available network paths to make the optimal usage of bandwidth.
- Performance: Increases speed and efficiency for elephant flows.

  Supports applications that need more bandwidth than a single link can provide.

Use cases for PPL

Scenario 1: Managing AI traffic between offices and data centers

AI tasks often involve sending large amounts of information from local offices to a central data center for processing. At the same time, users need quick, real-time responses from AI tools. This traffic is unpredictable

and comes in sudden bursts. When all this data is forced through a single network path, it can create network congestion, leading to slow AI performance and delayed data uploads.

How does PPL help?

PPL breaks these large data streams into smaller, manageable pieces and sends them across all available network connections at once. This prevents any single connection from becoming overloaded.

Scenario 2: High-volume workload transfers and backups in enterprise environments

Enterprise environments routinely perform large-scale data transfers for activities such as workload upgrades, Windows operating system or application updates, data center backups, and disaster recovery operations. These scenarios often require moving significant amounts of data between endpoints. Such transfers can easily exceed the capacity of individual WAN circuits, especially when using multiple mid-sized links.

How does PPL help?

PPL splits large, single flows into multiple flowlets and distributes them across all available WAN circuits. This approach maximizes total available bandwidth, prevents any single link from being overwhelmed, and ensures that even deprioritized or non-critical traffic, like updates and backups, completes efficiently without impacting business-critical applications.

Supported platforms for PPL

These tables outline the supported hardware models and their corresponding DRAM requirements for Receiver and Sender PPL roles.

Table 17: Receiver PPL supported platforms

Model	Minimum DRAM
C8500-12X4QC	32 GB
C8500-20X6C	64 GB
C8570-G2	32 GB
C8300-2N2S-6T	16 GB
C8300-2N2S-4T2X	16 GB

Table 18: Sender PPL supported platforms

Model	Minimum DRAM
C8300-2N2S-4T2X	16 GB
C8300-2N2S-6T	16 GB
C8500L-8S4X	15 GB
C8475-G2	32 GB
C8455-G2	32 GB
C8500-12X4QC	16 GB

Model	Minimum DRAM
C8500-12X	16 GB
C8550-G2	32 GB

Note Receiver platforms can act as senders, but sender platforms cannot act as receivers.

Restrictions for PPL

Restrictions for PPL:

- Maximum senders: A receiver can only support up to 256 senders.
- Maximum receivers: A sender can only support up to 512 receivers.
- PPL data policy: You can configure PPL data policy action only in the ‘from-service’ direction.

  Do not use packet duplication (packet dup) or forward error correction (FEC) actions with PPL.
- SLA classes: Limit SLA classes to 14 to reserve one for PPL. If 15 SLA classes are configured, remove one and reload the device before configuring PPL.
- Multicast traffic: Multicast traffic is not supported with PPL.
- Service insertion: Service insertion is not supported with PPL.
- Parallel routing configurations: Sender cannot be configured with any parallel routing decisions such as next-hop, and remote-TLOC through data policy when PPL is enabled.

Configure PPL

Use one of these procedures to configure PPL.

- Configure PPL using configuration groups
- Configure a data policy with load balancing using policy groups
- Configure maximum packet value in PPL using CLI commands
- Configure latency offset value in PPL using CLI commands

Configure PPL using configuration groups

Use these steps to configure PPL using configuration groups.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration groups**.

Step 2 Create or edit a System profile.

Step 3 Navigate to Basic Settings and select **Per Packet Load Balancing**.

Step 4 Enable **PPL** on the Sender and configure its parameters mentioned below.

Step 5 Enable packet reordering on the receiver.

Table 19: PPL fields

Field	Description
Local colors	Specify the WAN links (such as MPLS, Internet, and LTE) used to distribute traffic within the network.
Flow splitting	Specify how to divide traffic among the available tunnels. **Packets:** Sends a set number of packets (default: 1024) through one tunnel before switching to the next flowlet. This cycle repeats continuously to balance the network load. To set a maximum packet value, refer toConfigure maximum packet value in PPL using CLI commands **Packet gap:** Switches tunnels based on the time interval between packets. If the time gap between bursts exceeds the maximum latency difference among tunnels, a new flowlet is sent on a different tunnel. This helps distribute traffic more effectively during natural pauses in data transmission.
Candidate tunnel group	Dynamically defines the group of tunnels used in a PPL network. All tunnels: All available tunnels become part of PPL network. Latency offset: Includes only those tunnels whose latency is within a configurable threshold of the lowest-latency tunnel. To set latency offset value, refer to Configure latency offset value in PPL using CLI commands. Adaptive cluster: Groups tunnels dynamically into clusters based on real-time loss and latency measurements. Tunnels with similar loss (primary factor) and latency (secondary factor) are grouped together for packet distribution.
Reorder	Enables packet reordering at the receiving end to reduce out-of-order delivery. If you disable this option, ensure that your application can independently handle packet reordering.

Configure a data policy with load balancing using policy groups

Use these steps to configure data policy with load balancing.

Procedure

Step 1 From the Cisco SD-WAN Manager, choose **Configuration** > **Policy Groups** > **Application Priority and SLA**.
Step 2 Create or edit an existing policy.
Step 3 Add or create a new Traffic Policy within the policy.
Step 4 Set the direction to **From Service**.
Step 5 In the Match Clause, specify the traffic to be matched, such as the DSCP source.
Step 6 In the Action section, select the **radio button** to enable Load Balancing .
Step 7 From the **Load Balancing Algorithm** drop-down list, select **per packet**.
Step 8 Select **Save**.

Configure maximum packet value in PPL using CLI commands

Use these steps to configure the maximum packet value in PPL.

Procedure

Step 1 Enter configuration mode.

Example:

```
config-transaction
```

Step 2 Set the maximum number of packets in a flowlet.

Example:

```
sdwan ppl flow-splitting max-pkts 2048
```

Enter a value between 1024 and 10240 to define the maximum packet count for a flowlet.

Here's the complete configuration example for configuring maximum packet value in PPL.

```
sdwan
 ppl
  enable
  flow-splitting
   max-pkts 2048
   !
  !
 !
```

Configure latency offset value in PPL using CLI commands

Use these steps to configure the latency offset value in PPL.

Procedure

Step 1 Enter configuration mode.

Example:

```
config-transaction
```

Step 2 Configure the latency offset value for the CTG algorithm.

Example:

```
sdwan ppl ctg algo latency-offset 15
```

Enter the desired latency offset value to be used by the PPL CTG algorithm.

Here's the complete configuration example for configuring maximum packet value in PPL.

```
sdwan
 ppl
  ctg
    algo
     latency-offset 15
     !
    !
  !
!
```

Monitor PPL

Use these show commands to monitor PPL.

To verify the configured flow splitting and Candidate Tunnel Group (CTG) algorithms on a sender, use the **show platform software sdwan ftmd ppl cfg** command.

```
sender# show platform software sdwan ftmd ppl cfg
PPL Config
 Enable             : true
 Local Color List   : all-colors
 Path Selection     : round-robin
 OWL
   Interval : 20 sec
   DSCP     : 48 (0x30)
 Flow Splitting
   Algo     : pkt-gap adaptive

 CTG
   Algo     : adaptive-cluster
```

To verify if the PPL configuration flag is enabled for specific sites on a device, use the **show sdwan ftm site-db** command.

```
sender# show sdwan ftm site-db


  Site ID: 400, OD: No, status: Active, sys_cnt: 2, status_change: cnt
 2time 04:14:19:532, last_stats 0, is_pending_tloc_re
play=0, replay_cnt=1 Idle Timer: Not inited

  Remote    system: 172.16.255.41, OD: No, status: Active, tloc_cnt: 4,
 tun_cnt: 16, glean_ipc_cnt: 0, is_pending_tloc_repl
ay=0, replay_cnt=103, pkt_ro_ctx_id: 1, ppl_ctx_id: 1 ppl_cfg: True
ppl_adaptive_flowlet_gap: 1000
        TLOC   :: 172.16.255.41 : bronze : ipsec, nh_index: 32772, OD: No,
 status: Inactive
        TLOC   :: 172.16.255.41 : private2 : ipsec, nh_index: 32776, OD: No,
 status: Inactive
        TLOC   :: 172.16.255.41 : mpls : ipsec, nh_index: 32782, OD: No,
 status: Inactive
        TLOC   :: 172.16.255.41 : biz-internet : ipsec, nh_index: 32784, OD: No, status:
Inactive
<snip>
  Remote    system: 172.16.255.40, OD: No, status: Active, tloc_cnt: 5,
 tun_cnt: 20, glean_ipc_cnt: 0, is_pending_tloc_repl
ay=0, replay_cnt=177, pkt_ro_ctx_id: 2, ppl_ctx_id: 2 ppl_cfg: True
 ppl_adaptive_flowlet_gap: 1000
        TLOC   :: 172.16.255.40 : mpls : ipsec, nh_index: 32773, OD: No,
 status: Inactive
        TLOC   :: 172.16.255.40 : private1 : ipsec, nh_index: 32775, OD: No,
 status: Inactive
        TLOC   :: 172.16.255.40 : biz-internet : ipsec, nh_index: 32777, OD: No,
 status: Inactive
        TLOC   :: 172.16.255.40 : private2 : ipsec, nh_index: 32783, OD: No,
 status: Inactive
        TLOC   :: 172.16.255.40 : bronze : ipsec, nh_index: 32785, OD: No,
 status: Inactive
        Tunnel : 10.1.61.40/12346->10.1.41.40/12346 proto 0x32, idx 11,
sys 172.16.255.40, nh_index 65546
```

To verify Candidate Tunnel Group (CTG) selection for a destination site, use the **show sdwan bfd sessions alt** command.

```
sender# show sdwan bfd sessions alt
*Sus = Suspend
*GREinUDP = GREinUDP encap
*EAAR  = Enhanced Application-Aware Routing
*PPL = Per Packet Loadbalancing
*NA  = Flag Not Set
                                      SOURCE TLOC      REMOTE TLOC
              DST PUBLIC              DST PUBLIC
SYSTEM IP        SITE ID     STATE       COLOR            COLOR             SOURCE IP
              IP                       PORT        ENCAP  BFD-LD      FLAGS
UPTIME
------------------------------------------------------------------------------------------------------------------------
172.16.255.40    400         up          mpls             biz-internet      10.1.61.40
              10.1.40.40               12346       ipsec  20019       PPL
13:01:01:32
172.16.255.40    400         up          mpls             mpls              10.1.61.40
              10.1.41.40               12346       ipsec  20022       NA
13:01:01:31
172.16.255.40    400         up          mpls             private1          10.1.61.40
```

```
              10.1.42.40              12346       ipsec  20020       PPL
13:01:01:31
172.16.255.40    400         up          mpls             bronze          10.1.61.40
              10.1.43.40              12346       ipsec  20021       PPL
13:01:01:32
172.16.255.40    400         up          biz-internet     biz-internet    20.1.60.40
              10.1.40.40              12346       ipsec  20001       PPL
13:01:01:45
172.16.255.40    400         up          biz-internet     mpls            20.1.60.40
              10.1.41.40              12346       ipsec  20004       PPL
13:01:01:45
172.16.255.40    400         up          biz-internet     private1        20.1.60.40
              10.1.42.40              12346       ipsec  20002       PPL
13:01:01:45
172.16.255.40    400         up          biz-internet     bronze          20.1.60.40
              10.1.43.40              12346       ipsec  20003       PPL
13:01:01:45
172.16.255.40    400         up          biz-internet     private2        20.1.60.40
              10.1.44.40              12346       ipsec  20006       PPL
13:01:01:45
```

To verify absolute Candidate Tunnel Group (CTG) data and counters, use the **show platform software sdwan ftmd ppl ctg** command.

```
sender# show platform software sdwan ftmd ppl ctg
OWL-REQ RX TS: Timestamp at which the remote device received the OWL request. In brackets:
 Latency relative to tunnel with lowest OWL.
-------------------------------------------------------------------------------------------------------------------------
LOCAL COLOR     SYSTEM-IP       REMOTE COLOR    BFD-LD    PPL-ID    PKTRO-ID    EPOCH-ID
  OWL-REQ RX TS                PKT-TX      BYTES-TX    PKT-RX      BYTES-RX
-------------------------------------------------------------------------------------------------------------------------
biz-internet    172.16.255.40   biz-internet    20001     1         65535       55905
  1119300231(0)                 0           0           0         0
biz-internet    172.16.255.40   private1        20002     1         65535       55905
  1119300231(0)                 0           0           0         0
biz-internet    172.16.255.40   bronze          20003     1         65535       55905
  1119300231(0)                 0           0           0         0
biz-internet    172.16.255.40   mpls            20004     1         65535       55905
  1119300231(0)                 0           0           0         0
biz-internet    172.16.255.40   private2        20006     1         65535       55905
  1119300231(0)                 0           0           0         0
mpls            172.16.255.40   biz-internet    20019     1         65535       55905
  1119300231(0)                 0           0           0         0
mpls            172.16.255.40   private1        20020     1         65535       55905
  1119300231(0)                 0           0           0         0
mpls            172.16.255.40   bronze          20021     1         65535       55905
  1119300231(0)                 0           0           0         0
biz-internet    172.16.255.41   bronze          20005     4         65535       55905
  1119139948(0)                 0           0           0         0
biz-internet    172.16.255.41   biz-internet    20007     4         65535       55905
  1119139948(0)                 0           0           0         0
biz-internet    172.16.255.41   private2        20008     4         65535       55905
  1119139948(0)                 0           0           0         0
biz-internet    172.16.255.41   mpls            20009     4         65535       55905
  1119139948(0)                 0           0           0         0
mpls            172.16.255.41   bronze          20023     4         65535       55905
  1119139948(0)                 0           0           0         0
mpls            172.16.255.41   biz-internet    20025     4         65535       55905
  1119139948(0)                 0           0           0         0
mpls            172.16.255.41   private2        20026     4         65535       55905
  1119139948(0)                 0           0           0         0
```

```
mpls            172.16.255.41   mpls            20027     4         65535       55905
  1119139948(0)                 0           0           0           0
```

To verify the cumulative tunnel statistics, use the **show sdwan tunnel statistics table** command.

```
sender# show sdwan tunnel statistics ppl table
TUNNEL                          SOURCE  DEST
 PROTOCOL  SOURCE IP   DEST IP     PORT    PORT   ppl-tx-ipv4-pkts  ppl-tx-ipv6-pkts
ppl-tx-ipv4-octets  ppl-tx-ipv6-octets  ppl-rx-ipv4-pkts  ppl-rx-ipv6-pkts  ppl-rx-ipv4-octets
  ppl-rx-ipv6-octets
ipsec     10.1.61.40  10.1.40.40  12346   12346  1024             0                 1224704
             0                   0                  0                 0                   0
ipsec     10.1.61.40  10.1.42.40  12346   12346  1024             0                 1224704
             0                   0                  0                 0                   0
ipsec     10.1.61.40  10.1.43.40  12346   12346  1024             0                 1224704
              0                   0                  0                 0
0
ipsec     20.1.60.40  10.1.40.40  12346   12346  1024             0                 1224704
              0                   0                  0                 0
0
ipsec     20.1.60.40  10.1.41.40  12346   12346  1024             0                 1224704
              0                   0                  0                 0
0
ipsec     20.1.60.40  10.1.42.40  12346   12346  1024             0                 1224704
             0                   0                  0                 0                   0
ipsec     20.1.60.40  10.1.43.40  12346   12346  1024             0                 1224704
             0                   0                  0                 0                   0
ipsec     20.1.60.40  10.1.44.40  12346   12346  1024             0                 1224704
             0                   0                  0                 0                   0
```

To verify the configured packet reordering parameters on a receiver, use the **show platform software pkt-reorder cfg** command.

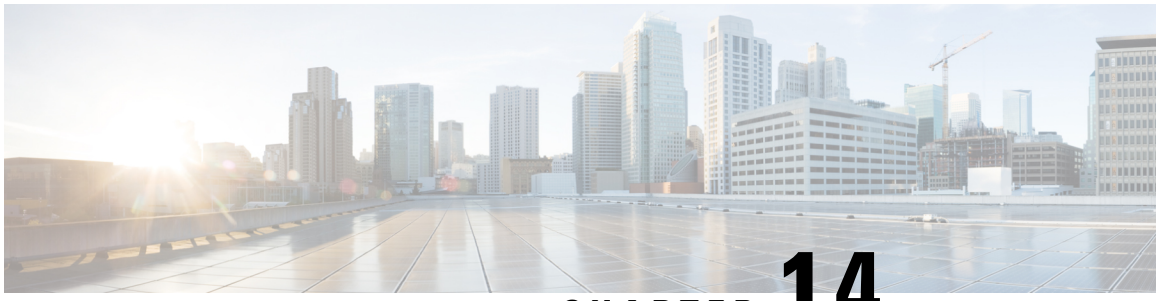
```
receiver# show platform software pkt-reorder cfg
Enable                : true
Max-allowed-memory    : 40%
Packets-per-context   : 8192
Out-of-order-tolerance : 2048
Dropped-packet-timeout : 100 ms
Idle-timeout          : 1000 ms
Dispact-size          : 2048
```

To check if the packet re-order context ID is correctly assigned for remote senders on a receiver device, use the **show sdwan ftm site-db** command.

```
receiver# show sdwan ftm site-db 300
OD Tunnel Idle Time   : 10 min
 Site ID: 300, OD: Yes, status: Inactive, sys_cnt: 1, status_change:
cnt ltime 08:59:05:953, last_stats 0, is_pending_tloc
_replay=0, replay_cnt=2 Idle Timer: Not running, time_to_expiry: 0 sec
 Remote   system: 172.16.255.80, OD: Yes, status: Inactive,
tloc_cnt: 4, tun_cnt: 20, glean_ipc_cnt: 0, is_pending_tloc_r
eplay=0, replay_cnt=1010, pkt_ro_ctx_id: 7, ppl_ctx_id: 5 ppl_cfg:
True ppl_adaptive_flowlet_gap: 1000
      TLOC  :: 172.16.255.80 : privatel : ipsec, nh_index: 32800,
OD: Yes, status: Inactive
      TLOC  :: 172.16.255.80 : bronze : ipsec, nh_index: 32801, OD:
Yes, status: Inactive
      TLOC  :: 172.16.255.80 : mpls : ipsec, nh_index: 32803, OD:
Yes, status: Inactive
      TLOC  :: 172.16.255.80 : biz-internet : ipsec, nh_index:
32804, OD: Yes, status: Inactive
      Tunnel : 10.1.41.40/12346->10.1.62.40/12346 proto 0x32, idx
960, sys 172.16.255.80, nh_index 66495
```

To view the reordering statistics of a PPL network, use the **show platform software pkt-reorder ctx sdwan** command.

```
receiver# show platform software pkt-reorder ctx sdwan
Context ID Module IP Address Total Packet Time Exceeded Too Old In Window Wrap DM Distance
 Mem Usage Total Mem Usage
1 sdwan 172.16.255.80 0 0 0 0 0 0 0 0
2 sdwan 172.16.255.81 132,213,316 403 130,239,435 829,845 1 43,955,369,702,422,670 0 0
3 sdwan 172.16.255.83 0 0 0 0 0 0 0 0
4 sdwan 172.16.255.82 0 0 0 0 0 0 0 0
```



CHAPTER 14

PPPoE

Feature history for PPPoE

Table 20: Feature History

Feature Name	Release Information	Description
Configure PPPoE over ATM	Cisco IOS XE Catalyst SD-WAN Release 17.4.1a Cisco vManage Release 20.4.1	This feature provides support for configuring PPPoEoA on Cisco IOS XE Catalyst SD-WAN devices. PPPoEoA uses AAL5MUX encapsulation which delivers better efficiency compared to other encapsulation methods.

PPPoE

A Point-to-Point Protocol over Ethernet (PPPoE) is a network protocol that

- connects multiple users over an Ethernet local area network to a remote site through common customer premises equipment,
- is commonly used in broadband aggregation such as by digital subscriber line (DSL), and
- provides authentication with the CHAP or PAP protocol.

In the Cisco Catalyst SD-WAN overlay network, CCisco IOS XE Catalyst SD-WAN devices can run the PPPoE client. The PPPoE server component is not supported. It is recommended that you configure quality

of service (QoS) and shaping rate on a PPPoE Dialer interface. Queuing based QoS policies on both Dialer interface and PPPoE-enabled physical interface at the same time, is not supported.

PPPoE-enabled physical interfaces are supported on ATM PVCs and Ethernet interfaces. A dialer interface must be used for cloning virtual access. Multiple PPPoE client sessions can be configured on an Ethernet interface, but each session must use a separate dialer interface and a separate dialer pool.

The Cisco Catalyst SD-WAN implementation of PPPoE does not support the Compression Control Protocol (CCP) options, as defined in RFC 1962.

PPPoE over ATM

A PPPoEoA is a protocol encapsulation method that

- uses ATM Adaptation Layer 5 Multiplexed Encapsulation (AAL5MUX) to carry PPPoE over ATM permanent virtual circuits (PVCs),
- provides efficiency gain over AAL5 LLC/SNAP encapsulation, and
- reduces Subnetwork Access Protocol (SNAP) encapsulation bandwidth usage by using multiplexed (MUX) encapsulation to reduce the number of cells needed to carry voice packets.

You can configure PPPoE over ATM interfaces (PPPoEoA) on Cisco IOS XE Catalyst SD-WAN devices that support ADSL. Deploying the PPPoEoA over ATM AAL5MUX feature in a VoIP environment results in improved throughput and bandwidth usage.

Supported platforms for PPPoE over ATM

The following platforms support PPPoE over ATM:

- Cisco 1100 4G/6G Series Integrated Services routers.
- Cisco1100 Series Integrated Service routers.
- Cisco1109 Series Integrated Service routers.
- Cisco111x Series Integrated Service routers.
- Cisco1111x Series Integrated Service routers.
- Cisco1120 Series Integrated Service routers.
- Cisco1160 Series Integrated Service routers.

Configure PPPoE using templates

To use Cisco SD-WAN Manager templates to configure PPPoE on Cisco IOS XE Catalyst SD-WAN device, you create three feature templates and one device template:

- Create a VPN-Interface-PPP feature template to configure PPP parameters for the PPP virtual interface.
- Create a VPN-Interface-PPP-Ethernet feature template to configure a PPPoE-enabled interface.

- Optionally, create a VPN feature template to modify the default configuration of VPN 0.
- Create a device template that incorporates the VPN-Interface-PPP, VPN-Interface-PPP-Ethernet, and VPN feature templates.

Follow these steps to configure PPPoE using feature templates.

Procedure

Step 1 Create a VPN-Interface-PPP feature template to configure PPP parameters for the PPP virtual interface.

a) From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
b) Click **Feature Templates**, and click **Add Template**.

In Cisco vManage Release 20.7.1 and earlier releases, **Feature Templates** is titled **Feature**.

c) Choose Cisco IOS XE Catalyst SD-WAN device Cloud or a router model.
d) Choose the **VPN-Interface-PPP** template.
e) In the template, configure the following parameters:

Table 21:

Parameter Field	Procedure
Template Name	Enter a name for the template. It can be up to 128 alphanumeric characters.
Description	Enter a description for the template. It can be up to 2048 alphanumeric characters.
Shutdown	Click **No** to enable the PPP virtual interface.
Interface Name	Enter the number of the PPP interface. It can be from 1 through 31.
Description (optional)	Enter a description for the PPP virtual interface.
Authentication Protocol	Select either CHAP or PAP to configure one authentication protocol, or select PAP and CHAP to configure both. For CHAP, enter the hostname and password provided by your ISP. For PAP, enter the username and password provided by your ISP. If you are configuring both PAP and CHAP, to use the same username and password for both, click Same Credentials for PAP and CHAP.
AC Name (optional)	Select the PPP tab, and in the AC Name field, enter the name of the the name of the access concentrator used by PPPoE to route connections to the Internet.
IP MTU	Click **Advanced**, and in the IP MTU field, ensure that the IP MTU is at least 8 bytes less than the MTU on the physical interface. The maximum MTU for a PPP interface is 1492 bytes. If the PPPoE server does not specify a maximum receive unit (MRU), the MTU value for the PPP interface is used as the MRU. Starting from Cisco vManage Release 20.9.1, there is 8 bytes overheads deduced based on the specified IP MTU value when configuration is pushed to the device.
Save	To save the feature template, click **Save**.

Step 2 Create a VPN-Interface-PPP-Ethernet feature template to enable the PPPoE client on the physical interfaces.

a) From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
b) Click **Feature Templates**, and click **Add Template**.

In Cisco vManage Release 20.7.1 and earlier releases, **Feature Templates** is titled **Feature**.

c) Choose Cisco IOS XE Catalyst SD-WAN device Cloud or a router model.
d) Choose the **VPN-Interface-PPP-Ethernet** template.
e) In the template, configure the following parameters:

Parameter Field	Procedure
Template Name	Enter a name for the template. It can be up to 128 alphanumeric characters.
Description	Enter a description for the template. It can be up to 2048 alphanumeric characters.
Shutdown	Click **No** to enable the PPPoE-enabled interface.
Interface Name	Enter the name of the physical interface in VPN 0 to associate with the PPP interface.
Description (optional)	Enter a description for the PPPoE-enabled interface.
IP Confguration	Assign an IP address to the physical interface: • To use DHCP, select Dynamic. The default administrative distance of routes learned from DHCP is 1. • To configure the IP address directly, enter of the IPv4 address of the interface.
DHCP Helper (optional)	Enter up to four IP addresses for DHCP servers in the network.
Save	To save the feature template, click **Save**.

Step 3 Create a VPN feature template to configure the PPPoE-enabled interface in VPN 0, the transport VPN.

a) From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
b) Click **Feature Templates**, and click **Add Template**.

In Cisco vManage Release 20.7.1 and earlier releases, **Feature Templates** is titled **Feature**.

c) Choose Cisco IOS XE Catalyst SD-WAN device Cloud or a router model.
d) Choose the **VPN** template.
e) In the template, configure the following parameters:

Parameter Field	Procedure
Template Name	Enter a name for the template. It can be up to 128 alphanumeric characters.
Description	Enter a description for the template. It can be up to 2048 alphanumeric characters.
VPN Identifier	Enter VPN identifier 0.
Name	Enter aname for the VPN.

Parameter Field	Procedure
Other interface parameters	Configure the desired interface properties.
Save	To save the feature template, click **Save**.

Step 4 Create a device template that incorporates the VPN-Interface-PPP, VPN-Interface-PPP-Ethernet, and VPN feature templates.

a) From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
b) Click **Device Templates**, and then click **Create Template**.

In Cisco vManage Release 20.7.1 and earlier releases, **Device Templates** is titled **Device**.

c) From the **Create Template** drop-down list, choose From Feature Template.

From the **Device Model** drop-down list, choose the type of device for which you are creating the device template.

Cisco SD-WAN Manager displays the feature templates for the device type you selected. Required templates are indicated with an asterisk (*).

Enter a name and description for the device template. These fields are mandatory. The template name cannot contain special characters.

d) In **Transport & Management VPN**, under **VPN 0**, from the drop-down list of available templates, select the desired feature template. The list of available templates are the ones that you have previously created.
e) In **Additional VPN 0 Templates**, click the plus sign (+) next to **VPN Interface PPP**.
f) From **VPN-Interface-PPP** and **VPN-Interface-PPP-Ethernet** fields, select the feature templates to use.
g) To configure multiple PPPoE-enabled interfaces in VPN 0, click the plus sign (+) next to Sub-Templates.
h) To include additional feature templates in the device template, in the remaining sections, select the feature templates in turn, and from the drop-down list of available templates, select the desired template. The list of available templates are the ones that you have previously created. Ensure that you select templates for all mandatory feature templates and for any desired optional feature templates.
i) To create the device template, click **Create**.

Step 5 Attach a device template to a device.

a) From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
b) Click **Device Templates**, and then click **Create Template**.

In Cisco vManage Release 20.7.1 and earlier releases, **Device Templates** is titled **Device**.

c) Choose a template.
d) Click …, and click **Attach Device**.
e) Search for a device or select a device from the Available Device(s) column to the left.
f) Click the arrow pointing right to move the device to the Selected Device(s) column on the right.
g) Click **Attach**.

Configure PPPoE over ATM using templates

Follow these steps to configure PPPoE using the device CLI template in Cisco SD-WAN Manager.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.

Step 2 From **Device Templates**, click **Create Template**.

In Cisco vManage Release 20.7.x and earlier releases, **Device Templates** is titled **Device**.

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

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

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

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

Step 5 Choose **Device configuration**.

Using this option, you can provide IOS-XE configuration commands that appear in the output of the **show sdwan running-config** command.

Step 6 (Optional) To load the running config of a connected device, select it from the Load Running config from reachable device list and click **Search**.

Step 7 In CLI Configuration, enter the configuration either by typing it, cutting and pasting it, or uploading a file.

```
Device(config)# interface atm number
Device(config)# no ip address
Device(config)# interface atm number point-to-point
Device(config)# no atm enable-ilmi-trap
Device(config)# encapsulation aal5mux  pppoe-client
Device(config)# pppoe-client dial-pool-number number
Device(config)# interface Dialer dialer-rotary-group-number
Device(config)# mtu bytes
Device(config)# ip address negotiated
Device(config-if)# encapsulation encapsulation-type
Device(config)# load-interval seconds
Device(config)# dialer pool number
Device(config)# dialer-group group-number
Device(config)# ppp mtu adaptive
Device(config)# ppp chap hostname hostname
Device(config)# ppp chap password secret
Device(config)# ppp ipcp address required
Device(config)# ppp link reorders
```

Step 8 To convert an actual configuration value to a variable, select the value and click Create Variable. Enter the variable name, and click Create Variable. You can also type the variable name directly, in the format {{*variable-name*}}; for example, {{hostname}}.

Step 9 Click **Add**. The new device template is displayed in the Device Template table. The **Type** column shows **CLI** to indicate that the device template was created from CLI text.

Configuration examples for PPPoE

PPPoE server on IPv4 interfaces

This example shows configuring PPPoE server on IPv4 interfaces:

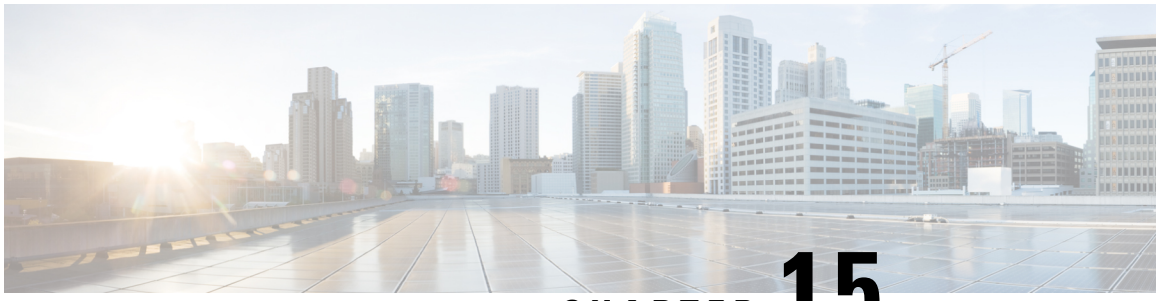
```
!
interface Dialer100
 mtu 1492
 ip address negotiated
 encapsulation ppp
 ip tcp adjust-mss 1460
 dialer pool 100
 dialer down-with-vInterface
 ppp authentication chap callin
 ppp chap hostname cisco
 ppp chap password 7 1511021F07257A767B
 ppp ipcp route default
```

To replace a template configured with PPPoE as WAN interface with a regular interface in Dialer100, remove the IP address assigned to the dialer interface using the **no ip address** command, and then add a new IP address for the dialer interface.

PPPoE over ATM interfaces

This example shows configuring PPPoE over ATM interfaces.

```
Device(config)# interface ATM0/1/0
Device(config)# no ip address
Device(config)# no atm enable-ilmi-trap
!
Device(config)# interface ATM0/1/0.10 point-to-point
Device(config)# no atm enable-ilmi-trap
Device(config)# cdp enable
Device(config)# pvc 22/62
Device(config)# ubr 1045
Device(config-if)# encapsulation aal5mux pppoe-client
Device(config)# pppoe-client dial-pool-number 120
!
!
Device(config)# interface Dialer 120
Device(config)# mtu 1492
Device(config)# ip address negotiated
Device(config)# ip nat outside
Device(config-if)# encapsulation ppp
Device(config)# load-interval 30
Device(config)# dialer pool 120
Device(config)# dialer-group 1
Device(config)# ppp mtu adaptive
Device(config)# ppp chap hostname test@cisco.com
Device(config)# ppp chap password 0 cisco
Device(config)# ppp ipcp address required
Device(config)# ppp link reorders
!
```



CHAPTER 15

TCP MSS and Clear Dont Fragment

Feature history for TCP MSS and clear dont fragment

Table 22: Feature History

Feature Name	Release Information	Description
Configure TCP MSS	Cisco IOS XE Catalyst SD-WAN Release 17.5.1a Cisco vManage Release 20.5.1	This feature adds support for TCP MSS adjustment on Cisco IOS XE Catalyst SD-WAN devices on both directions of the Cisco Catalyst SD-WAN tunnel interface.
Configure Clear Don’t Fragment Option	Cisco IOS XE Catalyst SD-WAN Release 17.5.1a Cisco vManage Release 20.5.1	This feature provides the option to clear the Don't Fragment bit in the IPv4 packet header for packets being sent out on a Cisco Catalyst SD-WAN tunnel . When you clear the Don't Fragment configuration, packets larger than the interface MTU are fragmented before being sent.

TCP MSS

A TCP maximum segment size (MSS) is a parameter that

- specifies the largest amount of data, in bytes, that a communications device can receive in a single TCP segment without counting the TCP header or the IP header, and

- is specified as TCP MSS, initially in the TCP SYN packet during the TCP handshake.

TCP MSS for SYN packets

Small MSS values reduce or eliminate IP fragmentation, which results in higher overhead. You can configure the MSS of TCP SYN packets passing through a device. By default, the device dynamically adjusts the MSS based on the interface or tunnel maximum transmission unit (MTU) to ensure that TCP SYN packets are never fragmented. For data sent over an interface, the device calculates the MSS by adding the interface MTU, the IP header length, and the maximum TCP header length.

Restrictions for TCP MSS and clear dont fragment

TCP MSS

- TCP MSS values can be adjusted for Cisco Catalyst SD-WAN tunnel interfaces only.

  From Cisco IOS XE Catalyst SD-WAN Release 17.9.1aand Cisco vManage Release 20.9.1, you can adjust the TCP MSS value for a service VPN or for Network Address Translation (NAT) Direct Internet Access (DIA) use cases. Adjusting the TCP MSS value helps prevent TCP sessions from being dropped.

  For more information on NAT DIA, see the *Cisco Catalyst SD-WAN NAT Configuration Guide*.

- From Cisco IOS XE Catalyst SD-WAN Release 17.18.1a, for an IPv4 SD-WAN tunnel carrying IPv6 traffic, the system uses the IPv6 TCP MSS value. Also, for an IPv6 tunnel carrying IPv4 traffic, the system uses the IPv4 TCP MSS value. This upgrade aligns MSS adjustment with the traffic protocol and tunnel configurations, ensuring accurate and efficient handling of traffic.

Clear dont fragment

- The option **Clear Dont Fragment** is available for Cisco Catalyst SD-WAN tunnel interfaces only.

Configure TCP MSS and clear dont fragment

Use one of these methods to configure TCP MSS and clear dont fragment:

- CLI commands
- Templates

Configure TCP MSS and clear dont fragment using CLI commands

Follow these steps to configure TCP MSS and clear dont fragment using CLI commands:

Procedure

Step 1 Configure an interface type and enter the interface configuration mode.

Example:

```
interface Tunnel 1
```

Step 2 Enable IP processing on an interface without assigning an explicit IP address to the interface.

Example:

```
ip unnumbered GigabitEthernet1
```

Step 3 Configure TCP MSS and clear dont fragment.

- Enable a maximum segment size (MSS) for TCP connections.

Example:

```
ip tcp adjust-mss 1460
```

- Clear the Don't Fragment (DF) bit in the IPv4 packet header for packets being transmitted out the interface.

Example:

```
ip clear-dont-fragment
```

The following is an example configuration of TCP MSS:

```
Device(config)#interface Tunnel 1
Device(config-if)#ip unnumbered GigabitEthernet1
Device(config-if)#ip tcp adjust-mss 1460
```

The following is an example to configure Clear Dont Fragment option:

```
Device(config)#interface Tunnel 1
Device(config-if)#ip unnumbered GigabitEthernet1
Device(config-if)#ip clear-dont-fragment
```

Configure TCP MSS and clear dont fragment using templates

Follow these steps to configure TCP MSS and clear dont fragment using feature templates.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.

Step 2 Click **Feature Templates**.

Note
In Cisco vManage Release 20.7.1 and earlier releases, **Feature Templates** is titled **Feature**.

Step 3 Create a new CLI add-on feature template or edit one of the following templates. You can use any of the following feature templates to configure TCP MSS and clear dont fragment:

- VPN Ethernet Interface
- VPN Interface DSL IPoE
- VPN Interface DSL PPoA

- VPN Interface DSL PPPoE
- VPN Interface Multilink
- VPN Interface T1/E1
- Cellular Interfaces

Step 4 Click **Tunnel**.

Step 5 Specify the MSS of TPC SYN packets passing through the Cisco IOS XE Catalyst SD-WAN device to configure TCP MSS in the tunnel TCP MSS. By default, the MSS is dynamically adjusted based on the interface or tunnel MTU so that TCP SYN packets are not fragmented.

- Range: 552 to 1460 bytes
- Default: None

TCP MSS affects any packet that contains an initial TCP header that flows through the router. When configured, TCP MSS is examined against the MSS exchanged in the three-way handshake. The MSS in the header is lowered if the configured setting is lower than the MSS in the header. If the MSS header value is already lower than the TCP MSS, it flows through unmodified. The host at the end of the tunnel uses the lower setting of the two hosts. If the TCP MSS is to be configured, it should be set at 40 bytes lower than the minimum path MTU.

Step 6 Click the **Clear-Dont-Fragment** option to clear the Dont Fragment bit in the IPv4 packet header for packets being transmitted out the interface. When the **Don't Fragment** bit is cleared, packets larger than that interface's MTU are fragmented before being sent.

Clear-Dont-Fragment clears the **Don't Fragment** bit when there is fragmentation needed and the **Don't Fragment** bit is set. For packets that don't require fragmentation, the **Don't Fragment** bit is not affected.

Step 7 Click **Save** or **Update**.

Verify TCP MSS and dont clear fragment configurations

TCP MSS configuration

The following is sample output of the **show platform hardware qfp active feature sdwan datapath session summary** command:

```
Device# show platform hardware qfp active feature sdwan datapath session summary

Src IP          Dst IP          Src Port Dst Port   Encap    Uidb     Bfd Discrim PMTU

------          ------          -------- -------    ------   -------  ----------- -----

10.1.15.25      10.1.14.14      12347    12346      IPSEC    65526    10007       1446

10.1.15.25      10.0.5.21       12347    12357      IPSEC    65526    10009       1446

10.1.15.25      10.0.5.11       12347    12347      IPSEC    65526    10008       1446

10.1.15.25      10.1.16.16      12347    12366      IPSEC    65526    10006       1446
```

Dont fragment configuration

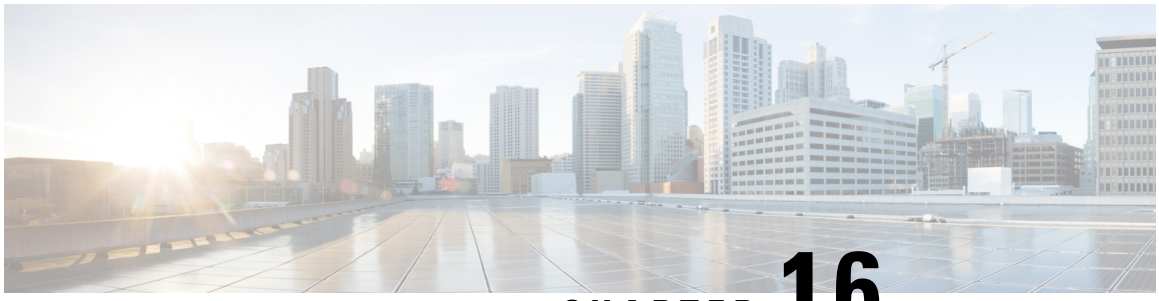
The following is sample output of the **show platform software interface rp active name Tunnel1** command to verify if **Clear-dont-fragment** is enabled or not.

```
Device# show platform software interface rp active name Tunnel1 | include dont
IP Clear-dont-fragment: TRUE
```

The following is sample output of the **show running-config interface Tunnel1** command that displays the running configuration when Clear-dont-fragment is enabled.

```
Device# show running-config interface Tunnel1
Building configuration...

Current configuration : 132 bytes
!
interface Tunnel1
ip unnumbered GigabitEthernet1
ip clear-dont-fragment
tunnel source GigabitEthernet1
tunnel mode sdwan
end
```



CHAPTER 16

Track Static Routes for Service VPNs

Feature history of track static routes for service VPNs

This table describes the developments of this feature, by release.

Table 23: Feature history

Feature Name	Release Information	Description
Static Route Tracker for Service VPNs	Cisco IOS XE Catalyst SD-WAN Release 17.6.1a Cisco vManage Release 20.6.1	This feature enables you to configure IPv4 static route endpoint tracking for service VPNs. For static routes, endpoint tracking determines whether the configured endpoint is reachable before adding that route to the route table of the device.
TCP/UDP Endpoint Tracker and Dual Endpoint Static Route Tracker for Cisco IOS XE Catalyst SD-WAN devices	Cisco IOS XE Catalyst SD-WAN Release 17.9.1a Cisco vManage Release 20.9.1	This feature enables you to configure the TCP/UDP static route endpoint trackers. Using this feature you can also configure IPv4, TCP/UDP dual endpoint static-route tracker groups for service VPNs to enhance the reliability of probes.

Track static routes for service VPNs

Tracking static routes for service VPNs allows you to monitor the reachability of the configured next-hop endpoint IP address. You can verify the endpoint before the device adds the static route to its routing table. This is particularly important in service VPNs where static routes are advertised over the Overlay Management Protocol (OMP). The static route tracker periodically sends Internet Control Message Protocol (ICMP) ping probes to the configured endpoint IP address. If the endpoint is unreachable (no response to probes), the static route is excluded from the routing table and is not advertised to OMP, preventing traffic blackholing. You can configure backup routes with higher administrative distance to provide alternate paths.. This mechanism enhances route reliability and network stability in SD-WAN environments. The tracker sends periodic probes (ICMP echo requests, TCP, or UDP probes) to the static route's next-hop IP address.

- If the endpoint is unreachable, the static route is removed from the routing table and not advertised to OMP.
- Configure backup static routes with higher administrative distance for failover.
- Only one endpoint tracker is supported per static route per next-hop.
- IPv6 static routes are not supported for tracking.
- You can configure dual endpoint tracker groups to enhance probe reliability.
- This feature is supported on Cisco IOS XE Catalyst SD-WAN platforms such as ASR 1000, ISR 1000 and 4000, and CSR 1000 series routers.
- This feature ensures that static routes in service VPNs are advertised only when their next-hop endpoints are reachable. As a result, it prevents traffic blackholing and improves network resilience.

From Cisco IOS XE Catalyst SD-WAN Release 17.7.1a, you can configure TCP or UDP individual endpoint trackers and configure a tracker group with dual endpoints (using two trackers), and associate the trackers and tracker group to a static route. Dual endpoints help avoid false negatives that route unavailability might introduce.

Starting with Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, Cisco SD-WAN Manager reports only UP/DOWN status changes. It does not report RTT value changes. This optimization improves efficiency in extensive networks.

Supported platforms

These devices support static route tracking for service VPNs:

- Cisco ASR 1000 Series Aggregated Services Routers
- Cisco ISR 1000 Series-Integrated Services Routers
- Cisco ISR 4000 Series Integrated Services Routers
- Cisco CSR 1000 Series Cloud Service Routers

Restrictions for IPv4 static route tracking

- Only one endpoint tracker is supported per static route per next-hop address.
- IPv6 static routes are not supported.
- To configure a static route with tracker:
  1. Delete any existing static route, if it is already configured without a tracker. Plan for any connectivity downtime that might occur during this step for static route advertisement.
  2. Configure a new static route with tracker using the same prefix and next-hop as the deleted static route.
- To add a new tracker after you reach maximum tracker limit per router:
  1. Delete an old tracker and attach the template to the device.
  2. Add a new tracker and attach the device to the template again.
- UDP tracker endpoint enabled with IP SLA UDP packet responder is supported only on Cisco IOS XE Catalyst SD-WAN devices.
- You cannot link the same endpoint-tracker to static routes in different VPNs. Endpoint-tracker is identified by a name and can be used for multiple static routes in a single VPN.

Configure tracker group using a configuration group

Before you begin

On the **Configuration** > **Configuration Groups** page, choose **SD-WAN** as the solution type.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration Groups**.

Step 2 Create and configure a tracker group in a Service Profile.

Table 24: Tracker Group

Field	Description
Tracker Elements*	This field is displayed only if you chose **Tracker Type** as the **Tracker Group**. Add the existing interface tracker names, separated with a space. When you add this tracker to the template, the tracker group is associated with these individual trackers, and you can then associate the tracker group to an interface.

Field	Description
Tracker Boolean	This field is displayed only if you chose **Tracker Type** as the **Tracker Group**. Select **AND** or **OR**. **OR** is the default boolean operation. An **OR** ensures that the transport interface status is reported as active if either one of the associated trackers of the tracker group reports that the interface is active. If you select the **AND** operation, the transport-interface status is reported as active if both the associated trackers of the tracker group report that the interface is active.

What to do next

Also see Deploy a configuration group.

Create a static route tracker

Before you begin

Use the **System Template** to create a tracker for static routes.

Delete existing static routes, if any, before you create a static route tracker. Configure a new static route tracker using the same prefix and next hop as the deleted static route.

Procedure

Step 1 From Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.

Step 2 Click **Feature Templates**.

Step 3 Navigate to the **Cisco System** template for the device.

For information about creating a System template, see Create System Template.

Step 4 Click **Tracker**. Click **New Endpoint Tracker** to configure the tracker parameters.

Table 25: Tracker Parameters

Field	Description
Name	Name of the tracker. The name can be up to 128 alphanumeric characters.
Threshold	Wait time for the probe to return a response before declaring that the configured endpoint is down. The range is from 100 to 1000 milliseconds. The default is 300 milliseconds.

Field	Description
Interval	Time interval between probes to determine the status of the configured endpoint. The default is 60 seconds (1 minute). The range is from 20 to 600 seconds.
Multiplier	Number of times probes are sent before declaring that the endpoint is down. The range is from 1 to 10. The default is 3.
Tracker Type	From the drop-down, choose Global. From the Tracker Type field drop-down, choose Static Route. From Cisco IOS XE Catalyst SD-WAN Release 17.7.1a, you can configure a tracker group with dual endpoints on Cisco IOS XE Catalyst SD-WAN devices. You can associate this tracker group to a static route.
Endpoint Type	Choose endpoint type IP address. **Note** Configuring the tracker type Static Route using endpoint URL or endpoint DNS name is not supported.
End-Point Type: IP Address	IP address of the static route end point. This is the destination on the internet to which the router sends probes to determine the status of the route.

Step 5 Click **Add**.

Step 6 Click **Save**.

Complete all mandatory actions before saving the template.

Step 7 To create a tracker group, click **Tracker Groups** > **New Endpoint Tracker Groups** and configure the tracker parameters.

Ensure that you have created two trackers to form a tracker group.

Table 26: Tracker Group Parameters

Fields	Description
Name	Name of the tracker group.
Tracker Type	From the drop-down, choose **Global**. From the Tracker Type field drop-down, choose **Static Route**. From Cisco IOS XE Catalyst SD-WAN Release 17.7.1a, you can configure a tracker group with dual endpoints on Cisco IOS XE Catalyst SD-WAN devices and associate this tracker group to a static route.
Tracker Elements	This field is displayed only if you chose **Tracker-group** as the tracker type. Add the existing interface tracker names (separated by a space). When you add this tracker to the template, the tracker group is associated with these individual trackers, and you can then associate the tracker group to a static route.

Fields	Description
Tracker Boolean	From the drop-down list, choose **Global**. This field is displayed only if you chose **tracker-group** as the **Tracker Type**. By default, the **OR** option is selected. Choose **AND** or **OR**. **OR** ensures that the static route status is reported as active if either one of the associated trackers of the tracker group report that the route is active. If you select **AND**, the static route status is reported as active if both the associated trackers of the tracker group report that the route is active.

Step 8 Click **Add**.

Step 9 Click **Save**.

Complete all mandatory actions before saving the template.

Configure a next hop static route with tracker

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.

Step 2 Click **Feature Templates** and navigate to **Cisco VPN Template** for the device.

For information about creating a VPN template, see Create VPN Template.

Step 3 Enter **Template Name** and **Description** as required.

Step 4 In Basic Configuration, by default, VPN is set to 0. Set a VPN value within (1–511, 513–65530) range for service VPNs, for service-side data traffic on Cisco IOS XE Catalyst SD-WAN devices.

You can configure static route tracker only on service VPNs.

Step 5 Click **IPv4 Route** and **New IPv4 Route**.

Step 6 In the **IPv4 Prefix** field, enter a value.

Step 7 Click **Next Hop**. Click **Add Next Hop** with **Tracker** and enter values for the fields listed in the table.

Parameter Name	Description
Address	Specify the next-hop IPv4 address.
Distance	Specify the administrative distance for the route.
Tracker	Enter the name of the gateway tracker to determine whether the next hop is reachable before adding that route to the route table of the device.

Parameter Name	Description
Add Next Hop with Tracker.	Enter the name of the gateway tracker with the next hop address to determine whether the next hop is reachable before adding that route to the route table of the device.

Step 8 Click **Add** and **Save** to create the static route with the next-hop tracker.

You need to fill all the mandatory fields in the form to save the VPN template.

Monitor static route tracker configuration

To view information about a static tracker on a transport interface:

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Monitor** > **Devices**.

Step 2 Choose a device from the list of devices.

Step 3 Click **Real Time**.

Step 4 From the **Device Options** drop-down list, choose **Endpoint Tracker Info**.

Configure static routes using CLI

These sections provide information about how to configure static routes using the CLI.

You can configure static route tracking using the Cisco SD-WAN Manager CLI Add-on feature templates and CLI device templates. For more information on configuring using CLI templates, see CLI Templates.

Configure a static route tracker

```
Device# config-transaction
Device(config)# endpoint-tracker <tracker-name>
Device(config-endpoint-tracker)# tracker-type <tracker-type>
Device(config-endpoint-tracker)# endpoint-ip <ip-address>
Device(config-endpoint-tracker)# threshold <value>
Device(config-endpoint-tracker)# multiplier <value>
Device(config-endpoint-tracker)# interval <value>
Device(config-endpoint-tracker)# exit
Device(config)# track <tracker-name> endpoint-tracker
```

Configure a static route tracker with TCP port as the endpoint

```
Device# config-transaction
Device(config)# endpoint-tracker <tracker-name>
Device(config-endpoint-tracker)# tracker-type <tracker-type>
Device(config-endpoint-tracker)# endpoint-ip <ip-address> tcp <port-number>
```

```
Device(config-endpoint-tracker)# threshold <value>
Device(config-endpoint-tracker)# multiplier <value>
Device(config-endpoint-tracker)# interval <value>
Device(config-endpoint-tracker)# exit
Device(config)# track <tracker-name> endpoint-tracker
```

Configure a static route tracker with UDP port as the endpoint

```
Device# config-transaction
Device(config)# endpoint-tracker <tracker-name>
Device(config-endpoint-tracker)# tracker-type <tracker-type>
Device(config-endpoint-tracker)# endpoint-ip <ip-address> udp <port-number>
Device(config-endpoint-tracker)# threshold <value>
Device(config-endpoint-tracker)# multiplier <value>
Device(config-endpoint-tracker)# interval <value>
Device(config-endpoint-tracker)# exit
Device(config)# track <tracker-name> endpoint-tracker
```

Configure tracker groups

You can create tracker groups to probe static routes from Cisco IOS XE Catalyst SD-WAN Release 17.7.1a and Cisco vManage Release 20.7.1.

```
Device# config-transaction
Device(config)# endpoint-tracker <tracker-name1>
Device(config-endpoint-tracker)# tracker-type <tracker-type>
Device(config-endpoint-tracker)# endpoint-ip <ip-address> tcp <port-number>
Device(config-endpoint-tracker)# threshold <value>
Device(config-endpoint-tracker)# multiplier <value>
Device(config-endpoint-tracker)# interval <value>
Device(config-endpoint-tracker)# exit
Device(config)# track <tracker-name1> endpoint-tracker


Device# config-transaction
Device(config)# endpoint-tracker <tracker-name2>
Device(config-endpoint-tracker)# tracker-type <tracker-type>
Device(config-endpoint-tracker)# endpoint-ip <ip-address> udp <port-number>
Device(config-endpoint-tracker)# threshold <value>
Device(config-endpoint-tracker)# multiplier <value>
Device(config-endpoint-tracker)# interval <value>
Device(config-endpoint-tracker)# exit
Device(config)# track <tracker-name2> endpoint-tracker


Device(config)# endpoint-tracker <static-tracker-group>
Device(config-endpoint-tracker)# tracker-type tracker-group
Device(config-endpoint-tracker)# tracker-elements <tracker-name1> <tracker-name2>
Device(config-endpoint-tracker)# boolean {and | or}
Device(config-endpoint-tracker)# exit
Device(config)# track <static-tracker-group> endpoint-tracker

Device(config)# ip route vrf <vrf-name> <prefix> <mask> <nexthop-ipaddress>
<administrative-distance> track name <static-tracker-group>
```

Note

- Use the **ip route** command to bind a tracker or tracker group with a static route and to configure a backup route for administrative distance that is higher than the default value of 1.
- You can apply only one tracker to an endpoint.
- A tracker group can have a mix of endpoint trackers. For example, you can create a tracker group with an IP address tracker and UDP tracker.

Configuration examples static route tracking

Configure Tracker

This example shows how to configure a single static route tracker:

```
config-transaction
!
 endpoint-tracker tracker1
!
  tracker-type static-route
  endpoint-ip 10.1.1.1
  threshold 100
  multiplier 5
  interval 20
  exit
!
track tracker1 endpoint-tracker
!
ip route vrf 1 192.168.0.0 255.255.0.0 10.1.19.16 100 track name tracker1
```

This example shows how to configure a tracker with TCP port as endpoint:

```
config-transaction
!
 endpoint-tracker tcp-10001
!
  tracker-type static-route
  endpoint-ip 10.0.0.1 tcp 10001
  threshold    100
  interval     10
  multiplier   1
  exit
!
track tcp-10001 endpoint-tracker
!
ip route vrf 1 192.168.0.0 255.255.0.0 10.1.19.16 100 track name tcp-10001
```

This example shows how to configure a tracker with UDP port as endpoint:

```
config-transaction
!
  endpoint-tracker udp-10001
!
    tracker-type static-route
    endpoint-ip 10.0.0.1 udp 10001
```

```
    threshold    100
    interval     10
    multiplier   1
    exit
!
track udp-10001 endpoint-tracker
!
ip route vrf 1 192.168.0.0 255.255.0.0 10.1.19.16 100 track name udp-10001
```

Configure Tracker Groups

This example shows how to configure a tracker group with two trackers (two endpoints). You can create tracker groups to probe static routes.

```
config-transaction
!
 endpoint-tracker tcp-10001
!
    tracker-type static-route
    endpoint-ip 10.1.1.1 tcp 10001
    threshold 100
    multiplier 5
    interval 20
    track tcp-10001 endpoint-tracker
!
 endpoint-tracker udp-10002
!
   tracker-type static-route
   endpoint-ip 10.2.2.2 udp 10002
   threshold 100
   multiplier 5
   interval 20
   track udp-10002 endpoint-tracker
!
endpoint-tracker static-tracker-group
!
  tracker-type tracker-group
  tracker-elements tcp-10001 udp-10002
  boolean and
  track static-tracker-group endpoint-tracker
!
ip route vrf 1 192.168.0.0 255.255.0.0 10.1.19.16 100 track name static-tracker-group
```

Note

- You must configure an administrative distance when you are configuring through CLI templates.
- Use the **ip route** command to bind the tracker or tracker group with a static route and to configure a backup route for administrative distance when it is higher than the default value of 1.
- You can apply only one tracker to an endpoint.

Verify static route tracking configuration

Use these commands to verify if the configuration is committed. The sample configuration shows tracker definition for a static route tracker and it's application to an IPv4 static route:

```
Device# show running-config | sec endpoint-tracker
endpoint-tracker tracker1
endpoint-ip 10.1.1.1
interval 60
multiplier 5
tracker-type static-route
endpoint-tracker tracker2
endpoint-ip 10.1.1.12
interval 40
multiplier 2
tracker-type static-route
track tracker2 endpoint-tracker
track tracker1 endpoint-tracker
```

Use this command to verify the IPv4 route:

```
Device# show running-config | inc ip route
ip route vrf 1 10.1.1.11 255.255.0.0 10.20.2.17 track name tracker2
ip route vrf 1 10.1.1.12 255.255.0.0 10.20.24.17 track name tracker1
```

The sample output from the **show endpoint-tracker static-route** command displaying individual static route tracker status:

```
Device#  show endpoint-tracker static-route
Tracker Name   Status     RTT (in msec) Probe ID
tcp-10001         UP         3             1
udp-10002         UP         1             6
```

The sample output from the **show endpoint-tracker tracker-group** command displaying tracker group status:

```
Device# show endpoint-tracker group
Tracker Name               Element trackers name    Status         RTT in msec  Probe ID

group-tcp-10001-udp-10002  tcp-10001, udp-10002     UP(UP AND UP)  5, 1          9, 10
```

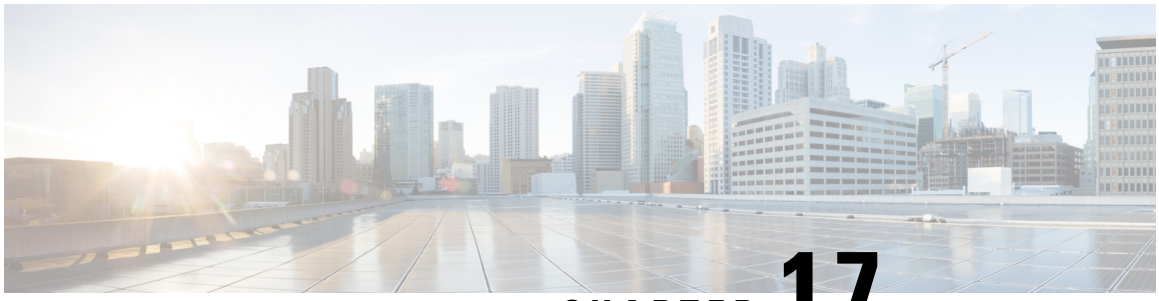
The sample output from the **show endpoint-tracker records** command displaying tracker/tracker group configuration:

```
Device# show endpoint-tracker records
Record Name                Endpoint                 EndPoint Type Threshold(ms) Multiplier
 Interval(s) Tracker-Type
group-tcp-10001-udp-10002  tcp-10001 AND udp-10002  N/A           N/A           N/A
 N/A         static-tracker-group
tcp-10001                  10.1.1.1                 TCP           100           1
 20          static-route
udp-10002                  10.2.2.2                 UDP           100           1
 20          static-route
```

The sample output from the **show ip static route vrf** command:

```
Device# show ip static route vrf 1
Codes: M - Manual static, A - AAA download, N - IP NAT, D - DHCP,
       G - GPRS, V - Crypto VPN, C - CASA, P - Channel interface processor,
       B - BootP, S - Service selection gateway
       DN - Default Network, T - Tracking object
       L - TL1, E - OER, I - iEdge
       D1 - Dot1x Vlan Network, K - MWAM Route
       PP - PPP default route, MR - MRIPv6, SS - SSLVPN
       H - IPe Host, ID - IPe Domain Broadcast
       U - User GPRS, TE - MPLS Traffic-eng, LI - LIIN
       IR - ICMP Redirect, Vx - VXLAN static route
       LT - Cellular LTE, Ev - L2EVPN static route
Codes in []: A - active, N - non-active, B - BFD-tracked, D - Not Tracked, P - permanent,
-T Default Track
Codes in (): UP - up, DN - Down, AD-DN - Admin-Down, DL - Deleted
```

```
Static local RIB for 1
T  192.168.0.0 [1/0] via 10.1.19.16 [A]
```



CHAPTER 17

VDSL and GSHDSL

VDSL and GSHDSL

Very High-Speed Digital Subscriber Line (VDSL) is a digital subscriber line (DSL) technology that provides high-speed data transmission over existing copper telephone wires. It is an advanced version of earlier DSL standards like ADSL, offering significantly faster speeds by utilizing higher frequency ranges

Symmetric High-Speed Digital Subscriber Line (G.SHDSL) is a data communications technology designed for symmetric high-speed data transmission over copper twisted pairs. While the query mentioned G.HDSL, G.SHDSL (standardized as ITU-T G.991.2) is the more modern and widely adopted international standard for symmetric DSL. G.HDSL (G.991.1) is an older standard.

Key characteristics of VDSL

- Speed: VDSL can offer speeds up to 52 Mbit/s downstream and 16 Mbit/s upstream. Its successor, VDSL2 (ITU-T G.993.2), can provide data rates exceeding 100 Mbit/s simultaneously in both upstream and downstream directions, and VDSL2+ can reach 300+ Mbit/s.
- Technology: It uses frequencies up to 12 MHz (VDSL) or up to 30 MHz (VDSL2) to achieve these higher speeds.
- Applications: VDSL is capable of supporting applications such as high-definition television (HDTV), voice over IP (VoIP), and general high-speed internet access over a single connection.
- Distance Sensitivity: The performance of VDSL degrades as the distance from the service provider's equipment increases.

For related information, see VDSL Commands.

Key characteristics of G.SHDSL

G.SHDSL is an international standard that allows devices to send and receive high-speed symmetrical data streams over a single pair of copper wires. This section provides information about the Cisco G.SHDSL EFM/ATM NIM and provides guidelines for configuring G.SHDSL in SD-WAN mode.

- Symmetry: Unlike asymmetric DSL (ADSL), G.SHDSL provides symmetrical transmit and receive data rates, meaning the upload and download speeds are equal.
- Speed: It offers symmetric rates typically ranging from 192 kbit/s to 15296 kbit/s on a single twisted pair.
- Applications: G.SHDSL is often used for high-speed commercial broadband services, enabling applications that require significant data transfer in both directions, such as LAN remote access, web hosting, and combining multiple voice and data channels.
- Compatibility: It is compatible with other DSL technologies and can extend transmission distances while maintaining high data rates.

For related information, see Configuring Cisco G.SHDSL HWICs in Cisco Access Routers and VDSL Commands.

VDSL configuration guidelines

This table provides usage information and guidelines for configuring asymmetric DSL (ADSL2/2+) and VDSL for supported Integrated Services Router Network Interface Modules (ISR NIMs) in SD-WAN mode. VDSL2 and ADSL2/2+ provide highly reliable WAN connections for remote sites.

Function	Command	Guidelines
Configure operating mode	Device# **configure terminal** Device(config)# **controller VDSL** *slot/subslot/port* Device(config)# **operating mode auto**	To switch from operating mode auto adsl1 (adsl2+/ or vdsl2) to operating mode auto ads2+ (adsl1 or vdsl2), switch to operating mode auto first. Before you change the operating mode, ensure that line-mode is changed to line-mode single-wire line 0.
Enable DSL on a line	Device(config)# **line-mode single-wire line** *line-number*	This command is supported only on DSL NIM-VAB-A.
Enable bonding	Device(config)# **line-mode bonding**	This command is supported only on DSL NIM-VAB-A.

Function	Command	Guidelines
Load firmware on a device	Device# **configure terminal** Device(config)# **controller VDSL** *slot* / *subslot* / *port* Device(config-controller)# **firmware phy filename** *filename*	The Cisco Catalyst SD-WAN CLI template does not support specifying the file location. Prepend the file name with flash: or with bootflash:, depending on its location.
Enable or disable SRA	Device(config-controller)# **sra**	The Cisco Catalyst SD-WAN CLI template does not support the sra *line number* command. In line-mode bonding, sra enables sra on both lines and no sra disables sra on both lines.
Enable or disable bitswap	Device(config-controller)# **bitswap**	The Cisco Catalyst SD-WAN CLI template does not support the bitswap *line number* command. In line-mode bonding, bitswap enables bitswap on both lines and no bitswap disables bitswap on both lines.
Enable modem features	Device(config-controller)# **modem** *keyword*	–
Display a description of a controller	Device(config-controller)# **description** *string*	–
Enable dual ended line testing	Device(config-controller)# **diagnostics DELT**	–
Modify the file in which the training log is stored	Device(config-controller)# **training log filename flash:** *filename*	The Cisco Catalyst SD-WAN CLI template does not support specifying the file location. Prepend the file name with flash: or with bootflash:, depending where the file should be stored.
Enable sync mode	Device(config-controller)# **sync mode** *mode*	To switch from one sync mode to another, delete the existing sync mode, then configure the new one.
Enable sync interval	Device(config-controller)# **sync interval** *seconds*	–

Cisco VDSL examples

Configuration example for VDSL:

```
Device# config-transaction
Device(config)# controller VDSL 0/0/0
Device(config)# operating mode auto


Device# config-transaction
Device(config)# line-mode single-wire line 1


Device# config-transaction
Device(config)# line-mode bonding


Device# config-transaction
Device(config)# controller VDSL 0/0/0
Device(config-controller)# firmware phy filename flash:IDC_1.7.2.6_DFE_FW_BETA_120111A.pkg


Device# config-transaction
Device(config-controller)# sra


Device# config-transaction
Device(config-controller)# bitswap


Device# config-transaction
Device(config)# controller VDSL 0/0/0
Device(config-controller)# modem customUKAnnexM


Device# config-transaction
Device(config)# controller VDSL 0/0/0
Device(config-controller)# description to ISP 1


Device# config-transaction
Device(config)# controller VDSL 0/0/0
Device(config-controller)# diagnostics DELT


Device# config-transaction
Device(config)# controller VDSL 0/0/0
Device(config-controller)# training log filename bootflash:VDSLLOG.log



Device# config-transaction
Device(config)# controller VDSL 0/0/0
Device(config-controller)# sync mode ansi previous



Device# configure terminal
Device(config)# ptp clock ordinary domain 0
Device(config-ptp-clk)# clock-port slave slaveport
Device(config-ptp-port)# sync interval -4
Device(config-ptp-port)# end
```

The following example show the VDSL configuration:

```
Device(config)# show controllers vdSL 0/2/0
Controller VDSL 0/2/0 is UP

Daemon Status:          UP

                        XTU-R (DS)              XTU-C (US)
```

```
Chip Vendor ID:          'BDCM'                   'BDCM'
Chip Vendor Specific:    0x0000                   0xA39A
Chip Vendor Country:     0xB500                   0xB500
Modem Vendor ID:         'CSCO'                   'BDCM'
Modem Vendor Specific:   0x4602                   0x0000
Modem Vendor Country:    0xB500                   0xB500
Serial Number Near:      FGL2149956Y C1117-4P 16.7.20180
Serial Number Far:
Modem Version Near:      16.7.20180709:09395
Modem Version Far:       0xa39a

Modem Status:            TC Sync (Showtime!)
DSL Config Mode:         AUTO
Trained Mode:            G.993.2 (VDSL2) Profile 17a

TC Mode:                 PTM
Selftest Result:         0x00
DELT configuration:      disabled
DELT state:              not running

Failed full inits:       0
Short inits:             0
Failed short inits:      0

Modem FW  Version:       4.14L.04
Modem PHY Version:       A2pv6F039t.d26d

Line 0:

                         XTU-R (DS)               XTU-C (US)
Trellis:                  ON                       ON
SRA:                      enabled                  enabled
SRA count:                0                        0
Bit swap:                 enabled                  enabled
Bit swap count:           1                        3
Line Attenuation:         18.4 dB                   0.0 dB
Signal Attenuation:        0.0 dB                   0.0 dB
Noise Margin:              5.2 dB                  6.0 dB
Attainable Rate:         46022 kbits/s            18866 kbits/s
Actual Power:            14.5 dBm                 10.4 dBm
Per Band Status:         D1      D2      D3      U0      U1      U2      U3
Line Attenuation(dB):    13.9    32.7    50.1    N/A     25.6    37.7    42.3
Signal Attenuation(dB):  13.5    32.4    N/A     N/A     25.0    36.9    41.9
Noise Margin(dB):        5.3     5.1     N/A     N/A     6.0     6.0     5.9
Total FECC:              446                      0
Total ES:                3                        0
Total SES:               0                        0
Total LOSS:              0                        0
Total UAS:               50                       50
Total LPRS:              0                        0
Total LOFS:              0                        0
Total LOLS:              0                        0


                    DS Channel1     DS Channel0   US Channel1       US Channel0
Speed (kbps):              NA           47610            NA             18859
SRA Previous Speed:        NA               0            NA                 0
Previous Speed:            NA               0            NA                 0
Reed-Solomon EC:           NA             446            NA                 0
CRC Errors:                NA              51            NA                 0
Header Errors:             NA            3935            NA                 0
Interleave (ms):           NA            1.00            NA              1.00
Actual INP:                NA            0.00            NA              0.00
```

```
Training Log :  Stopped
Training Log Filename : flash:vdsllog.bin
```

GSHDSL configuration guidelines

This table provides usage information and guidelines that apply when you configure the Cisco G.SHDSL EFM/ATM in CPE or CO mode.

Function	Command	Guidelines
Configure a device with the dsl-group auto command	Device(config-controller)# **dsl-group auto**	Use customer premises equipment (CPE) mode when configuring a device with the dsl-group auto command. If you use this command in Central Office (CO) mode, the configuration does not take effect.
Add or delete a link	—	The efm-grp command is not supported. To add or delete a link to a dsl-group, delete the dsl-group, then create a new dsl-group.
Load firmware on a device	Device(config-controller)# **firmware phy filename** *location*	File name location options are not supported when using the firmware phy command. Prepend the file name with flash: or with bootflash:, depending on the location.
Create or delete an annex	Device(config-controller-dsl-group)# **no shdsl annex** Device(config-controller-dsl-group)# **no shdsl rate** *rate*	To avoid Cisco IOS and Cisco Catalyst SD-WAN configuration from going out of sync when you create or delete an annex, create or delete the rate in the same transaction.
Enable SHDSL to use enhanced mode	(config-controller-dsl-group)# **shdsl 4-wire mode enhanced**	To enable SHDSL to use the enhanced mode in a 2-pair digital subscriber line (DSL) group, use the shdsl 4-wire mode enhanced command in configuration controller DSL group mode.
Ignore CRC errors	(config-controller-dsl-group)# **ignore** *seconds*	To configure a device to ignore CRC errors, use the ignore command. Replace *timeout* with a value from 0 through 60, which indicates the number of seconds that the device ignores CRC errors that do not resolve before the device terminates an action.

Function	Command	Guidelines
Shutdown a DSL group	(config-controller-dsl-group)# **shutdown**	To shut down a DSL group, use the shutdown command.

GSHDSL EFM-ATM NIM

G.SHDSL EFM/ATM NIM

The Cisco G.SHDSL EFM/ATM NIM connects Cisco 4000 Series Integrated Services Routers with central office Digital Subscriber Line Access Multiplexers (DSLAMs) and supports up to four DSL pairs. The DSL pairs are bundled in groups and configured in the Cisco IOS CLI by using the dsl-group command. Use the mode command to choose the mode (ATM or EFM).

The NIM supports the following configuration:

- You can configure up to four DSL groups.
- You can configure auto mode on only one DSL group. For example, DSL group 0.
- In ATM Mode, you can configure the lines to use 2-wire, 4-wire (standard or enhanced), or m-pair.
- In EFM mode, you can configure a DSL group with any one of the lines in 2-wire non-bonding mode or with multiple lines in bonding mode.
- Depending on the mode (ATM or EFM), the corresponding interface (ATM or EFM) is automatically created.

Cisco GSHDSL examples

Configuration example for GSHDSL:

```
Device# config-transaction
Device(config)# controller SHDSL 0/0/0
Device(config-controller)# dsl-group auto


Device# config-transaction
Device(config)# controller VDSL 0/0/0
Device(config-controller)# firmware phy filename bootflash:IDC_1.1.1.0_DFE_1.1-1.8.1__001.pkg



Device# config-transaction
Device(config)# controller SHDSL 0/0/0
Device(config-controller)# dsl-group 0 pairs 0
Device(config-controller-dsl-group)# no shdsl annex
Device(config-controller-dsl-group)# no shdsl rate 5696


Device# config-transaction
Device(config)# controller SHDSL 0/0/0
Device(config-controller)# termination cpe
Device(config-controller)# dsl-group 0 pairs 0
```

```
(config-controller-dsl-group)# shdsl 4-wire mode enhanced


Device# config-transaction
Device(config)# controller SHDSL 0/0/0
Device(config-controller)# termination cpe
Device(config-controller)# dsl-group 0 pairs 0
config-controller-dsl-group)# ignore 30


Device# config-transaction
Device(config)# controller SHDSL 0/0/0
Device(config-controller)# termination cpe
Device(config-controller)# dsl-group 0 pairs 0
config-controller-dsl-group)# shutdown
```

The following example show the GSHDSL configuration:

```
Device# show controllers shDSL 0/1/0
Controller SHDSL 0/1/0 is UP
  Hardware is NIM-SHDSL-EA, on slot 0,bay 0
  Capabilities: EFM: 2-wire, EFM-Bond, Annex A, B, F & G
                ATM: 2-wire, Mpair, Annex A, B, F & G
  CPE termination
  cdb=0x7F7EB723D8A8
  Vendor: Intel, Chipset: SOCRATES-4e
  PHY Source: System
  IDC Firmware version: 0.0.0.0
  DFE Firmware version:
  Group 0 info:
        Type: EFM Auto status: Down
         Ethernet Interface: Ethernet0/1/0, hwidb: 0x7F7EB723B648
         ATM Interface: ATM0/1/0, hwidb: 0x7F7EB724CE08
        Configured/active num links: 4/0, bit map: 0xF/0x0
        Line termination: CPE, Annex: auto
        PMMS disabled,Line coding: AUTO-TCPAM
        Configured/actual rate: AUTO/0 kbps
        Dying Gasp: Present
        SHDSL wire-pair (0) is in DSL DOWN state
                LOSWS Defect alarm: none
                SNR Margin alarm: none
                Loop Attenuation alarm: none
                Termination: CPE, Line mode: EFM Auto, Annex: auto
                Line coding: AUTO-TCPAM
                Configured/actual rate: AUTO/0 kbps
                Modem status: DOWN_NOT_READY,Condition: NO_COND_
        DSL Stats:
                Power Back Off: 0dB
                LoopAttn: 0dB, SnrMargin: 0dB
                Current 15 minute statistics (Time elapased 1 seconds)
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Previous 15 minute statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Current 24 hr statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Previous 24 hr statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
        EFM Stats:
                EFM-TC Tx: data frames: 0
                EFM-TC Rx: data frames: 0
        SHDSL wire-pair (1) is in DSL DOWN state
                LOSWS Defect alarm: none
                SNR Margin alarm: none
                Loop Attenuation alarm: none
                Termination: CPE, Line mode: EFM Auto, Annex: auto
```

```
                Line coding: AUTO-TCPAM
                Configured/actual rate: AUTO/0 kbps
                Modem status: DOWN_NOT_READY,Condition: NO_COND_
        DSL Stats:
                Power Back Off: 0dB
                LoopAttn: 0dB, SnrMargin: 0dB
                Current 15 minute statistics (Time elapased 1 seconds)
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Previous 15 minute statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Current 24 hr statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Previous 24 hr statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
        EFM Stats:
                EFM-TC Tx: data frames: 0
                EFM-TC Rx: data frames: 0
        SHDSL wire-pair (2) is in DSL DOWN state
                LOSWS Defect alarm: none
                SNR Margin alarm: none
                Loop Attenuation alarm: none
                Termination: CPE, Line mode: EFM Auto, Annex: auto
                Line coding: AUTO-TCPAM
                Configured/actual rate: AUTO/0 kbps
                Modem status: DOWN_NOT_READY,Condition: NO_COND_
        DSL Stats:
                Power Back Off: 0dB
                LoopAttn: 0dB, SnrMargin: 0dB
                Current 15 minute statistics (Time elapased 1 seconds)
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Previous 15 minute statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Current 24 hr statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Previous 24 hr statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
        EFM Stats:
                EFM-TC Tx: data frames: 0
                EFM-TC Rx: data frames: 0
        SHDSL wire-pair (3) is in DSL DOWN state
                LOSWS Defect alarm: none
                SNR Margin alarm: none
                Loop Attenuation alarm: none
                Termination: CPE, Line mode: EFM Auto, Annex: auto
                Line coding: AUTO-TCPAM
                Configured/actual rate: AUTO/0 kbps
                Modem status: DOWN_NOT_READY,Condition: NO_COND_
        DSL Stats:
                Power Back Off: 0dB
                LoopAttn: 0dB, SnrMargin: 0dB
                Current 15 minute statistics (Time elapased 1 seconds)
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Previous 15 minute statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Current 24 hr statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Previous 24 hr statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
        EFM Stats:
                EFM-TC Tx: data frames: 0
                EFM-TC Rx: data frames: 0
Group 1 is not configured
Group 2 is not configured
Group 3 is not configured
```

CHAPTER 18

VFR and Underlay Fragmentation

Feature history for VFR and underlay fragmentation

Table 27: Feature History

Feature Name	Release Information	Description
VFR and Underlay Fragmentation	Cisco IOS XE Catalyst SD-WAN Release 17.12.1a Cisco Catalyst SD-WAN Manager Release 20.12.1	In Cisco Catalyst SD-WAN networks, the VFR (Virtual Fragmentation Reassembly) actively fragments and reassembles packets. The packets undergo fragmentation to improve transportation efficiency while passing through a VFR-enabled Cisco IOS XE Catalyst SD-WAN device. The VFR reassembles the fragmented packets to match the original incoming packet. The reassembled packet contains critical Layer 4 or Layer 7 information necessary for proper reception by the destination device. Underlay fragmentation refers to the process of breaking down a large data packet into smaller fragments at the network layer. Underlay fragmentation allows the successful transmission of packets that exceed the MTU limitations by breaking them down into manageable fragments and ensuring their reliable delivery.

Virtual Fragmentation Reassembly

A Virtual Fragmentation Reassembly (VFR) is a Cisco feature that

- allows a router to collect IP fragments even if they lack necessary Layer 4 information,
- performs dynamic access control list (ACL) analysis for security, and
- enables security features such as Cisco IOS Firewall and NAT to inspect traffic.

While transmitting data across a network, due to various network constraints, the original data packets fragment into smaller fragments to facilitate seamless transmission. While the packets travel through the Cisco IOS XE Catalyst SD-WAN device, they are fragmented. VFR allows fragmented packets to be reassembled efficiently before reaching their destination.

Packet reassembly modes

In Cisco Catalyst SD-WAN network, data packets undergo reassembly in two modes:

- Default mode: Packets are virtually reassembled by default. Upon the delivery of the first fragment, each feature in the network receives the entire payload of the virtually reassembled packet. When the last fragment is received, the remaining features reassemble the packet. The original packet is fragmented, and the internal fragment information structure is shared. The fragments are then queued for refragmentation based on the fragment-offset sequence. The VFR mechanism reconstructs the packets using information from the fragment headers, such as fragment identifiers, sequence numbers, and offsets.
- Reassembly mode: Packets undergo physical reassembly, and fragment header information isn't saved. Upon receiving the last fragment, the fragments reassemble via a metapacket, and the internal fragment information structure is released.

If the packets were originally fragmented using the default mode, they undergo reassembly as if they were the original incoming packets. On the other hand, when the reassembly mode is utilized to virtually fragment the packets, they experience fragmentation based on the MTU of the egress interface before reassembly.

Some features (such as NAT, Cisco IOS XE Firewall, IPSec) automatically enable VFR to obtain Layer 4 or Layer 7 information.

When a particular interface enables VFR, it overrides the existing firewall or NAT's VFR mode configuration by default, ensuring interoperability with the firewall or NAT.

Underlay fragmentation

An underlay fragmentation is a network layer process that

- breaks down large data packets that exceed the Maximum Transmission Unit (MTU) size supported by the Cisco Catalyst SD-WAN network infrastructure,
- enables the transmission of packets that exceed MTU limitations by fragmenting them into smaller parts, and
- ensures the successful delivery of these fragments across the network.

Benefits of VFR and underlay fragmentation

- VFR enables the Cisco IOS XE Firewall to create appropriate dynamic access control lists (ACLs) to protect the network from various fragmentation attacks.
- VFR is responsible for detecting and preventing various types of fragment attacks.
- VFR drops all fragments within a fragment chain if an overlap of a fragment is detected.

Prerequisites for configuring VFR and underlay fragmentation

Properly configure the Maximum Transmission Unit (MTU) size on all network devices. The MTU defines the maximum packet size that can be transmitted without fragmentation. Ensure the MTU is set appropriately on every device along the network path to avoid unintended underlay fragmentation.

Restrictions for configuring VFR and underlay fragmentation

Fragment handling requirements

- The VFR process requires all fragments within an IP datagram to be present. If load balancing causes fragments to be sent to different devices, VFR may fail and drop fragments.
- If any fragments in a series of fragmented packets are lost or arrive out of order, the reassembly process may fail, resulting in incomplete or corrupted packets.

Feature integration

- VFR is designed to operate with features that require fragment reassembly, such as Cisco Catalyst SD-WAN NAT and IPsec. By default, NAT, Crypto-based IPSec, and NAT64 internally enable or disable VFR on an interface when these features are activated. If multiple features enable VFR on the same interface, VFR uses a reference count to track the number of enabling features. VFR is automatically disabled when the reference count reaches zero.
- The VFR CLIs are unavailable under port-channel sub-interfaces.

Underlay fragmentation

- The underlay fragmentation mechanism operates only at the network layer and is limited to the underlying network infrastructure. It does not manage fragmentation and reassembly across multiple network segments or provide end-to-end fragmentation handling.

Use cases for VFR and underlay fragmentation

Networks such as long-distance connections such as a connection between an airplane and aiport signal towers, can experience interruptions, due to the time it takes for large packets to traverse these links. When VFR is enabled, the fragments reassemble into a complete datagram, and then are fragmented within the Cisco Catalyst SD-WAN tunnel interface. With this, the first fragment is sent out first and there is no interruption in receiving the packets.

Underlay fragmentation helps in fragmenting large packets into smaller sizes, and reconstruct the packet back into the original one. This improves the overall application performance.

Boost mode

The boost mode helps in resolving one of the identified bottlenecks related to the memory management of fragments within the data plane of the network.

The boost mode is disabled by default on Cisco IOS XE Catalyst SD-WAN devices.

Before Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Cisco IOS XE Catalyst SD-WAN Release 17.12.1a and later
The memory allocation to reassembly of fragments occured from a global chunk, necessitating a lock in period for the memory until the reassembly is complete. This leads to potential competition among multiple threads for the same global chunk and results in waiting for the same memory.	The boost mode enhances performance by utilizing CVLA, an alternative data plane memory infrastructure. Unlike the chunk mechanism, CVLA is lock-free and is an efficient memory management mechanism within Cisco IOS XE devices.

Configure VFR and underlay fragmentation

Use one of these methods to configure VFR and underlay fragmentation:

- Configuration groups
- CLI commands

Configure underlay fragmentation using a configuration group

Before you begin

Minimum supported releases: Cisco IOS XE Catalyst SD-WAN Release 17.15.1a

Procedure

Step 1 From the Cisco SD-WAN Manager, choose **Configuration** > **Configuration Groups**.
Step 2 Click **Transport & Management Profile**.
Step 3 Select the desired transport profile and click **Edit**.
Step 4 Click **Edit Ethernet Interface** > **Tunnel**.
Step 5 Enable **Allow Fragmentation** and **MTU To Max**.
Step 6 Click **Save**.

Configure VFR using CLI commands

Enable virtual fragment reassembly (VFR) on interfaces to prevent fragmentation attacks and ensure correct packet delivery.

You can configure VFR using a CLI template. For more information about using CLI templates, see *CLI Add-On Feature Templates* and *CLI Templates*.

Note By default, CLI templates execute commands in global config mode.

Procedure

Enable VFR.

- Enable VFR for IPv4 packets
- Enable VFR for IPv6 packets

To enable VFR for IPv4 packets on Inbound Interface Traffic perform the following steps:

a) Configure an interface type and enter interface configuration mode.

interface *interface-type interface-number*

b) Enable VFR on the interface and specify the maximum threshold values.

ip virtual-reassembly [**max-reassemblies** *number*] [**max-fragments** *number*] [**timeout** *seconds*] [**mode** *modes*][**drop-fragments**]

Example:

Here is the complete configuration example to enable VFR for IPv4 packets:

```
interface GigabitEthernet5
ip virtual-reassembly max-reassemblies 64 max-fragments 16 mode default timeout 5
```

To enable VFR for IPv4 packets on outbound interface traffic perform the following steps:

a) Configure an interface type and enter interface configuration mode.

interface *interface-type interface-number*

b) Enable VFR for outbound interface traffic on the interface and specify the maximum threshold values.

ip virtual-reassembly-out [**max-reassemblies** *number*] [**max-fragments** *number*] [**timeout** *seconds*] [**mode** *modes*][**drop-fragments**]

Example:

Here is the complete configuration example to enable VFR for IPv4 packets:

```
interface GigabitEthernet 5
ip virtual-reassembly-out mode default max-fragments 64
```

To enable VFR for IPv6 packets on inbound interface traffic perform the following steps:

a) Configure an interface type and enter interface configuration mode.

interface *interface-type interface-number*

b) Enable VFR for IPv6 packets on inbound interface traffic.

ipv6 virtual-reassembly [**in**|**out**] [**max-reassemblies** *number*] [**max-fragments** *number*] [**timeout** *seconds*] [**mode** *modes*][**drop-fragments**]

Example:

Here is the complete configuration example to enable VFR for IPv6 packets:

```
interface GigabitEthernet 5
ipv6 virtual-reassembly in mode default max-fragments 25
max-reassemblies 1024
```

To enable VFR for IPv6 packets on outbound interface traffic perform the following steps:

a) Configure an interface type and enter interface configuration mode.

interface *interface-type interface-number*

b) Enable VFR for IPv6 packets on outbound interface traffic.

ipv6 virtual-reassembly [**in**|**out**] [**max-reassemblies** *number*] [**max-fragments** *number*] [**timeout** *seconds*] [**mode** *modes*][**drop-fragments**]

Example:

Here is the complete configuration example to enable VFR for IPv6 packets:

```
interface GigabitEthernet 5
ipv6 virtual-reassembly out mode default max-fragments 25
```

Configure underlay fragmentation using CLI commands

You can configure underlay fragmentation using CLI templates. For more information about using CLI templates, see *CLI Add-On Feature Templates* and *CLI Templates*.

Note By default, CLI templates execute commands in global config mode.

This section provides example CLI configurations to configure underlay fragmentation.

Procedure

Step 1 Enter the config-sdwan mode

Example:

```
sdwan
```

Step 2 Configure an interface type and enter interface configuration mode.

Example:

```
interface interface-name interface-number
```

Step 3 Configure the tunnel interface.

Example:

```
tunnel-interface
```

Step 4 Skip Layer 3 fragmentation and clear overlay DF bit.

Example:

```
inner-fragmentation-disable
```

Step 5 Perform the encapsulation for the GRE interface of the TLOC.

Example:

```
encapsulation gre
```

Only GRE encapsulation is supported for underlay fragmentation in Cisco IOS XE Catalyst SD-WAN Release 17.12.1a.

Here is the complete configuration example to enable underlay fragmentation:

```
sdwan
interface GigabitEthernet1
tunnel-interface
inner-fragmentation-disable
encapsulation gre
```

Enable boost mode using CLI commands

You can enable boost mode using a CLI template. For more information about using CLI templates, see *CLI Add-On Feature Templates* and *CLI Templates*.

Note By default, CLI templates execute commands in global config mode.

Procedure

Enable the boost mode.

Example:

```
platform ipreass boost-mode
```

Here is the complete configuration example to enable the boost mode:

```
platform ipreass boost-mode
```

Verify VFR and underlay fragments

Boost mode

The following is a sample output of the **show platform hardware qfp active infrastructure cvla client handles** command:

```
Device# show platform hardware qfp active infrastructure cvla client handles
Handles for cpp 0:

---------------------

Entity name: IPREASS_CVLA_0

Handle: 0xeea45000

Number of allocations: 0

Memory allocated: 0


Entity name: FNF_AOR

Handle: 0xeea0d000

Number of allocations: 0

Memory allocated: 0


Entity name: NBAR_CVLA_ENTITY

Handle: 0xee946000

Number of allocations: 0

Memory allocated: 0


Entity name: FNF Chunk 2

Handle: 0xef929000

Number of allocations: 0

Memory allocated: 0


Entity name: FNF Chunk 1

Handle: 0xef928000

Number of allocations: 0

Memory allocated: 0
```

```
--------------------
```

The boost mode is disabled if entity for **IPREASS_CVLA_*** is not displayed. Once the boost mode is disabled, the **IPREASS_CVLA_*** disappears after 64 seconds.

VFR for IPv4 packets

The following is a sample output from the **show ip virtual-reassembly** command:

```
Device# show ip virtual-reassembly GigabitEthernet 5
GigabitEthernet5:

   Virtual Fragment Reassembly (VFR) is ENABLED [out]

   Concurrent reassemblies (max-reassemblies): 16

   Fragments per reassembly (max-fragments): 32

   Reassembly timeout (timeout): 3 seconds

   Drop fragments: OFF


   Current reassembly count:0

   Current fragment count:0

   Total reassembly count:12

   Total reassembly timeout
```

The example shows if VFR for IPv4 is enabled or not. **Virtual Fragment Reassembly (VFR) is ENABLED [out]** signifies that VFR is enabled. The total packets that underwent reassembly are also displayed.

VFR for IPv6 packets

The following is a sample output from the **show ipv6 virtual-reassembly** command:

```
Device# show ipv6 virtual-reassembly GigabitEthernet 5
GigabitEthernet5:

   IPv6 Virtual Fragment Reassembly (IPV6VFR) is ENABLED [out]

   IPv6 configured concurrent reassemblies (max-reassemblies): 64

   IPv6 configured fragments per reassembly (max-fragments): 16

   IPv6 configured reassembly timeout (timeout): 3 seconds

   IPv6 configured drop fragments: OFF


   IPv6 current reassembly count:0
```

```
   IPv6 current fragment count:0

   IPv6 total reassembly count:12

   IPv6 total reassembly timeout count:0
```

The example shows if VFR for IPv6 is enabled or not. **Virtual Fragment Reassembly (VFR) is ENABLED [out]** signifies that VFR is enabled. The total packets that underwent reassembly are also displayed.

Underlay fragmentation

The following is a sample output from the **show ip traffic interface GigabitEthernet 1** command:

```
Device# show ip traffic interface GigabitEthernet 1
GigabitEthernet 1 statistics :

  Rcvd:  11048818 total, 749458331 total_bytes

         0 format errors, 0 hop count exceeded

         0 bad header, 0 no route

         0 bad destination, 0 not a router

         0 no protocol, 0 truncated

         0 forwarded

         0 fragments, 0 total reassembled

         0 reassembly timeouts, 0 reassembly failures

         0 discards, 0 delivers

  Sent:  0 total, 0 total_bytes 0 discards

         0 generated, 0 forwarded

         0 fragmented into, 0 fragments, 0 failed

  Mcast: 0 received, 0 received bytes

         0 sent, 0 sent bytes

  Bcast: 0 received, 1256 sent
```

The example shows the number of packets that were sent and received, including the total number of packets. A change from the previous number of packet tranfer indicates that underlay fragmentation is enabled.

The following is a sample output from **show sdwan ftm tloc-list** command:

```
Device# show sdwan ftm tloc-list

--- LOCAL  TLOC LIST ---



Id: 32775 (binosId=0xf808007f), Tenant Id: 0      LocalTLOC, num-nhops: 0   ,hash: 0, ref:
 1    SLA 0x0:0x0 Inner-fragmentation
```

```
-disable: No


[TOTAL-LOCAL-TLOC:1]



--- REMOTE TLOC LIST ---



Id: 32768 (binosId=0xf808000f), Tenant Id: 0       SLAClass, num-nhops: 0   ,hash: 0, ref:
 1    SLA 0x0:0x0
 num-active-nhops: 0


Id: 32774 (binosId=0xf808006f), Tenant Id: 0       SLAClass, num-nhops: 1   ,hash: 0, ref:
 1    SLA 0x1:0x0
 [nhop1] nhop-Id: 19   , Type: IPsec     , Encap: IPSEC SLA 0x1:0x0hw_record_index:   5
198.100.1.5/12366->198.100.1.6/12346 pr
oto 0x800 hash 0x13 wan-if 3 tloc 32774 R-color mpls local-tloc 32775 L-color mpls BFD UP
tloc-capability 0 SLA 0x1:0x0 weight
1   pref 0

 num-active-nhops: 1


[TOTAL-REMOTE-TLOC:2]


--- PENDING TLOC LIST (is_pending_updates:FALSE)---



[TOTAL-PENDING-TLOC:0]



--- UNMATCHED TLOC LIST (is_pending_updates:FALSE)---



[TOTAL-UNMATCHED-TLOC:0]


--- TENANT LOCAL  TLOC LIST ---
```

The example displays all the local TLOCs in the network.

The following is a sample output from **show platform software sdwanR0 next-hop overlay all** command:

```
Device# show platform software sdwan R0 next-hop overlay all

Show sdwan next-hop oce all :


OCE ID: 0xf800013f, OCE Type: SDWAN_NH_OVERLAY

Overlay: client_handle (nil), ppe addr (nil)

  overlay encap: ipsec

  src-ip: 198.100.1.5, src-port: 12366

  dst-ip: 198.100.1.6, dst-port: 12346

  flags: 0x0, linktype: MCP_LINK_IP, ifhandle: 15, encap type: MCP_ET_NULL

  encap rewrite: 00

  mtu: 1446, fixup: 0x0, fixup_flags_2: 0x0, color: mpls, phy_oce_handle: 31, nh_overlay_h:
 0xf800013f
    Overlay_CFG:

    encap type: ipsec

    src-ip: 198.100.1.5, src-port: 12366

    dst-ip: 198.100.1.6, dst-port: 12346

    local_system_ip: 1.1.1.1

    remote_system_ip: 2.2.2.2

    local_color: 2 [mpls], remote_color: 2 [mpls]

    wan_ifindex: 8 [GigabitEthernet2], tun_ifindex: 15 [Tunnel0]

    tun_adj_id: 0, l2_adj_id: 0x1f, tunnel_qos_dpidx: 0x0

    bfd-ld: 20005, ipsec_flow_id: 603979786, session_id: 5

    Inner-fragmentation-disable: yes
```

The example demonstrates whether the inner fragmentation is disabled or enabled in a particular next-hop overlay.

The following is a sample output from **show platform software sdwan F0 next-hop overlay all** command:

```
Device# show platform software sdwan F0 next-hop overlay all

OCE ID: 0xf800013f, OCE Type: SDWAN_NH_OVERLAY

Overlay: client_handle 0x63d321350ba0, ppe addr db910710

  overlay encap: ipsec

  src-ip: 198.100.1.5, src-port: 12366

  dst-ip: 198.100.1.6, dst-port: 12346
```

```
  flags: 0x0, linktype: MCP_LINK_SDWAN, ifhandle: 15, encap type: MCP_ET_ARPA

  encap rewrite: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

  mtu: 1446, fixup: 0x0, fixup_flags_2: 0x800000, color: mpls, phy_oce_handle: 31,
nh_overlay_h: 0xf800013f
    Overlay_CFG:

    encap type: ipsec

    src-ip: 198.100.1.5, src-port: 12366

    dst-ip: 198.100.1.6, dst-port: 12346

    local_system_ip: 1.1.1.1

    remote_system_ip: 2.2.2.2

    local_color: 2 [mpls], remote_color: 2 [mpls]

    wan_ifindex: 8 [GigabitEthernet2], tun_ifindex: 15 [Tunnel0]

    tun_adj_id: 0, l2_adj_id: 0x1f, tunnel_qos_dpidx: 0x0

    bfd-ld: 20005, ipsec_flow_id: 603979786, session_id: 5

    Inner-fragmentation-disable: yes
```

The example demonstrates whether the inner fragmentation is disabled or enabled in all the available overlays.