

# SD-WAN Tunnel Interface Commands


- access-list, on page 1
- allow-service, on page 2
- auto-bandwidth-detect, on page 4
- bandwidth-downstream, on page 4
- bandwidth-upstream, on page 6
- carrier, on page 6
- color, on page 7
- encapsulation, on page 8
- gre-in-udp, on page 9
- exclude-controller-group-list, on page 10
- hello-interval, on page 11
- hello-tolerance, on page 12
- iperf-server, on page 13
- last-resort-circuit, on page 14
- low-bandwidth-link, on page 15
- max-control-connections, on page 16
- nat-refresh-interval, on page 17
- port-hop, on page 18
- tloc-extension, on page 19
- tunnel-interface, on page 19
- vbond-as-stun-server, on page 20
- vmanage-connection-preference, on page 21


# access-list

To apply an access list to an interface, use the **access-list** command in the SD-WAN physical interface configuration mode. To remove the access list, use the **no** form of the command.

**access-list** *acl-name* { **in** | **out** }

**no access-list** *acl-name* { **in** | **out** }

**Syntax Description**

| | |
|---|---|
| *acl-name* | Name of the access list to apply to the interface. |

| | |
|---|---|
| **in** \| **out** | Direction in which to apply the access list. Applying it in the inbound direction (**in**) affects packets being received on the interface. Applying it in the outbound direction (**out**) affects packets being transmitted on the interface. |

**Command Default**

An access list is not applied to an interface.

**Command Modes**

SD-WAN physical interface configuration mode (config-interface-*interface-name*)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco vManage CLI templates. |

**Usage Guidelines**

**Example**

```
Device(config)# sdwan

Device(config-sdwan)# interface ge0/2.101

Device(config-interface-ge0/2.101)# access-list acl1 in
```

# allow-service

To configure the services that are allowed on a tunnel interface, use the **allow-service** command in tunnel interface configuration mode. To disallow a service on a tunnel interface, use the **no** form of the command.

**allow-service** *service-name*

**no allow-service** *service-name*

## Syntax Description

| | |
|---|---|
| *service-name* | Type of service to allow or disallow on the WAN tunnel connection.<br><br>*service-name* can be **all** or one of more of **bfd**, **bgp**, **dhcp**, **dns**, **https**, **icmp**, **netconf**, **ntp**, **ospf**, **sshd**, and **stun**. By default, DHCP (for DHCPv4 and DHCPv6), DNS, HTTPS, and ICMP are enabled on a tunnel interface.<br><br>You cannot disallow the following services: DHCP, DNS, NTP, and STUN. If you allow the NTP service on the tunnel interface, you must configure the address of an NTP server with the **system ntp** command. The **allow-service stun** command pertains to allowing or disallowing a Cisco IOS XE SD-WAN device to generate requests to a generic STUN server so that the device can determine whether it is behind a NAT and, if so, what kind of NAT it is and what the device's public IP address and public port number are. On a Cisco IOS XE SD-WAN device that is behind a NAT, you can also have tunnel interface to discover its public IP address and port number from the Cisco vBond orchestrator, by configuring the **vbond-as-stun-server** command on the tunnel interface.<br><br>To configure more than one service, include multiple **allow-service** commands. Configuring **allow-service all** overrides any commands that allow or disallow individual services.<br><br>**Note**<br>Starting from Cisco IOS XE Catalyst SD-WAN Release 17.16.1a, configuring **allow-service all** on Cisco Catalyst SD-WAN Controllers is only applicable for the following services **bgp**, **dhcp**, **dns**, **https**, **icmp**, **netconf**, **ntp**, **ospf**, **sshd**, and **stun**. |

## Command Default

By default, DHCP (for DHCPv4 and DHCPv6), DNS, HTTPS, and ICMP are enabled on a tunnel interface.

## Command Modes

tunnel interface configuration mode (config-tunnel-interface)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco vManage CLI templates. |

## Usage Guidelines

### Example

```
Device(config)# sdwan

Device(config-sdwan)# interface GigabitEthernet1

Device(config-interface-GigabitEthernet1)# tunnel-interface
Device(config-tunnel-interface)# no allow-service all
Device(config-tunnel-interface)# no allow-service bgp
Device(config-tunnel-interface)# allow-service dhcp
Device(config-tunnel-interface)# allow-service dns
Device(config-tunnel-interface)# allow-service icmp
Device(config-tunnel-interface)# no allow-service sshd
Device(config-tunnel-interface)# no allow-service netconf
Device(config-tunnel-interface)# no allow-service ospf
Device(config-tunnel-interface)# allow-service https
Device(config-tunnel-interface)# no allow-service netconf
Device(config-tunnel-interface)# no allow-service snmp
```

# auto-bandwidth-detect

Configure a device to automatically detect the bandwidth for WAN interfaces in VPN0 during day 0 onboarding. The device detects the bandwidth by contacting an iPerf3 server to perform a speed test. To remove the configuration, use the **no** form of this command.

**auto-bandwidth-detect**
**no auto-bandwidth-detect**

**Syntax Description** This command has no arguments or keywords.

**Command Default** None

**Command Modes** SD-WAN physical interface configuration mode (config-interface-*interface-name*)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.5.1a | This command was introduced. |

**Usage Guidelines** Use the auto-bandwidth-detect to configure a device to automatically detect the bandwidth for the VPN interface when the device boots up and connects to Cisco SD-WAN Manager after completing the PnP process. By default, the device uses a public iPerf3 server to perform a speed test for bandwidth detection. You can specify a private iPerf3 server to use instead by using the iperf-server command.

The private iPerf3 server should run on port 5201, which is the default iPerf3 port.

### Example

The following example shows how to enable automatic bandwidth detection:

```
Device(config)# sdwan
Device(config-sdwan)# interface GigabitEthernet
Device(config-interface-GigabitEthernet) auto-bandwidth-detect
```

***Table 1: Related Commands***

| Command | Description |
|---|---|
| iperf-server | Specifies a local iPerf3 server that a device contacts to perform a speed test for automatic bandwidth detection. |

# bandwidth-downstream

To generate notifications when the bandwidth of traffic received on a physical interface in the WAN transport VPN (VPN 0) exceeds a specific limit, use the **bandwidth-downstream** command in the SD-WAN physical interface configuration mode.

Notifications are generated when traffic exceeds 85 percent of the bandwidth you configure with this command. To stop notification generation, use the **no** form of the command.

**bandwidth-downstream** *kbps*

**no bandwidth-downstream**

## Syntax Description

| | |
|---|---|
| *kbps* | Maximum received on a physical interface to allow before generating a notification. When the transmission rates exceeds 85 percent of this rate, an SNMP trap is generated.<br><br>Range: 1 through 2147483647 kbps |

## Command Default

By default, bandwidth notifications are not generated.

## Command Modes

SD-WAN physical interface configuration mode (config-interface-*interface-name*)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco vManage CLI templates. |
| Cisco IOS XE Catalyst SD-WAN Release 17.16.1a | From Cisco IOS XE Catalyst SD-WAN Release 17.2.1v through 17.15.x, the **bandwidth-downstream** command works only if you have also configured per-tunnel QoS for the device.<br><br>From Cisco IOS XE Catalyst SD-WAN Release 17.16.1a the **bandwidth-downstream** command works even without configuring per-tunnel QoS. |

## Usage Guidelines

Notifications generated include Netconf notifications, which are sent to Cisco SD-WAN Manager, SNMP traps, and syslog messages. Notifications are sent when the received bandwidth exceeds 85 percent of the bandwidth configured for that type of traffic.

A second use of **bandwidth-downstream** command is for enabling per-tunnel QoS, together with the **tunnel-qos spoke** command. For more information about enabling QoS, see Per-Tunnel QoS.

### Example

```
Device(config)# sdwan

Device(config-sdwan)# interface GigabitEthernet1

Device(config-interface-GigabitEthernet1)# bandwidth-downstream 300000000
```

## Related Commands

| Command | Description |
|---|---|
| **bandwidth-upstream** | Generates notifications when the bandwidth of traffic transmitted on a physical interface exceeds 85% of the configured bandwidth reference value. |

# bandwidth-upstream

To generate notifications when the bandwidth of traffic transmitted on a physical interface exceeds a specific limit, use the **bandwidth-upstream** command.

Notifications are generated when traffic exceeds 85 percent of the bandwidth you configure with this command. To stop notification generation, use the **no** form of the command.

**bandwidth-upstream** *kbps*

**no bandwidth-upstream**

## Syntax Description

| | |
|---|---|
| *kbps* | Maximum kbps transmitted on a physical interface to allow before generating a notification. When the transmission rates exceeds 85 percent of this rate, an SNMP trap is generated.<br>Range: 1 through 2147483647 kbps |

## Command Default

By default, bandwidth notifications are not generated.

## Command Modes

Interface configuration mode (config-interface-*interface-name*)

## Command History

| Release | Modification |
|---|---|
| Cisco Catalyst SD-WAN Manager Release 20.16.1 | This command was introduced. |

### Example

```
Device(config)# sdwan
Device(config-sdwan)# interface GigabitEthernet1
Device(config-interface-GigabitEthernet1)# bandwidth-upstream 300000000
```

## Related Commands

| Command | Description |
|---|---|
| **bandwidth-downstream** | Generates notifications when the bandwidth of traffic received on a physical interface exceeds 85% of the configured bandwidth reference value. |

# carrier

To associate a carrier name or private network identifier with a tunnel interface, use the **carrier**

command in tunnel interface configuration mode. To remove the association, use the **no** form of the command.

**carrier** *carrier-name*

**no carrier**

**Syntax Description**

| | |
|---|---|
| *carrier-name* | Carrier name to associate with a tunnel interface.<br>Values:<br>**carrier1**, **carrier2**, **carrier3**, **carrier4**, **carrier5**, **carrier6**, **carrier7**, **carrier8**, **default**<br>Default: **default** |

**Command Default**

The carrier name 'default' is associated with a tunnel interface.

**Command Modes**

tunnel interface configuration mode (config-tunnel-interface)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco vManage CLI templates. |

**Usage Guidelines**

**Example**

```
Device(config)# sdwan

Device(config-sdwan)# interface GigabitEthernet1

Device(config-interface-GigabitEthernet1)# tunnel-interface
Device(config-tunnel-interface)# carrier default
```

# color

To assign a color to a WAN transport tunnel, use the **color** command in tunnel interface configuration mode. To remove the color assignment and revert to the default configuration, use the **no** form of the command.

**color** *color*

**no color**

**Syntax Description**

| | |
|---|---|
| *color* | Identify an individual WAN transport tunnel by assigning it a color. The color is one of the TLOC parameters associated with the tunnel. On a Cisco IOS XE SD-WAN device, you can configure only one tunnel interface that has the color **default**. The colors **metro-ethernet**, **mpls**, and **private1** through **private6** are private colors. They use private addresses to connect to the remote side Cisco IOS XE SD-WAN device in a private network. You can use these colors in a public network provided that there is no NAT device between the local and remote vEdge routers.<br>Values:<br>**3g**, **biz-internet**, **blue**, **bronze**, **custom1**, **custom2**, **custom3**, **default**, **gold**, **green**, **lte**, **metro-ethernet**, **mpls**, **private1**, **private2**, **private3**, **private4**, **private5**, **private6**, **public-internet**, **red**, and **silver** |

**Command Default**

The transport tunnel is assigned the color **default**

**Command Modes** tunnel interface configuration mode (config-tunnel-interface)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco vManage CLI templates. |

**Usage Guidelines**

**Example**

```
Device(config)# sdwan

Device(config-sdwan)# interface GigabitEthernet1

Device(config-interface-GigabitEthernet1)# tunnel-interface
Device(config-tunnel-interface)# color lte
```

# encapsulation

To configure the encapsulation for a tunnel interface, use the **encapsulation** command in the tunnel interface configuration mode. To disable the encapsulation configuration, use the **no** form of the command.

**encapsulation** { **gre** | **ipsec** } [ **weight** *number* ]

**no encapsulation** { **gre** | **ipsec** } [ **weight** ]

**Syntax Description**

| | |
|---|---|
| { **gre** \| **ipsec** } | Configure the encapsulation to use on the tunnel interface. This encapsulation is one of the TLOC properties associated with the tunnel, along with the IP address and the color. The default IP MTU for GRE is 1468 bytes, and for IPsec it is 1442 bytes because of the larger overhead. |
| **weight** *number* | Weight to use to balance traffic across multiple tunnels (that is, across multiple TLOCs). A higher value sends more traffic to the tunnel. You typically set the weight based on the bandwidth of the TLOC. When a Cisco IOS XE SD-WAN device has multiple TLOCs, all with the highest preference, traffic distribution is weighted according to the configured weight value. For example, if TLOC A has weight 10, and TLOC B has weight 1, and both TLOCs have the same preference value, then roughly 10 flows are sent out TLOC A for every 1 flow sent out TLOC B.<br>Range: 1 through 255<br>Default: 1 |

**Command Default** Encapsulation is not configured for a tunnel interface.

**Command Modes** Tunnel interface configuration mode (config-tunnel-interface)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco vManage CLI templates. |

**Usage Guidelines**

For a single tunnel, you can configure both IPsec and GRE encapsulations, by including two **encapsulation** commands. Cisco SD-WAN then creates two TLOCs for the tunnel interface. Both TLOCs have the same IP address and color, but one has IPsec encapsulation while the other has GRE encapsulation.

### GRE encapsulation

```
Device(config)# sdwan

Device(config-sdwan)# interface GigabitEthernet1

Device(config-interface-GigabitEthernet1)# tunnel-interface
Device(config-tunnel-interface)# encapsulation gre weight 1
```

### IPsec encapsulation

```
Device(config)# sdwan

Device(config-sdwan)# interface GigabitEthernet1

Device(config-interface-GigabitEthernet1)# tunnel-interface
Device(config-tunnel-interface)# encapsulation ipsec weight 1
```

# gre-in-udp

To enable GRE-in-UDP packet encapsulation, use the **gre-in-udp** command in tunnel interface configuration mode. To disable GRE-in-UDP packet encapsulation, use the **no** form of the command.

**gre-in-udp**

**no gre-in-udp**

**Command Default**

**gre-in-udp** is not enabled.

**Command Modes**

tunnel interface configuration mode (config-tunnel-interface)

***Table 2: Command History***

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.11.1a | This command was introduced. |

**Usage Guidelines**

Use the command **encapsulation gre** to enable GRE packet encapsulation. Then enable GRE-in-UDP packet encapsulation mode.

### Example

The following example shows how to enable GRE-in-UDP.

```
device(config)# sdwan

device(config-sdwan)# interface GigabitEthernet1

device(config-interface-GigabitEthernet1)# tunnel-interface
device(config-tunnel-interface)# encapsulation gre
device(config-tunnel-interface)# gre-in-udp
```

# exclude-controller-group-list

To configure Cisco vSmart Controllers with which a tunnel interface is not allowed to establish control connections, use the **exclude-controller-group-list** command in tunnel interface configuration mode. To remove the configuration, use the **no** form of the command.

**exclude-controller-group-list** *number*

**no exclude-controller-group-list** *number*

## Syntax Description

| | |
|---|---|
| *number* | Identifiers of one or more Cisco vSmart controller groups that this tunnel is not allowed to establish control connections with. Separate multiple numbers with a space.<br><br>Range: 0 through 100 |

## Command Default

No Cisco vSmart controller group is excluded.

## Command Modes

tunnel interface configuration mode (config-tunnel-interface)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco vManage CLI templates. |

## Usage Guidelines

On a system-wide basis, you configure all the Cisco vSmart controllers that the router can connect to using the **system controller-group-list** command. Use the **exclude-controller-group-list** command to restrict the Cisco vSmart controllers to which a particular tunnel interface can establish connections.

### Example

```
Device(config)# sdwan

Device(config-sdwan)# interface GigabitEthernet1

Device(config-interface-GigabitEthernet1)# tunnel-interface
Device(config-tunnel-interface)# exclude-controller-group-list 1
```

# hello-interval

To configure the keepalive interval between Hello packets sent on a DTLS or TLS WAN transport connection, use the **hello-interval** command in tunnel interface configuration mode. To revert to the default configuration, use the **no** form of the command.

**hello-interval** *milliseconds*

**no hello-interval**

## Syntax Description

*milliseconds* Interval between Hello packets sent on a DTLS or TLS WAN tunnel connection. The combination of the hello interval and hello tolerance determines how long to wait before declaring a DTLS or TLS tunnel to be down.

The hello tolerance interval must be at least two times the tunnel hello interval. The default hello interval is 1000 milliseconds (1 second).

**Note**
The hello interval is configured in milliseconds, and the hello tolerance is configured in seconds.

With the default hello interval of 1 second and the default tolerance of 12 seconds, if no Hello packet is received within 11 seconds, the tunnel is declared down at 12 seconds. If the hello interval or the hello tolerance, or both, are different at the two ends of a DTLS or TLS tunnel, the tunnel chooses the interval and tolerance as follows:

- For a tunnel connection between a Cisco IOS XE SD-WAN device and any controller device, the tunnel uses the hello interval and tolerance times configured on the Cisco IOS XE SD-WAN device. This choice is made to minimize the amount traffic sent over the tunnel, to allow for situations where the cost of a link is a function of the amount of traffic traversing the link. The hello interval and tolerance times are chosen separately for each tunnel between a Cisco IOS XE SD-WAN device and a controller device.

Range: 100 through 600000 milliseconds (10 minutes)

Default: 1000 milliseconds (1 second)

**Note**
If the tunnel interface is configured as a low-bandwidth link, the control connection might flap if you use a hello-interval of 100 milliseconds. For low-bandwidth link interfaces, use hello-interval of more than 100 milliseconds.

## Command Default

The default hello interval is 1000 milliseconds.

## Command Modes

tunnel interface configuration mode (config-tunnel-interface)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco vManage CLI templates. |

## Usage Guidelines

### Example

```
Device(config)# sdwan

Device(config-sdwan)# interface GigabitEthernet1

Device(config-interface-GigabitEthernet1)# tunnel-interface
Device(config-tunnel-interface)# hello-interval 1000
```

# hello-tolerance

To configure how long to wait for a Hello packet on a DTLS or TLS WAN transport connection before declaring that transport tunnel to be down, use the **hello-tolerance** command in tunnel interface configuration mode. To revert to the default configuration, use the **no** form of the command.

**hello-tolerance** *seconds*

**no hello-tolerance**

## Syntax Description

| | |
|---|---|
| *seconds* | How long to wait since the last Hello packet was sent on a DTLS or TLS WAN tunnel connection before declaring the tunnel to be down. The hello tolerance interval must be at least twice the hello interval, to ensure that at least one keepalive packet reaches and then returns from the remote side before timing out the peer. The default hello interval is 1000 milliseconds (1 second).<br>**Note**<br>The hello interval is configured in milliseconds, and the hello tolerance is configured in seconds.<br><br>The combination of the hello interval and hello tolerance determines how long to wait before declaring a DTLS or TLS tunnel to be down. With the default hello interval of 1 second and the default tolerance of 12 seconds, if no Hello packet is received within 11 seconds, the tunnel is declared down at 12 seconds. If the hello interval or the hello tolerance, or both, are different at the two ends of a DTLS or TLS tunnel, the tunnel chooses the interval and tolerance as follows:<br>• For a tunnel connection between a Cisco IOS XE SD-WAN device and any controller device, the tunnel uses the hello interval and tolerance times configured on the Cisco IOS XE SD-WAN device. This choice is made to minimize the amount traffic sent over the tunnel, to allow for situations where the cost of a link is a function of the amount of traffic traversing the link. The hello interval and tolerance times are chosen separately for each tunnel between a Cisco IOS XE SD-WAN device and a controller device.<br><br>Range: 12 through 6000 seconds (10 minutes)<br><br>Default: 12 seconds |

## Command Default

The default hello tolerance is 12 seconds.

**Command Modes** tunnel interface configuration mode (config-tunnel-interface)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco vManage CLI templates. |

**Usage Guidelines**

**Example**

```
Device(config)# sdwan

Device(config-sdwan)# interface GigabitEthernet1

Device(config-interface-GigabitEthernet1)# tunnel-interface
Device(config-tunnel-interface)# hello-tolerance 12
```

# iperf-server

Specify a private iPerf3 server that a device contacts to perform a speed test for automatic bandwidth detection. To remove the private iPerf3 server specification, use the **no** form of this command.

**iperf-server** *ipv4-address*
**no iperf-server**

**Syntax Description**

| | |
|---|---|
| *ipv4-address* | IPv4 address of a private iPerf3 server used for automatic bandwidth detection. |

**Command Default** None

**Command Modes** SD-WAN physical interface configuration mode (config-interface-*interface-name*)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.5.1a | This command was introduced. |

**Usage Guidelines** When you use the auto-bandwidth-detect command to configure a device to perform automatic bandwidth detection, the device contacts an iPerf3 server to perform a speed test to determine the bandwidth. By default, the device contacts a public iPerf3 server for this speed test. Use the iperf-server to designate a private iPerf3 server that a device contacts instead.

We recommend that you use a private iPerf3 server. If a private iPerf3 server is not specified, the device pings a system defined set of public iPerf3 servers and selects for the speed test the public server with the minimum hops value. If all servers have the same minimum hops value, the device selects the server with the minimum and latency value. If the speed test fails, the device selects another public server from the list. The device continues to select other public iPerf3 servers until the speed test is successful or until it has tried all servers. Therefore, a speed test on a public iPerf3 server can use a server that is far away and there can be a larger latency than the minimum.

### Example

The following example shows how to specify a private iPerf3 server for automatic bandwidth detection:

```
Device(config)# sdwan
Device(config-sdwan)# interface GigabitEthernet1
Device(config-interface-GigabitEthernet1) auto-bandwidth-detect
Device(config-interface-GigabitEthernet1) iperf-server 10.1.1.1
```

***Table 3: Related Commands***

| Command | Description |
|---|---|
| auto-bandwidth-detect | Configure a device to automatically determine the bandwidth for WAN interfaces in VPN0 during day 0 onboarding by performing a speed test using an iPerf3 server. |

# last-resort-circuit

To configure a tunnel interface as the circuit of last resort, use the **last-resort-circuit** command in tunnel interface configuration mode. To remove the configuration as the circuit of last resort, use the **no** form of the command.

**last-resort-circuit**

**no last-resort-circuit**

## Syntax Description

This command has no arguments or keywords.

## Command Default

By default, this feature is disabled, and the tunnel interface is not considered to be the circuit of last resort.

## Command Modes

tunnel interface configuration mode (config-tunnel-interface)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco vManage CLI templates. |

## Usage Guidelines

There is a delay of 7 seconds before switching back to the primary tunnel interface from a circuit of last resort. This delay is to ensure that the primary interface is once again fully operational and is not still flapping.

When you configure a tunnel interface to be a last-resort circuit, the cellular modem becomes dormant and no traffic is sent over the circuit. However, the cellular modem is kept in online mode so that the modem radio can be monitored at all times and to allow for faster switchover in the case the tunnel interface needs to be used as the last resort.

To minimize the amount of extraneous data plane traffic on a cellular interface that is a circuit of last resort, increase the BFD Hello packet interval and disable PMTU discover.

### Example

```
Device(config)# sdwan

Device(config-sdwan)# interface GigabitEthernet1

Device(config-interface-GigabitEthernet1)# tunnel-interface
Device(config-tunnel-interface)# last-resort-circuit
```

# low-bandwidth-link

To configure a tunnel interface as a low bandwidth link, use the **low-bandwidth-link** command in tunnel interface configuration mode. To remove the low bandwidth link configuration, use the **no** form of the command.

**low-bandwidth-link**

**no low-bandwidth-link**

## Syntax Description

This command has no arguments or keywords.

## Command Default

This option is disabled by default.

**Note** Starting from Cisco IOS XE Catalyst SD-WAN Release 17.6.x, LTE enabled CPE is disabled by default.

## Command Modes

tunnel interface configuration mode (config-tunnel-interface)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco SD-WAN Manager CLI templates. |

## Usage Guidelines

This configuration command is relevant only for a spoke router in a hub-and-spoke deployment scenario, where the spoke has a low-bandwidth link, such as an LTE link.

The low bandwidth synchronizes all the BFD sessions and control session hello-interval on LTE WAN circuits to timeout at the same time. The periodic heartbeat messages are sent out at the same time to make optimal usage of LTE circuits radio waves or radio frequency energy to transmit and receive packets. The low bandwidth feature cannot reduce the number of hello packets to be transmitted (Tx) or received (Rx) for the sessions, but synchronizes the hello interval timeout for the sessions.

For example, if the BFD session and control connection hello-interval is 1 sec, and there is no user data traffic active on LTE circuits, then the sessions hello packets transmitted is spread across 1 sec window interval. Each session will timeout anywhere within that 1 sec interval and transmits the hello packet. This makes the LTE radio to be active almost all the time. With low bandwidth feature, all the session hello packets transmits at the same time, and leave the rest of the 1sec interval idle, makes optimal usage of LTE modem radio energy.

**Note** To prevent control-connection flapping when an interface is configured as a low-bandwidth link, use a hello-interval of greater than 100 milliseconds.

### Example

```
Device(config)# sdwan

Device(config-sdwan)# interface GigabitEthernet1

Device(config-interface-GigabitEthernet1)# tunnel-interface
Device(config-tunnel-interface)# low-bandwidth-link
```

# max-control-connections

To configure the maximum number of Cisco Catalyst SD-WAN Controllers that a Cisco IOS XE Catalyst SD-WAN device is allowed to connect to, use the **max-control-connections** command in tunnel interface configuration mode. To remove the configuration, use the **no** form of the command.

**Note** For control connection traffic without dropping any data, a minimum of 650-700 kbps bandwidth is recommended with default parameters configured for hello-interval (10) and hello-tolerance (12).

**max-control-connections** *number*

**no max-control-connections**

## Syntax Description

| | |
|---|---|
| *number* | Sets the maximum number of Cisco Catalyst SD-WAN Controllers that the vEdge router can connect to. These connections are DTLS or TLS control plane tunnels.<br>Range: 0 through 100<br>Default: |

## Command Default

By default, the maximum number of controller connections is set to the same value as the maximum number of OMP sessions configured using the **system max-omp-sessions** command.

## Command Modes

tunnel interface configuration mode (config-tunnel-interface)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco SD-WAN Manager CLI templates. |

## Usage Guidelines

When **max-control-connections** is configured without affinity, devices establish control connection with Cisco Catalyst SD-WAN Controllers having higher System-IP.

### Example

```
Device(config)# sdwan

Device(config-sdwan)# interface GigabitEthernet1

Device(config-interface-GigabitEthernet1)# tunnel-interface
Device(config-tunnel-interface)# max-control-connections 1
```

# nat-refresh-interval

To configure the interval between NAT refresh packets sent on a DTLS or TLS WAN transport connection, use the **nat-refresh-interval** command in tunnel interface configuration mode. This interval is how often a tunnel interface sends a refresh packet to maintain the UDP packet streams that traverse a NAT. To revert to the default configuration, use the **no** form of the command.

**nat-refresh-interval** *seconds*

**no nat-refresh-interval**

## Syntax Description

| | |
|---|---|
| *seconds* | Interval between NAT refresh packets sent on a DTLS or TLS WAN tunnel connection. These packets are sent to maintain the UDP packet streams that traverse a NAT between the device and the Internet or other public network. You might want to increase the interval on interfaces where you are charged for bandwidth, such as LTE interfaces.<br>Range: 1 through 60 seconds<br>Default: 5 seconds |

## Command Default

A tunnel interface has a default NAT refresh interval of 5 seconds.

## Command Modes

tunnel interface configuration mode (config-tunnel-interface)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco vManage CLI templates. |

## Usage Guidelines

### Example

```
Device(config)# sdwan

Device(config-sdwan)# interface GigabitEthernet1

Device(config-interface-GigabitEthernet1)# tunnel-interface
Device(config-tunnel-interface)# nat-refresh-interval 5
```

# port-hop

On a Cisco IOS XE SD-WAN device behind a NAT device, to configure a tunnel interface to rotate through a pool of preselected OMP port numbers, known as base ports, to establish DTLS connections with other WAN edge devices when a connection attempt is unsuccessful, use the **port-hop** command in tunnel interface configuration mode. To disable port hopping for a tunnel interface, use the **no** form of the command.

**port-hop**

**no port-hop**

## Syntax Description

This command has no arguments or keywords.

## Command Default

Port hopping is enabled on a tunnel interface.

## Command Modes

tunnel interface configuration mode (config-tunnel-interface)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco vManage CLI templates. |

## Usage Guidelines

For a tunnel interface (TLOC) on a Cisco IOS XE SD-WAN device behind a NAT device, you can configure the interface to rotate through a pool of preselected OMP port numbers, known as base ports, to establish DTLS connections with other WAN edge devices when a connection attempt is unsuccessful. By default, port hopping is enabled on Cisco IOS XE SD-WAN devices and on all tunnel interfaces on Cisco IOS XE SD-WAN devices.

There are five base ports: 12346, 12366, 12386, 12406, and 12426. These port numbers determine the ports used for connection attempts. The first connection attempt is made on port 12346. If the first connection does not succeed after about 1 minute, port 12366 is tried. After about 2 minutes, port 12386 is tried; after about 5 minutes, port 12406; after about 6 minutes, port 12426 is tried. Then the cycle returns to port 12346.

If you have configured a port offset with the **port-offset** command, the five base ports are a function of the configured offset. For example, with a port offset of 2, the five base ports are 12348, 12368, 12388, 12408, and 12428. Cycling through these base ports happens in the same way as if you had not configured an offset.

### Example

```
Device(config)# sdwan

Device(config-sdwan)# interface GigabitEthernet1

Device(config-interface-GigabitEthernet1)# tunnel-interface
Device(config-tunnel-interface)# port-hop
```

# tloc-extension

To bind an interface, which connects to another WAN edge device at the same physical site, to the local device's WAN transport interface, use the **tloc-extension** command in the SD-WAN physical interface configuration mode. Note that you can configure the two devices themselves with different site identifiers. To remove the binding, use the **no** form of the command.

**tloc-extension** *interface-name*

**no tloc-extension**

**Syntax Description**

| | |
|---|---|
| *interface-name* | Physical interface on the local router that connects to the WAN transport circuit. The interface can be a Gigabit Ethernet interface (**ge**) or a PPP interface (ppp). |

**Command Default**

**Command Modes**

SD-WAN physical interface configuration mode (config-interface-*interface-name*)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco vManage CLI templates. |

**Usage Guidelines**

**Example**

```
Device(config)# sdwan

Device(config-sdwan)# interface ge0/2.101

Device(config-interface-ge0/2.101)# tloc-extension ge0/0
```

# tunnel-interface

To configure an interface as a secure DTLS or TLS WAN transport connection, use the **tunnel-interface** command in the GigabitEthernet interface configuration mode. To disable the tunnel interface configuration, use the **no** form of the command.

**tunnel-interface**

**no tunnel-interface**

**Syntax Description**

This command has no arguments or keywords.

**Command Default**

A GigabitEthernet interface is not configured as a transport connection.

**Command Modes**

GigabitEthernet interface configuration mode (config-interface-GigabitEthernet)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco vManage CLI templates. |

**Usage Guidelines**

Configuring an interface to be a transport tunnel enables the flow of control and data traffic on the interface. On a Cisco IOS XE SD-WAN device, you must also configure the interface's TLOC attributes, which are carried in the TLOC OMP routes that the device sends to the Cisco vSmart controllers in its domain. For the TLOC attributes on the device, you must configure, at a minimum, a color and an encapsulation type. These two attributes, along with the router's system IP address, are the 3-tuple that uniquely identify each TLOC.

**Example**

```
Device(config)# sdwan

Device(config-sdwan)# interface GigabitEthernet1

Device(config-interface-GigabitEthernet1)# tunnel-interface
```

# vbond-as-stun-server

To enable Session Traversal Utilities for NAT (STUN) and allow the tunnel interface to discover its public IP address and port number when the Cisco IOS XE SD-WAN device is located behind a NAT, use the **vbond-as-stun-server** command in tunnel interface configuration mode. When you configure this command, Cisco IOS XE SD-WAN devices can exchange their public IP addresses and port numbers over private TLOCs. To disable STUN, use the **no** form of the command.

**vbond-as-stun-server**

**no vbond-as-stun-server**

**Syntax Description**

This command has no arguments or keywords.

**Command Default**

STUN is not enabled by default.

**Command Modes**

tunnel interface configuration mode (config-tunnel-interface)

**Command History**

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco vManage CLI templates. |

**Usage Guidelines**

With this configuration, the Cisco IOS XE SD-WAN device uses the Cisco vBond orchestrator as a STUN server, so that the device can determine its public IP address and public port number. The device cannot learn the type of NAT that it is behind. No overlay network control traffic is sent and no keys are exchanged over tunnel interface configured to use the Cisco vBond orchestrator as a STUN server. However, BFD does come up on the tunnel, and data traffic can be sent on it.

### Example

```
Device(config)# sdwan

Device(config-sdwan)# interface GigabitEthernet1

Device(config-interface-GigabitEthernet1)# tunnel-interface
Device(config-tunnel-interface)# vbond-as-stun-server
```

# vmanage-connection-preference

To configure the preference for using a tunnel interface to exchange control traffic with the Cisco vManage NMS, use the **vmanage-connection-preference** command in tunnel interface configuration mode. Configuring this option is useful for LTE and other links on which you want to minimize traffic. To remove the configured preference and revert to the default configuration, use the **no** form of the command.

**vmanage-connection-preference** *number*

**no vmanage-connection-preference**

## Syntax Description

| | |
|---|---|
| *number* | Preference for using the tunnel interface to exchange control traffic with the Cisco vManage NMS. The tunnel with the higher value has a greater preference to be used for connections to the Cisco vManage NMS. To have a tunnel interface never connect to the Cisco vManage NMS, set the preference value to 0. At least one tunnel interface on the Cisco IOS XE SD-WAN device must have a non-0 preference value.<br>Range: 0 through 8<br>Default: 5 |

## Command Default

A tunnel interface has a default preference of 5.

## Command Modes

tunnel interface configuration mode (config-tunnel-interface)

## Command History

| Release | Modification |
|---|---|
| Cisco IOS XE Catalyst SD-WAN Release 17.2.1v | Command qualified for use in Cisco vManage CLI templates. |

## Usage Guidelines

### Example

```
Device(config)# sdwan

Device(config-sdwan)# interface GigabitEthernet1

Device(config-interface-GigabitEthernet1)# tunnel-interface
Device(config-tunnel-interface)# vmanage-connection-preference 5
```