Layer 2 Configuration Guide, Cisco Catalyst IE3x00 Rugged, IE3400 Heavy Duty, and ESS3300 Series Switches

Cisco Catalyst IE3200 Rugged Series
Updated August 31, 2025

1 Configuring Layer 2 Protocol Tunneling

Topics:

Information about layer 2 protocol tunneling

Describes the Layer 2 protocol tunneling feature that allows service providers to tunnel Layer 2 control protocols across their networks while maintaining protocol transparency for customer networks.

The following sections provide information about Layer 2 protocol tunneling:

Prerequisites for configure layer 2 protocol tunneling

Lists prerequisites and considerations for configuring Layer 2 protocol tunneling.

This topic lists prerequisites and considerations for configuring Layer 2 protocol tunneling.

To configure Layer 2 point-to-point tunneling to facilitate the automatic creation of EtherChannels, you need to configure both the SP (service-provider) edge switch and the customer device.

Layer 2 protocol tunneling

Describes Layer 2 protocol tunneling functionality and implementation across service-provider networks.

Layer 2 protocol tunneling is a service-provider network feature. It encapsulates Layer 2 protocol packets with a special MAC address and allows their transport across service-provider networks. This feature enables proper operation of STP, CDP, and VTP protocols across remote customer sites. It also delivers Layer 2 protocol data units (PDUs) to customer devices on the outbound side of the service-provider network.

Protocol tunneling benefits

When protocol tunneling is enabled, the edge device on the inbound side of the service-provider network encapsulates Layer 2 protocol packets with a special MAC address and sends them across the service-provider network. Core devices in the network do not process these packets. Instead, they forward the packets as standard packets. Layer 2 protocol data units (PDUs) for CDP, STP, or VTP cross the service-provider network. They reach customer devices on the outbound side of the service-provider network. All customer ports on the same VLANs receive identical packets. The results include:

- Users on each of a customer's sites can properly run STP, and every VLAN can build a correct spanning tree based on parameters from all sites and not just from the local site.
- CDP discovers and shows information about the other Cisco devices connected through the service-provider network.
- VTP provides consistent VLAN configuration throughout the customer network, propagating to all devices through the service provider.

Layer 2 protocol tunneling can be used independently or to enhance IEEE 802.1Q tunneling.

If protocol tunneling is not enabled on IEEE 802.1Q tunneling ports, remote devices at the receiving end of the service-provider network will not receive PDUs and cannot properly run STP, CDP, and VTP.

Enabling protocol tunneling ensures that Layer 2 protocols within each customer's network remain separate from those running within the service-provider network.

When customer devices at different sites send traffic through the service-provider network with IEEE 802.1Q tunneling, they achieve complete knowledge of the customer's VLAN.

If IEEE 802.1Q tunneling is not used, Layer 2 protocol tunneling can still be enabled by connecting to the customer device through access ports and enabling tunneling on the service-provider access port.

Customer network topology scenario

Customer X has four switches in the same VLAN, that are connected through the service-provider network. If the network does not tunnel PDUs, switches on the far ends of the network cannot properly run STP, CDP, and VTP. For example, STP for a VLAN on a switch in Customer X, Site 1, will build a spanning tree on the switches at that site without considering

convergence parameters based on Customer X's switch in Site 2. This could result in the topology shown in the Layer 2 Network Topology without Proper Convergence figure.

Figure 1: Layer 2 protocol tunneling

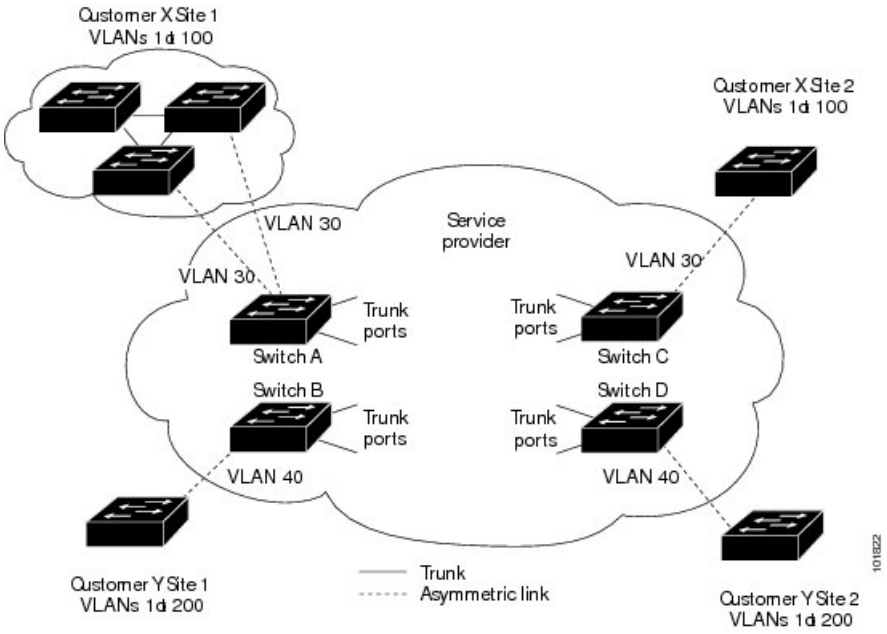
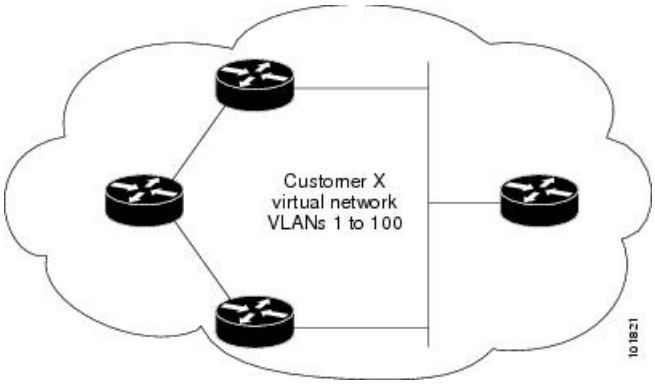


Figure 2: Layer 2 network topology without proper convergence



Layer 2 protocol tunneling on ports

Describes Layer 2 protocol tunneling on ports that connect to customers in service-provider network edge devices.

Layer 2 protocol tunneling is a network service feature that enables tunneling of Layer 2 protocols on ports connected to customers in service-provider network edge devices. It allows service-provider edge devices to perform the tunneling process for customer devices and preserves Layer 2 PDUs intact for delivery across service-provider infrastructure to the other side of the customer network.

Layer 2 protocol tunneling configuration and operation

You can enable Layer 2 protocol tunneling on ports that are configured as access ports or tunnel ports or trunk ports. You cannot enable Layer 2 protocol tunneling on ports configured in either **switchport mode dynamic auto** mode (the default mode) or **switchport mode dynamic desirable** mode.

The device supports Layer 2 protocol tunneling for CDP, STP, VTP, and LLDP. The device does not support Layer 2 protocol tunneling for UDLD.

> **Note**
>
> PAgP and LACP protocol tunneling is intended only to emulate a point-to-point topology. An erroneous configuration that sends tunneled packets to multiple ports could cause a network failure.

Edge device tunnel ports are connected to customer IEEE 802.1Q trunk ports. Edge device access ports are connected to customer access ports. The edge devices connected to the customer device handle the tunneling process.

When Layer 2 PDUs enter the service-provider inbound edge device through a Layer 2 protocol-enabled port and exit through the trunk port into the service-provider network, the device overwrites the customer PDU-destination MAC address with a well-known Cisco proprietary multicast address (01-00-0c-cd-cd-d0). If IEEE 802.1Q tunneling is enabled, packets are also double-tagged; the outer tag is the customer metro tag, and the inner tag is the customer's VLAN tag. The core devices ignore the inner tags and forward the packet to all trunk ports in the same metro VLAN. The edge devices on the outbound side restore the proper Layer 2 protocol and MAC address information and forward the packets to all tunnel or access ports in the same metro VLAN.

You can also enable Layer 2 protocol tunneling on access ports on the edge switch connected to access or trunk ports on the customer switch. In this case, the encapsulation and decapsulation process is the same as described in the previous paragraph, except that the packets are not double-tagged in the service-provider network. The single tag is the customer-specific access VLAN tag.

In switch stacks, Layer 2 protocol tunneling configuration is distributed among all stack members. Each stack member that receives an ingress packet on a local port encapsulates or decapsulates the packet and forwards it to the appropriate destination port. On a single switch, ingress Layer 2 protocol-tunneled traffic is sent across all local ports in the same VLAN on which Layer 2 protocol tunneling is enabled. In a stack, packets received by a Layer 2 protocol-tunneled port are distributed to all ports in the stack that are configured for Layer 2 protocol tunneling in the same VLAN. The stack master handles all Layer 2 protocol tunneling configuration and distributes it to all stack members.

Layer 2 protocol tunneling packet flow

In the Layer 2 Protocol Tunneling figure in Layer 2 Protocol Tunneling Overview, Customer X and Customer Y are in access VLANs 30 and 40, respectively. Asymmetric links connect the customers in Site 1 to edge switches in the service-provider network. The Layer 2 PDUs (for example, BPDUs) coming into Switch B from Customer Y in Site 1 are forwarded to the infrastructure as double-tagged packets with the well-known MAC address as the destination MAC address. These double-tagged packets have the metro VLAN tag of 40, as well as an inner VLAN tag (for example, VLAN 100). When the double-tagged packets enter Switch D, the outer VLAN tag 40 is removed, the well-known MAC address is replaced with the respective Layer 2 protocol MAC address, and the packet is sent to Customer Y on Site 2 as a single-tagged frame in VLAN 100.

Layer 2 protocol tunneling for EtherChannels

Explains how Layer 2 protocol tunneling enhances EtherChannel creation in service provider networks by emulating point-to-point network topology.

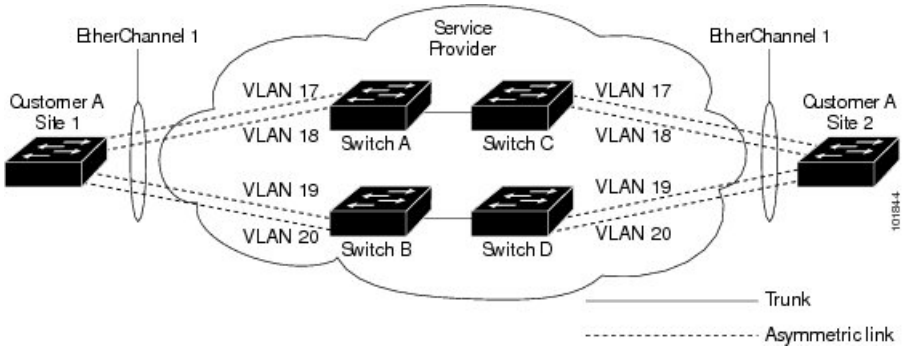
Layer 2 protocol tunneling for EtherChannels is a network feature that enhances the creation of EtherChannels by emulating a point-to-point network topology in an SP network. When protocol tunneling (PAgP or LACP) is enabled on the SP switch, remote customer switches can receive PDUs. This feature also allows switches on the far ends of the network to negotiate the automatic creation of EtherChannels without dedicated lines.

Configuration requirements

While configuring Layer 2 Protocol Tunneling on trunk ports, both the trunk ports on the SP edge device should be configured with different native VLANs. The native VLAN of one trunk port should not be in the list of allowed VLANs of the other trunk port to avoid loops.

This topology shows Layer 2 Protocol Tunneling for EtherChannels.

Figure 3: Layer 2 protocol tunneling for EtherChannels



Customer network scenario

Customer A has two switches in the same VLAN that are connected through the SP network. When the network tunnels PDUs, switches on the far ends of the network can negotiate the automatic creation of EtherChannels without needing dedicated lines.

Default layer 2 protocol tunneling configuration

Lists the default Layer 2 Protocol Tunneling configuration settings.

This table shows the default Layer 2 protocol tunneling configuration.

Table 1: Default layer 2 ethernet interface VLAN configuration

Feature	Default Setting
Layer 2 protocol tunneling	Disabled.
Shutdown threshold	None set.
Drop threshold	None set.
CoS Value	If a CoS value is configured on the interface, that value is used to set the BPDU CoS value for Layer 2 protocol tunneling. If no CoS value is configured at the interface level, the default value for CoS marking of L2 protocol tunneling BPDUs is 5. This does not apply to data traffic.

Configure layer 2 protocol tunneling

Configure Layer 2 protocol tunneling to enable protocol tunneling for desired protocols on IEEE 802.1Q tunnel or trunk ports.

Configure Layer 2 protocol tunneling to allow Layer 2 protocols (CDP, LLDP, point-to-point, STP, VTP) to be tunneled across a service provider network while maintaining protocol transparency between customer sites.

Layer 2 protocol tunneling enables service providers to tunnel Layer 2 protocols across their networks, allowing customer protocols to maintain end-to-end connectivity. This configuration is typically performed on provider edge interfaces connected to customer equipment.

Before you begin

Follow these steps to configure Layer 2 protocol tunneling:

Procedure

1. Use the **configure terminal** command to enter global configuration mode. Then, use the **interface** *interface-id* command to specify the interface to configure and enter interface configuration mode.

   Example

   ```
   Device(config)# interface gigabitethernet1/0/1
   ```

2. Use the **switchport mode dot1q-tunnel** or **switchport mode trunk** command to configure the interface as an IEEE 802.1Q tunnel port or a trunk port.

   Example

   ```
   Device(config-if)# switchport mode dot1q-tunnel
   ```

   or

   ```
   Device(config-if)# switchport mode trunk
   ```

3. Use the **l2protocol-tunnel** [**cdp** | **lldp** | **point-to-point** | **stp** | **vtp**] command to enable protocol tunneling for the desired protocol.

   Example

   ```
   Device(config-if)# l2protocol-tunnel cdp
   ```

   If no keyword is entered, tunneling is enabled for all Layer 2 protocols.

   > **Note**
   >
   > Use the **no l2protocol-tunnel** interface configuration command to disable protocol tunneling for one Layer 2 protocol or for all protocols.

4. (Optional) Use the **l2protocol-tunnel shutdown-threshold** *packet_second_rate_value* [**cdp** | **lldp** | **point-to-point** | **stp** | **vtp**] command to configure the packets-per-second threshold for encapsulation. When the threshold is exceeded, the interface is disabled.

   Example

   ```
   Device(config-if)# l2protocol-tunnel shutdown-threshold 100 cdp
   ```

   If no protocol option is specified, the threshold applies to each tunneled Layer 2 protocol type. The range is 1 to 4096. The default is to have no threshold configured.

> **Note**
>
> If you also set a drop threshold on this interface, the **shutdown-threshold** value must be greater than or equal to the **drop-threshold** value.

> **Note**
>
> Use the **no l2protocol-tunnel shutdown-threshold** and **no l2protocol-tunnel drop-threshold** commands to return the shutdown and drop thresholds to the default settings.

5. (Optional) Use the **l2protocol-tunnel drop-threshold** *packet_second_rate_value* [**cdp** | **lldp** | **point-to-point** | **stp** | **vtp**] command to configure the packets-per-second threshold for encapsulation. If the rate exceeds this threshold, the interface drops packets.

   Example

   ```
   Device(config-if)# l2protocol-tunnel drop-threshold 100 cdp
   ```

   If no protocol option is specified, the threshold applies to each tunneled Layer 2 protocol type. The range is 1 to 4096. The default is to have no threshold configured.

   > **Note**
   >
   > If you also set a shutdown threshold on this interface, the **drop-threshold** value must be less than or equal to the **shutdown-threshold** value.

   > **Note**
   >
   > Use the **no l2protocol-tunnel shutdown-threshold** and **no l2protocol-tunnel drop-threshold** commands to return the shutdown and drop thresholds to the default settings.

6. Use the **exit** command to return to global configuration mode.

   Example

   ```
   Device(config-if)# exit
   ```

7. Use the **spanning-tree bpdufilter enable** command to insert a BPDU filter for spanning tree.

Example

```
Device(config)# spanning-tree bpdufilter enable
```

> **Note**
>
> While configuring Layer 2 protocol tunneling on a trunk port, you must enable a BPDU filter for spanning tree.

8. Use the **end** command to return to privileged EXEC mode.

Example

```
Device(config)# end
```

9. (Optional) Use the **show l2protocol** command to verify the Layer 2 tunnel ports, configured protocols, thresholds, and counters.

Example

```
Device# show l2protocol
```

How to configure layer 2 protocol tunneling for EtherChannels

Describes how to configure Layer 2 protocol tunneling for EtherChannels on both service provider edge devices and customer devices to enable protocol transparency across the network.

For EtherChannels, you need to configure both the SP (service-provider) edge devices and the customer devices for Layer 2 protocol tunneling. The following sections provide configuration information on how to configure the SP edge device and how to configure the customer device:

Configure the SP edge switch

Configure the SP Edge Switch for Layer 2 protocol tunneling.

Configure an SP Edge Switch to enable Layer 2 protocol tunneling. This feature allows service provider networks to transparently tunnel customer Layer 2 protocols across the provider network.

Service providers use Layer 2 protocol tunneling to preserve customer protocol communications across provider infrastructure. This configuration is essential when connecting customer sites through a service provider network while maintaining protocol transparency.

Before you begin

Follow these steps to configure the SP Edge Switch:

Procedure

1. Use the **configure terminal** command to enter global configuration mode. Then, use the **interface** *interface-id* command to specify the interface to configure and enter interface configuration mode.

Example

```
Device(config)# interface gigabitethernet1/0/1
```

2. Use the **switchport trunk native vlan** *vlan-id* command to configure the native VLAN.

   Example

   ```
   Device(config-if)# switchport trunk native vlan 2
   ```

   Note

   While configuring Layer 2 protocol tunneling for EtherChannels on trunk ports, you must configure different native VLANs on both trunk ports on the SP edge device.

3. Use the **switchport trunk allowed vlan** *vlan-id-list* command to specify the list of allowed VLANs.

   Example

   ```
   Device(config-if)# switchport trunk allowed vlan 1,2,4-3003,3005-4094
   ```

   Note

   When configuring Layer 2 protocol tunneling for EtherChannels on trunk ports, make sure that the native VLAN of one trunk port on the SP edge device is not included in the list of allowed VLANs of the other trunk port. This precaution avoids loops.

4. Use the **switchport mode dot1q-tunnel** or **switchport mode trunk** command to configure the interface as an IEEE 802.1Q tunnel port or as a trunk port.

   Example

   ```
   Device(config-if)# switchport mode dot1q-tunnel
   ```

   or

   ```
   Device(config-if)# switchport mode trunk
   ```

5. Configure these optional steps as required.
   a) Use the **l2protocol-tunnel point-to-point** [**pagp** | **lacp** | **udld**] command to enable point-to-point protocol tunneling for the desired protocol.

      Example

      ```
      Device(config-if)# l2protocol-tunnel point-to-point pagp
      ```

      If no keyword is entered, tunneling is enabled for all three protocols.

> **Note**
>
> To avoid a network failure, make sure that the network is a point-to-point topology before you enable tunneling for PAgP, LACP, or UDLD packets.

> **Note**
>
> Use the **no l2protocol-tunnel point-to-point** interface configuration command to disable point-to-point protocol tunneling for one Layer 2 protocol or for all three protocols.

b) Use the **l2protocol-tunnel shutdown-threshold** [**point-to-point** [**pagp** | **lacp** | **udld**]] *value* command to configure the packets-per-second threshold accepted for encapsulation before the interface is disabled.

Example

```
Device(config-if)# l2protocol-tunnel shutdown-threshold point-to-point pagp
 100
```

If you do not specify a protocol option, the threshold applies to each tunneled Layer 2 protocol type. The valid range is 1 to 4096. By default, no threshold is configured.

> **Note**
>
> If you also set a drop threshold on this interface, the **shutdown-threshold** value must be greater than or equal to the **drop-threshold** value.

> **Note**
>
> Use the **no l2protocol-tunnel shutdown-threshold** and **no l2protocol-tunnel drop-threshold** commands to return the shutdown and drop thresholds to the default settings.

c) Use the **l2protocol-tunnel drop-threshold** [**point-to-point** [**pagp** | **lacp** | **udld**]] *value* command to configure the packets-per-second threshold accepted for encapsulation before the interface drops packets.

Example

```
Device(config-if)# l2protocol-tunnel drop-threshold point-to-point pagp
500
```

If you do not specify a protocol option, the threshold applies to each tunneled Layer 2 protocol type. The valid range is 1 to 4096. By default, no threshold is configured.

> **Note**
>
> If you also set a shutdown threshold on this interface, the **drop-threshold** value must be less than or equal to the **shutdown-threshold** value.

6. Use the **no cdp enable** command to disable CDP on the interface.

   Example

   ```
   Device(config-if)# no cdp enable
   ```

7. Use the **spanning-tree bpdufilter enable** command to enable BPDU filtering on the interface.

   Example

   ```
   Device(config-if)# spanning-tree bpdufilter enable
   ```

8. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config-if)# end
   ```

9. (Optional) Use the **show l2protocol** command to verify the Layer 2 tunnel ports, configured protocols, thresholds, and counters.

   Example

   ```
   Device# show l2protocol
   ```

Configure the customer device

Configure the customer switch for Layer 2 protocol tunneling with EtherChannels.

Configure the customer device to enable Layer 2 protocol tunneling for EtherChannels, which allows proper communication between the service provider edge device and the customer equipment.

This configuration is required when implementing EtherChannels in a service provider environment where Layer 2 protocols need to be tunneled across the provider network.

Before you begin

To enable Layer 2 protocol tunneling for EtherChannels, configure both the service provider edge device and the customer device.

Follow these steps to configure the customer device for EtherChannel Layer 2 protocol tunneling:

Procedure

1. Use the **configure terminal** command to enter global configuration mode. Then, use the **interface** *interface-id* command to specify the interface to configure and enter interface configuration mode.

   Example

   ```
   Device(config)# interface gigabitethernet1/0/1
   ```

2. Use the **switchport trunk encapsulation dot1q** command to set the trunking encapsulation format to IEEE 802.1Q.

   Example

   ```
   Device(config-if)# switchport trunk encapsulation dot1q
   ```

3. Use the **switchport mode trunk** command to enable trunking on the interface.

   Example

   ```
   Device(config-if)# switchport mode trunk
   ```

4. Use the **udld port** command to enable UDLD in normal mode on the interface.

   Example

   ```
   Device(config-if)# udld port
   ```

5. Use the **channel-group** *channel-group-number* **mode desirable** command to assign the interface to a channel group and specify desirable PAgP mode.

   Example

   ```
   Device(config-if)# channel-group 25 mode desirable
   ```

6. Use the **exit** command to return to global configuration mode.

   Example

   ```
   Device(config-if)# exit
   ```

   Note

   Use the **no switchport mode trunk**, **no udld port**, and **no channel-group** *channel-group-number* interface configuration commands to return the interface to the default settings.

7. Use the **interface port-channel** *port-channel-number* command to enter port-channel interface mode.

Example

```
Device(config)# interface port-channel 25
```

8. Run these commands on the interface.
   a) Use the **shutdown** command to shut down the interface.
   b) Use the **no shutdown** command to enable the interface.
   c) Use the **end** command to return to privileged EXEC mode.
9. (Optional) Use the **show l2protocol** command to verify the Layer 2 tunnel ports, configured protocols, thresholds, and counters.

Example

```
Device# show l2protocol
```

Configuration examples for layer 2 protocol tunneling

Describes configuration examples for layer 2 protocol tunneling across different network scenarios. Explains how to implement L2PT to tunnel layer 2 protocols like CDP, STP, and VTP across service provider networks.

The following sections provide various configuration examples for layer 2 protocol tunneling:

Configure layer 2 protocol tunneling

Provides an example configuration for Layer 2 protocol tunneling for CDP, STP, and VTP protocols.

This example shows how to configure Layer 2 protocol tunneling for CDP, STP, and VTP and to verify the configuration.

```
Device(config)# interface gigabitethernet1/0/11
Device(config-if)# l2protocol-tunnel cdp
Device(config-if)# l2protocol-tunnel stp
Device(config-if)# l2protocol-tunnel vtp
Device(config-if)# l2protocol-tunnel shutdown-threshold 1500
Device(config-if)# l2protocol-tunnel drop-threshold 1000
Device(config-if)# exit
Device(config)# l2protocol-tunnel cos 7
Device(config)# end
Device# show l2protocol

COS for Encapsulated Packets: 7
Port Protocol Shutdown Drop Encapsulation Decapsulation Drop
Threshold Threshold Counter Counter Counter
------- -------- --------- --------- ------------- ------------- -------------
Gi0/11 cdp 1500 1000 2288 2282 0
stp 1500 1000 116 13 0
vtp 1500 1000 3 67 0
pagp ---- ---- 0 0 0
lacp ---- ---- 0 0 0
udld ---- ---- 0 0 0
```

Configure the SP edge and customer switches

Provides configuration examples for SP edge switches and customer switches with dot1q tunneling, L2 protocol tunneling, and EtherChannel settings.

This example shows how to configure the SP edge switch 1 and edge switch 2. VLANs 17, 18, 19, and 20 are the access VLANs, Fast Ethernet interfaces 1 and 2 are point-to-point tunnel ports with PAgP and UDLD enabled, the drop threshold is 1000, and Fast Ethernet interface 3 is a trunk port.

SP edge switch 1 configuration:

```
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# switchport access vlan 17
Device(config-if)# switchport mode dot1q-tunnel
Device(config-if)# l2protocol-tunnel point-to-point pagp
Device(config-if)# l2protocol-tunnel point-to-point udld
Device(config-if)# l2protocol-tunnel drop-threshold point-to-point pagp 1000
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# switchport access vlan 18
Device(config-if)# switchport mode dot1q-tunnel
Device(config-if)# l2protocol-tunnel point-to-point pagp
Device(config-if)# l2protocol-tunnel point-to-point udld
Device(config-if)# l2protocol-tunnel drop-threshold point-to-point pagp 1000
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/3
Device(config-if)# switchport trunk encapsulation isl
Device(config-if)# switchport mode trunk
```

SP edge switch 2 configuration:

```
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# switchport access vlan 19
Device(config-if)# switchport mode dot1q-tunnel
Device(config-if)# l2protocol-tunnel point-to-point pagp
Device(config-if)# l2protocol-tunnel point-to-point udld
Device(config-if)# l2protocol-tunnel drop-threshold point-to-point pagp 1000
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# switchport access vlan 20
Device(config-if)# switchport mode dot1q-tunnel
Device(config-if)# l2protocol-tunnel point-to-point pagp
Device(config-if)# l2protocol-tunnel point-to-point udld
Device(config-if)# l2protocol-tunnel drop-threshold point-to-point pagp 1000
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/3
Device(config-if)# switchport trunk encapsulation isl
Device(config-if)# switchport mode trunk
```

This example shows how to configure the customer switch at Site 1. Fast Ethernet interfaces 1, 2, 3, and 4 are set for IEEE 802.1Q trunking, UDLD is enabled, EtherChannel group 1 is enabled, and the port channel is shut down and then enabled to activate the EtherChannel configuration.

```
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# switchport trunk encapsulation dot1q
Device(config-if)# switchport mode trunk
Device(config-if)# udld enable
Device(config-if)# channel-group 1 mode desirable
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/2
```

```
Device(config-if)# switchport trunk encapsulation dot1q
Device(config-if)# switchport mode trunk
Device(config-if)# udld enable
Device(config-if)# channel-group 1 mode desirable
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/3
Device(config-if)# switchport trunk encapsulation dot1q
Device(config-if)# switchport mode trunk
Device(config-if)# udld enable
Device(config-if)# channel-group 1 mode desirable
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/4
Device(config-if)# switchport trunk encapsulation dot1q
Device(config-if)# switchport mode trunk
Device(config-if)# udld enable
Device(config-if)# channel-group 1 mode desirable
Device(config-if)# exit
Device(config)# interface port-channel 1
Device(config-if)# shutdown
Device(config-if)# no shutdown
Device(config-if)# exit
```

Monitor tunneling status

Lists commands used to monitor tunneling status.

This table describes the commands used to monitor tunneling status.

Table 2: Commands for monitoring tunneling

Command	Purpose
clear l2protocol-tunnel counters	Clears the protocol counters on Layer 2 protocol tunneling ports.
show dot1q-tunnel	Displays IEEE 802.1Q tunnel ports on the device.
show dot1q-tunnel interface *interface-id*	Verifies if a specific interface is a tunnel port.
show l2protocol-tunnel	Displays information about Layer 2 protocol tunneling ports.
show errdisable recovery	Verifies if the recovery timer from a Layer 2 protocol-tunnel error disable state is enabled.
show l2protocol-tunnel interface *interface-id*	Displays information about a specific Layer 2 protocol tunneling port.
show l2protocol-tunnel summary	Displays only Layer 2 protocol summary information.
show VLAN dot1q tag native	Displays the status of native VLAN tagging on the device.

2 Configuring SPAN and RSPAN

Topics:

Information about SPAN and RSPAN

Describes the SPAN (Switched Port Analyzer) and RSPAN (Remote SPAN) features that enable network administrators to monitor and analyze network traffic by copying packets from source ports to destination ports for troubleshooting and security purposes.

The following sections provide information about SPAN and RSPAN.

Prerequisites for SPAN and RSPAN

Outlines prerequisites for SPAN and RSPAN configuration.

SPAN

You can limit SPAN traffic to specific VLANs by using the **filter vlan** keyword. If a trunk port is being monitored, only traffic on the VLANs specified with this keyword is monitored. By default, all VLANs are monitored on a trunk port.

RSPAN

We recommend that you configure an RSPAN VLAN before you configure an RSPAN source or a destination session.

Restrictions for SPAN and RSPAN

Consider these restrictions when configuring SPAN and RSPAN on your device.

SPAN restrictions

SPAN restrictions include these items:

- SPAN VLAN configuration does not support (Egress) Transmit (Tx).
- On each device, you can configure up to two different session IDs and a maximum of two monitor sessions. Each source session is either a local SPAN session or an RSPAN source session.
- For SPAN sources, you can monitor traffic for a single port or VLAN or a series or range of ports or VLANs for each session.

  Note

  You cannot mix source ports and source VLANs within a single SPAN session.

- The destination port and source port must be different; a port cannot serve as both destination and source.
- A source port can be used in only one monitor session.
- You cannot have two SPAN sessions using the same destination port.
- When you configure a device port as a SPAN destination port, it is no longer a normal device port; only monitored traffic passes through the SPAN destination port.
- Entering SPAN configuration commands does not remove previously configured SPAN parameters. You must enter the **no monitor session** {*session_number* | **all** | **local** | **remote**} global configuration command to delete configured SPAN parameters.
- For local SPAN, outgoing packets through the SPAN destination port carry the original encapsulation headers—untagged, ISL, or IEEE 802.1Q—if the **encapsulation replicate** keywords are specified. If the keywords are not specified, the packets are sent in native form.

- You can configure a disabled port as a source or destination, but SPAN does not start until both the destination port and at least one source port or source VLAN are enabled.
- You cannot mix source VLANs and filter VLANs within a single SPAN session.

Traffic monitoring restrictions

SPAN session traffic monitoring has these restrictions.

> **Note**
>
> IE3400 does not support multiple destinations for SPAN traffic.

- Sources can be ports or VLANs, but you cannot mix source ports and source VLANs in the same session.
- Wireshark does not capture egress packets when egress SPAN is active.
- You can run both a local SPAN and an RSPAN source session on the same device. The device supports a total of 2 source and RSPAN destination sessions.
- Both switched and routed ports can be configured as SPAN sources and destinations.
- SPAN sessions do not interfere with the normal operation of the device. However, an oversubscribed SPAN destination, for example, a 10-Mb/s port monitoring a 100-Mb/s port, can result in dropped or lost packets.
- When SPAN or RSPAN is enabled, each packet being monitored is sent twice, once as normal traffic and once as a monitored packet. Monitoring a large number of ports or VLANs could potentially generate large amounts of network traffic.
- You can configure SPAN sessions on disabled ports; however, a SPAN session does not become active unless you enable the destination port and at least one source port or VLAN for that session.
- The device does not support a combination of local SPAN and RSPAN in a single session.
  - An RSPAN source session cannot have a local destination port.
  - An RSPAN destination session cannot have a local source port.
  - An RSPAN destination session and an RSPAN source session that are using the same RSPAN VLAN cannot run on the same device.
- SPAN sessions capture only Dynamic Host Configuration Protocol (DHCP) ingress packets when DHCP snooping is enabled on the device.

RSPAN restrictions

RSPAN has platform-specific and configuration limitations that you must consider.

> **Caution**
>
> Use the RSPAN feature cautiously, as it could impact your production network.

- IE31xx, IE3x00, and ESS3300 platforms support only one RSPAN session, due to a hardware limitation.
- Due to the processing chip's hardware limitation, it cannot differentiate between control packets and data traffic. As a result, enabling mirroring causes all packets—including control packets—to be mirrored.

- The RSPAN VLAN is configured only on trunk ports and not on access ports. To avoid unwanted traffic in RSPAN VLANs, make sure that the VLAN remote-SPAN feature is supported in all the participating devices.
- You can configure the RSPAN VLAN on only one trunk interface. If you attempt to configure remote VLAN on more than one trunk interface, the system displays an error, for example:

```
Switch(config-if)#do sh vlan id 2508

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------

2508 VLAN2508                         active    Gi1/1, Gi1/2
>>>>>>>>>>>>

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2

---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------

2508 enet  102508     1500  -      -      -        -    -        0      0

Remote SPAN VLAN
----------------
Enabled

Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------


Switch(config-if)#exit
Switch(config)#mon sess 1 destination remote vlan 2508
% Platform cannot support remote-span mirroring on VLAN with more than one
member ports.
```

> **Note**
>
> To prevent errors on the receiving interface of the RSPAN caused by the addition of the 4-byte RSPAN VLAN ID, it is recommended to set the MTU size to 4 bytes larger than the Maximum packet size monitored.

- The platform does not support RSPAN mirroring on a VLAN that is associated with a port-channel containing more than one member port.
- RSPAN VLANs are included as sources for port-based RSPAN sessions when source trunk ports have active RSPAN VLANs. RSPAN VLANs can also be sources in SPAN sessions. However, since the device does not monitor spanned traffic, it does not support egress spanning of packets on any RSPAN VLAN identified as the destination of an RSPAN source session on the device.
- If you enable VTP and VTP pruning, RSPAN traffic is pruned in the trunks to prevent the unwanted flooding of RSPAN traffic across the network for VLAN IDs that are lower than 1005.
- Do not configure the RSPAN VLAN as the native VLAN.
- SPAN and RSPAN can work simultaneously.
- ERSPAN only works internally for CV on IOx. It is not supported as an independent feature.
- SPAN and RSPAN are not supported when CV is enabled on IOx.

- When SPAN/RSPAN is configured after CV on IOx enablement, or vice versa, the latest configuration is applied.

SPAN and RSPAN

Describes network traffic analysis technologies that copy traffic to monitoring devices.

SPAN and RSPAN are network traffic mirroring technologies that

- copy traffic from source ports or VLANs to destination ports for analysis by network analyzers or monitoring devices,
- enable traffic analysis without affecting the switching of network traffic on source ports or VLANs, and
- require dedicated destination ports that do not receive or forward regular traffic except for SPAN or RSPAN session requirements.

Traffic monitoring capabilities and limitations

SPAN copies (or mirrors) traffic received or sent (or both) on source ports or source VLANs to a destination port for analysis.

Traffic monitoring has these limitations:

- Only traffic that enters or leaves source ports or traffic that enters or leaves source VLANs can be monitored by using SPAN; traffic routed to a source VLAN cannot be monitored.
- If incoming traffic is being monitored, traffic that gets routed from another VLAN to the source VLAN cannot be monitored; however, traffic that is received on the source VLAN and routed to another VLAN can be monitored.

Network security device integration

You can use the SPAN or RSPAN destination port to inject traffic from a network security device. For example, if you connect a Cisco Intrusion Detection System (IDS) sensor appliance to a destination port, the IDS device can send TCP reset packets to close down the TCP session of a suspected attacker.

Local SPAN

Describes a SPAN session configuration that operates entirely within a single network device.

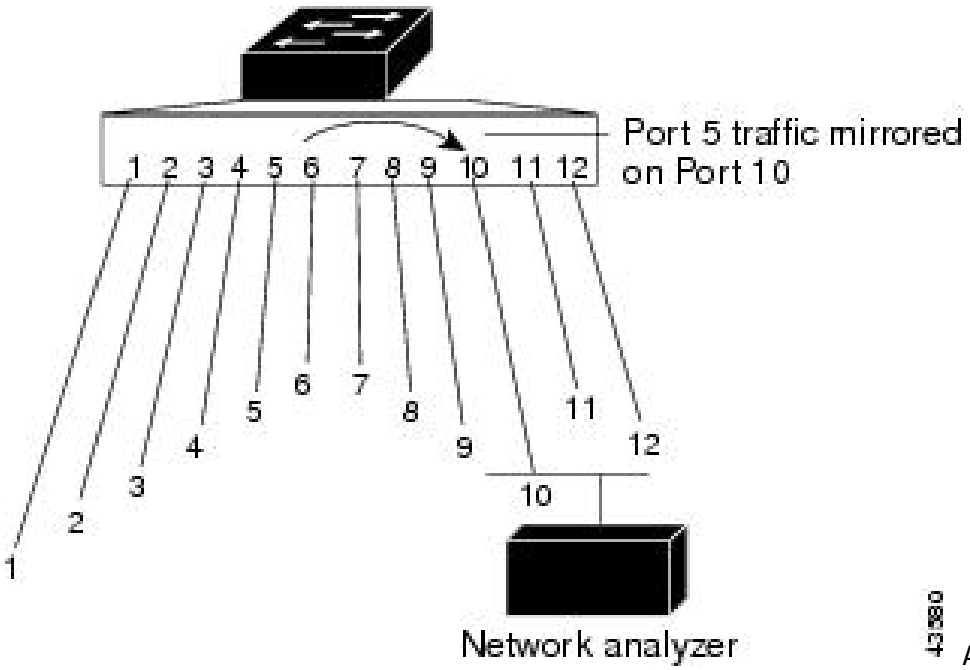
Local SPAN is a network traffic monitoring feature that

- supports a SPAN session entirely within one device or device stack
- places all source ports or source VLANs and destination ports in the same device, and
- copies traffic from one or more source ports in any VLAN or from one or more VLANs to a destination port for analysis.

Local SPAN configuration details

Local SPAN supports a SPAN session entirely within one switch; all source ports and destination ports are in the same switch. Local SPAN copies traffic from one or more source ports to a destination port for analysis.

Figure 4: Example of local SPAN configuration on a single device



All traffic on port 5 (the source port) is mirrored to port 10 (the destination port). A network analyzer on port 10 receives all network traffic from port 5 without being physically attached to port 5.

Remote SPAN

Describes remote SPAN functionality that enables monitoring across multiple network devices.

Remote SPAN is a monitoring technology that supports source ports, source VLANs, and destination ports on different devices, enabling remote monitoring of multiple devices across your network.

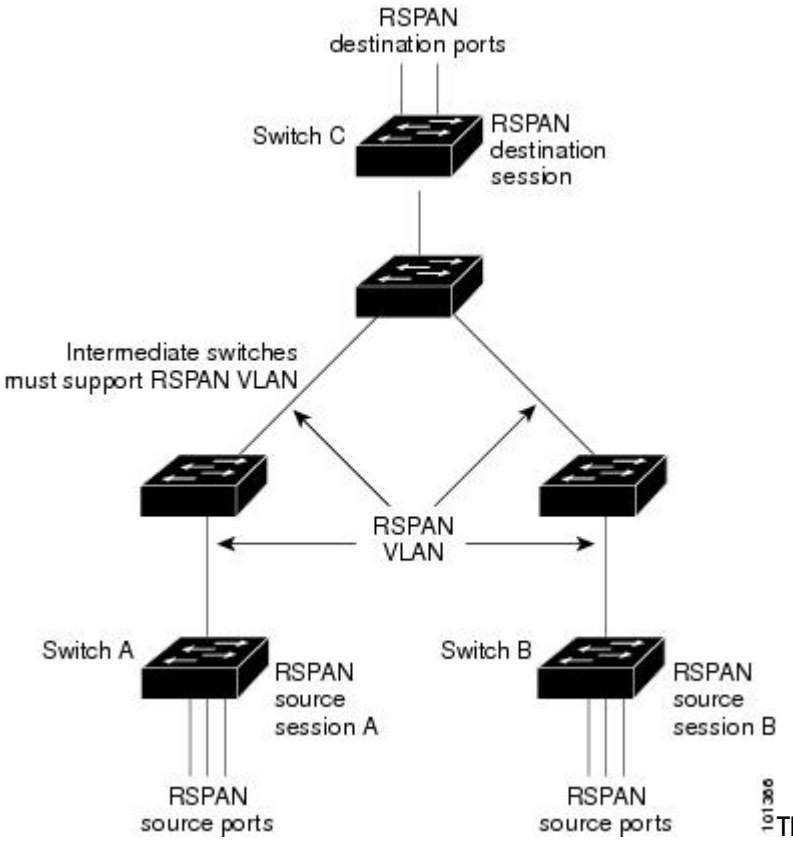
RSPAN configuration details

Each RSPAN session uses a dedicated, user-specified RSPAN VLAN. The RSPAN traffic from the source ports or VLANs is copied into this VLAN, then forwarded over trunk ports that carry the RSPAN VLAN to the destination session. Each RSPAN source device must have either ports or VLANs as sources. The destination is always a physical port.

RSPAN configuration

This figure shows source ports on Device A and Device B, with the destination on Device C.

Figure 5: Example of RSPAN configuration



The figure below shows source ports on Device A and Device B. The traffic for each RSPAN session is carried over a user-specified RSPAN VLAN that is dedicated for that RSPAN session in all participating devices. The RSPAN traffic from the source ports or VLANs is copied into the RSPAN VLAN and forwarded over trunk ports carrying the RSPAN VLAN to a destination session monitoring the RSPAN VLAN. Each RSPAN source device must have either ports or VLANs as RSPAN sources. The destination is always a physical port, as shown on Device C in the figure.

SPAN and RSPAN concepts and terminology

Describes the fundamental concepts and terminology used in Switched Port Analyzer (SPAN) and Remote SPAN (RSPAN) technologies for monitoring network traffic by copying packets from source ports to destination ports for analysis.

SPAN sessions

Describes traffic monitoring sessions that allow you to monitor traffic on ports or VLANs and send monitored traffic to destination ports.

A SPAN session is a traffic monitoring mechanism that

- allows you to monitor traffic on a port or on one or more VLANs,
- sends the monitored traffic to one or more destination ports, and
- can be configured as local SPAN or remote SPAN (RSPAN).

SPAN session types

SPAN sessions include local SPAN sessions and RSPAN sessions:

- **Local SPAN session**: An association of a destination port with source ports or source VLANs, all on a single network device. Local SPAN does not have separate source and destination sessions. Local SPAN sessions gather a set of ingress and egress packets specified by the user. These packets are formed into a stream of SPAN data that is directed to the destination port.

- **RSPAN**: Consists of at least one RSPAN source session, an RSPAN VLAN, and at least one RSPAN destination session. You separately configure RSPAN source sessions and RSPAN destination sessions on different network devices.

RSPAN configuration includes these components:

- **RSPAN source session**: You associate a set of source ports or source VLANs with an RSPAN VLAN. The output of this session is the stream of SPAN packets that are sent to the RSPAN VLAN. This session is similar to a local SPAN session, except SPAN packets are relabeled with the RSPAN VLAN ID and directed over normal trunk ports to the destination device.
- **RSPAN destination session**: You associate the destination port with the RSPAN VLAN. The destination session collects all RSPAN VLAN traffic and sends it out the RSPAN destination port. This session takes all packets received on the RSPAN VLAN, strips off the VLAN tagging, and presents them on the destination port. The session presents a copy of all RSPAN VLAN packets (except Layer 2 control packets) to the user for analysis.

Traffic monitoring in a SPAN session has these restrictions:

- Sources can be ports or VLANs, but you cannot mix source ports and source VLANs in the same session.
- SPAN sessions do not interfere with the normal operation of the device. However, an oversubscribed SPAN destination, for example, a 10-Mb/s port monitoring a 100-Mb/s port, can result in dropped or lost packets.
- When SPAN or RSPAN is enabled, each packet being monitored is sent twice, once as normal traffic and once as a monitored packet. Monitoring a large number of ports or VLANs could generate large amounts of network traffic.
- You can configure SPAN sessions on disabled ports; however, a SPAN session does not become active unless you enable the destination port and at least one source port or VLAN for that session.
- The device does not support a combination of local SPAN and RSPAN in a single session.
  - An RSPAN source session cannot have a local destination port.
  - An RSPAN destination session cannot have a local source port.
  - An RSPAN destination session and an RSPAN source session that are using the same RSPAN VLAN cannot run on the same device or device stack.

Monitored traffic

Describes the types of traffic that SPAN sessions can monitor and their characteristics.

Monitored traffic refers to the network traffic types that SPAN sessions can observe and analyze.

SPAN traffic types

SPAN sessions can monitor these traffic types:

- Receive (Rx) SPAN—Receive (or ingress) SPAN monitors as much as possible all of the packets received by the source interface or VLAN before any modification or processing is performed by the device. A copy of each packet received by the source is sent to the destination port for that SPAN session.

  Packets that are modified because of routing or Quality of Service (QoS)—for example, modified Differentiated Services Code Point (DSCP)—are copied before modification.

  Features that can cause a packet to be dropped during receive processing have no effect on ingress SPAN; the destination port receives a copy of the packet even if the actual incoming packet is dropped. These features include IP standard and extended input Access Control Lists (ACLs), ingress QoS policing, VLAN ACLs, and egress QoS policing.
- Transmit (Tx) SPAN—Transmit (or egress) SPAN monitors as much as possible all of the packets sent by the source interface after all modification and processing is performed by the device. A copy of each packet sent by the source is sent to the destination port for that SPAN session. The copy is provided after the packet is modified.

Packets that are modified because of routing (for example, with modified time-to-live (TTL), MAC address, or QoS values) are duplicated (with the modifications) at the destination port.

Features that can cause a packet to be dropped during transmit processing also affect the duplicated copy for SPAN. These features include IP standard and extended output ACLs and egress QoS policing.

- Both—In a SPAN session, you can monitor a port or VLAN for both received and sent packets. This configuration is the default.

Therefore, a local SPAN session with encapsulation replicate enabled can have a mixture of untagged and IEEE 802.1Q tagged packets appear on the destination port.

Device congestion can cause packets to be dropped at ingress source ports, egress source ports, or SPAN destination ports. These packet-loss scenarios are generally independent. For example:

- A packet might be forwarded normally but dropped from monitoring due to an oversubscribed SPAN destination port.
- An ingress packet might be dropped from normal forwarding, but still appear on the SPAN destination port.
- An egress packet dropped because of device congestion is also dropped from egress SPAN.

In some SPAN configurations, multiple copies of the same source packet are sent to the SPAN destination port. For example, a bidirectional (both Rx and Tx) SPAN session is configured for the Rx monitor on port A and Tx monitor on port B. If a packet enters the device through port A and is switched to port B, both incoming and outgoing packets are sent to the destination port. Both packets are the same unless a Layer 3 rewrite occurs, in which case the packets are different because of the packet modification.

Source ports

Describes switched or routed ports that are monitored for network traffic analysis in SPAN and RSPAN sessions.

A source port is a switched or routed port that

- monitors network traffic for analysis purposes,
- can be configured for traffic monitoring in one or both directions in SPAN sessions, and
- supports any port type including EtherChannel and Gigabit Ethernet.

Source port characteristics

A source port has these characteristics:

- A source port can be used in only one monitor session.
- Each source port can be configured with a direction (ingress, egress, or both) to monitor.
- It can be any port type (for example, EtherChannel, Gigabit Ethernet, and so forth).
- For EtherChannel sources, you can monitor traffic for the entire EtherChannel or individually on a physical port as it participates in the port channel.
- It can be an access port, trunk port, routed port, or voice VLAN port.
- It cannot be a destination port.
- Source ports can be in the same or different VLANs.
- You can monitor multiple source ports in a single session.

The device supports any number of source ports (up to the maximum number of available ports on the device) and any number of source VLANs (up to the maximum number of VLANs supported).

You cannot mix ports and VLANs in a single session.

Source VLANs

Describes VLAN-based SPAN monitoring capabilities and characteristics for network traffic analysis.

VLAN-based SPAN (VSPAN) is a network monitoring feature that

- monitors network traffic in one or more VLANs,
- uses a VLAN ID as the SPAN or RSPAN source interface, and
- monitors traffic on all ports for that VLAN.

VSPAN characteristics

VSPAN has these characteristics:

- All active ports in the source VLAN are included as source ports and can be monitored in either or both directions.
- On a given port, only traffic on the monitored VLAN is sent to the destination port.
- If a destination port belongs to a source VLAN, it is excluded from the source list and is not monitored.
- If ports are added to or removed from the source VLANs, the traffic on the source VLAN received by those ports is added to or removed from the sources being monitored.
- You cannot use filter VLANs in the same session with VLAN sources.
- You can monitor only Ethernet VLANs.

VLAN filtering

Describes a method to limit SPAN traffic monitoring on trunk source ports to specific VLANs.

VLAN filtering is a SPAN monitoring feature that

- limits SPAN traffic monitoring on trunk source ports to specific VLANs,
- applies only to trunk ports or to voice VLAN ports, and
- affects only traffic forwarded to the destination SPAN port without affecting the switching of normal traffic.

VLAN filtering characteristics

When you monitor a trunk port as a source port, by default, all VLANs active on the trunk are monitored. You can limit SPAN traffic monitoring on trunk source ports to specific VLANs by using VLAN filtering.

VLAN filtering has these characteristics:

- VLAN filtering applies only to trunk ports or to voice VLAN ports.
- VLAN filtering applies only to port-based sessions and is not allowed in sessions with VLAN sources.
- When a VLAN filter list is specified, only those VLANs in the list are monitored on trunk ports or on voice VLAN access ports.
- VLAN filtering does not affect SPAN traffic from other port types. All VLANs are allowed on these ports.
- VLAN filtering affects only traffic forwarded to the destination SPAN port and does not affect the switching of normal traffic.

Destination port

Describes the monitoring port that receives SPAN traffic copies and forwards them to network analysis tools.

A destination port is a monitoring port that

- receives a copy of traffic from the source ports or VLANs in SPAN sessions,
- sends the SPAN packets to the user, usually a network analyzer, and
- must be present in each local SPAN session or RSPAN destination session.

Destination port characteristics

A destination port has these characteristics:

- For a local SPAN session, the destination port must reside on the same device or device stack as the source port. For an RSPAN session, it is located on the device containing the RSPAN destination session. There is no destination port on a device or device stack running only an RSPAN source session.
- When a port is configured as a SPAN destination port, the configuration overwrites the original port configuration. When the SPAN destination configuration is removed, the port reverts to its previous configuration. If a configuration change is made to the port while it is acting as a SPAN destination port, the change does not take effect until the SPAN destination configuration has been removed.

  Note

  When QoS is configured on the SPAN destination port, QoS takes effect immediately.

- If the port belonged to an EtherChannel group, it is removed from the group while it acts as a destination port. If the port functioned as a routed port, it no longer operates in this role.
- It can be any Ethernet physical port.
- It cannot be a secure port.
- It cannot be a source port.
- It can participate in only one SPAN session at a time (a destination port in one SPAN session cannot be a destination port for a second SPAN session).
- When it is active, incoming traffic is disabled. The port does not transmit any traffic except that required for the SPAN session. Incoming traffic is never learned or forwarded on a destination port.
- If ingress traffic forwarding is enabled for a network security device, the destination port forwards traffic at Layer 2.
- It does not participate in any of the Layer 2 protocols (STP, VTP, CDP, DTP, PagP).
- A destination port that belongs to a source VLAN of any SPAN session is excluded from the source list and is not monitored.
- The maximum number of destination ports in a device or device stack is 64.

Local SPAN and RSPAN destination ports function differently with VLAN tagging and encapsulation:

- For local SPAN, if the **encapsulation replicate** keywords are specified for the destination port, these packets appear with the original encapsulation (untagged, ISL, or IEEE 802.1Q). If these keywords are not specified, packets appear in the untagged format. Therefore, the output of a local SPAN session with **encapsulation replicate** enabled can contain a mixture of untagged, ISL, or IEEE 802.1Q-tagged packets.
- For RSPAN, the RSPAN VLAN ID is inserted into the packet while preserving the original VLAN ID.

RSPAN VLAN

Describes a specialized VLAN that carries SPAN traffic between RSPAN source and destination sessions with specific characteristics and configuration requirements.

An RSPAN VLAN is a specialized VLAN that

- carries SPAN traffic between RSPAN source and destination sessions,
- floods all traffic without MAC address learning, and
- flows only on trunk ports with specific configuration requirements.

RSPAN VLAN characteristics

RSPAN VLAN has these special characteristics:

- All traffic in the RSPAN VLAN is always flooded.
- No MAC address learning occurs on the RSPAN VLAN.
- RSPAN VLAN traffic only flows on trunk ports.
- RSPAN VLANs must be configured in VLAN configuration mode by using the **remote-SPAN** VLAN configuration mode command.
- STP can run on RSPAN VLAN trunks but not on SPAN destination ports.
- An RSPAN VLAN cannot be a private-VLAN primary or secondary VLAN.

For VLANs 1 to 1005 that are visible to VLAN Trunking Protocol (VTP), the VLAN ID and its associated RSPAN characteristic are propagated by VTP. If you assign an RSPAN VLAN ID in the extended VLAN range (1006 to 4094), you must manually configure all intermediate devices.

Multiple RSPAN VLANs can exist in a network at the same time, with each VLAN defining a network-wide RSPAN session. Multiple RSPAN source sessions in the network can contribute packets to a session. Multiple RSPAN destination sessions can monitor the same RSPAN VLAN and present traffic to the user. The RSPAN VLAN ID separates these sessions.

SPAN and RSPAN interaction with other features

Lists the features that interact with SPAN and RSPAN and describes how they work together.

SPAN interacts with various network features in specific ways that affect monitoring functionality, traffic handling, and port behavior.

- Routing—SPAN does not monitor routed traffic. VSPAN only monitors traffic that enters or exits the VLAN. It does not monitor traffic that is routed between VLANs. For example, if a VLAN is being Rx-monitored and the device routes traffic from another VLAN to the monitored VLAN, that traffic is not monitored or received on the SPAN destination port.
- STP—A destination port does not participate in STP while its SPAN or RSPAN session is active. The destination port can participate in STP after the SPAN or RSPAN session is disabled. On a source port, SPAN does not affect the STP status. STP can be active on trunk ports carrying an RSPAN VLAN.
- CDP—A SPAN destination port does not participate in CDP while the SPAN session is active. After the SPAN session is disabled, the port again participates in CDP.
- VTP—You can use VTP to prune an RSPAN VLAN between the source and destination session .
- VLAN and trunking—You can modify VLAN membership or trunk settings for source or destination ports at any time. However, changes in VLAN membership or trunk settings for a destination port do not take effect until you remove the SPAN destination configuration. Changes in VLAN membership or trunk settings for a source port immediately take effect, and the respective SPAN sessions automatically adjust accordingly.

- EtherChannel–You can configure an EtherChannel group as a source port or as a SPAN destination port. When a group is configured as a SPAN source, the entire group is monitored.

  If a physical port is added to a monitored EtherChannel group, the new port is added to the SPAN source port list. If a port is removed from a monitored EtherChannel group, it is automatically removed from the source port list.

  A physical port that belongs to an EtherChannel group can be configured as a SPAN source port and still be a part of the EtherChannel. In this case, data from the physical port is monitored as it participates in the EtherChannel. However, if a physical port that belongs to an EtherChannel group is configured as a SPAN destination, it is removed from the group. After the port is removed from the SPAN session, it rejoins the EtherChannel group. Ports removed from an EtherChannel group remain members of the group, but they are in the inactive or suspended state.

  If a physical port that belongs to an EtherChannel group is a destination port and the EtherChannel group is a source, the port is removed from the EtherChannel group and from the list of monitored ports.
- Multicast traffic can be monitored. For egress and ingress port monitoring, only a single unedited packet is sent to the SPAN destination port. It does not reflect the number of times the multicast packet is sent.
- A private-VLAN port cannot be a SPAN destination port.
- A secure port cannot be a SPAN destination port.

  For SPAN sessions, do not enable port security on ports with monitored egress when ingress forwarding is enabled on the destination port. For RSPAN source sessions, do not enable port security on any ports with monitored egress.
- An IEEE 802.1x port can be a SPAN source port. You can enable IEEE 802.1x on a port that is a SPAN destination port; however, IEEE 802.1x is disabled until the port is removed as a SPAN destination.

  For SPAN sessions, do not enable IEEE 802.1x on ports with monitored egress when ingress forwarding is enabled on the destination port. For RSPAN source sessions, do not enable IEEE 802.1x on any ports that are egress monitored.

Flow-based SPAN

Describes a network monitoring feature that applies access control lists to filter traffic in SPAN or RSPAN sessions.

Flow-based SPAN (FSPAN) is a network monitoring feature that

- controls the type of network traffic to be monitored in SPAN or RSPAN sessions by applying access control lists (ACLs) to the monitored traffic on the source ports,
- filters IPv4, IPv6, and non-IP monitored traffic through configurable FSPAN ACLs, and
- copies only packets permitted by the ACL to the SPAN destination port while allowing original traffic to continue forwarding normally.

FSPAN ACL types and operation

You apply an ACL to a SPAN session through the interface. It is applied to all the traffic that is monitored on all interfaces in the SPAN session. The packets that are permitted by this ACL are copied to the SPAN destination port. No other packets are copied to the SPAN destination port.

The original traffic continues to be forwarded, and any port, VLAN, and router ACLs attached are applied.The FSPAN ACL does not affect forwarding decisions. Similarly, the port, VLAN, and router ACLs do not affect traffic monitoring. If a security input ACL denies a packet and it is not forwarded, the packet is still copied to the SPAN destination ports if the FSPAN ACL permits it. If the security output ACL denies a packet and it is not sent, the switch does not copy it to the SPAN destination ports. If the security output ACL permits the packet to go out, the switch copies it to the SPAN destination ports only if the FSPAN ACL also permits it. The same behavior applies to an RSPAN session.

You can attach three types of FSPAN ACLs to the SPAN session:

- IPv4 FSPAN ACL filters only IPv4 packets.
- IPv6 FSPAN ACL filters only IPv6 packets.

- MAC FSPAN ACL filters only non-IP packets.

If a VLAN-based FSPAN session configured on a stack cannot fit in the hardware memory on one or more devices, it is treated as unloaded on those devices, and traffic meant for the FSPAN ACL and sourcing on that device is not copied to the SPAN destination ports. The FSPAN ACL continues to be correctly applied, and traffic is copied to the SPAN destination ports on the devices where the FSPAN ACL fits in the hardware memory.

When an empty FSPAN ACL is attached, some hardware functions copy all traffic to the SPAN destination ports for that ACL. If sufficient hardware resources are not available, even an empty FSPAN ACL can be unloaded.

Default SPAN and RSPAN configuration

Lists the default configuration settings for SPAN and RSPAN features.

Provides the default configuration settings for SPAN and RSPAN features on network switches.

Table 3: Default SPAN and RSPAN configuration

Feature	Default Setting
SPAN state (SPAN and RSPAN)	Disabled.
Source port traffic to monitor	Both received and sent traffic (**both**).
Encapsulation type (destination port)	Native form (untagged packets).
Ingress forwarding (destination port)	Disabled.
VLAN filtering	On a trunk interface used as a source port, all VLANs are monitored.
RSPAN VLANs	None configured.

Configuring SPAN and RSPAN

Describes how to configure Switched Port Analyzer (SPAN) and Remote SPAN (RSPAN) to monitor network traffic by copying packets from source ports to destination ports for analysis and troubleshooting.

Best practice for SPAN configuration

Outlines guidelines for configuring SPAN sessions, including removing source and destination ports or VLANs and monitoring all VLANs on trunk ports.

Follow these guidelines when you configure SPAN sessions to ensure proper monitoring and the correct removal of ports or VLANs.

- To remove a source or destination port or VLAN from the SPAN session, use the **no monitor session** *session_number* **source interface** *interface-id* {**interface** *interface-id* | **VLAN** *VLAN-id*} global configuration command, the **no monitor session all** global configuration command or the **no monitor session** *session_number* **destination interface** *interface-id* global configuration command. For destination interfaces, the **encapsulation** options are ignored with the **no** form of the command.
- To monitor all VLANs on the trunk port, use the **no monitor session** *session_number* **filter** global configuration command.

RSPAN configuration guidelines

Consider these guidelines when configuring Remote Switched Port Analyzer (RSPAN) to ensure proper network monitoring functionality.

Follow these guidelines when configuring RSPAN to ensure proper functionality and network performance.

- All the SPAN configuration guidelines apply to RSPAN.
- As RSPAN VLANs have special properties, you should reserve a few VLANs across your network for use as RSPAN VLANs; do not assign access ports to these VLANs.
- You can apply an output ACL to RSPAN traffic to selectively filter or monitor specific packets. Specify these ACLs on the RSPAN VLAN in the RSPAN source.
- For RSPAN configuration, you can distribute the source ports and the destination ports across multiple devices in your network.
- Access ports (including voice VLAN ports) on the RSPAN VLAN are put in the inactive state.
- You can configure any VLAN as an RSPAN VLAN as long as these conditions are met:
  - The same RSPAN VLAN is used for an RSPAN session.
  - All the VLANs participating in an RSPAN session use the same RSPAN VLAN.

FSPAN and FRSPAN configuration guidelines

Outlines guidelines for configuring FSPAN and FRSPAN to ensure proper traffic monitoring and filtering behavior.

Follow these guidelines when configuring FSPAN and FRSPAN:

- When at least one FSPAN ACL is attached, FSPAN is enabled.
- If you attach at least one FSPAN ACL that is not empty to a SPAN session, but do not attach one or more of the other FSPAN ACLs (for example, you attach a non-empty IPv4 ACL and do not attach IPv6 or MAC ACLs), FSPAN blocks any traffic that would have been filtered by the unattached ACLs. As a result, that traffic is not monitored.

How to configure SPAN and RSPAN

Describes how to configure Switched Port Analyzer (SPAN) and Remote SPAN (RSPAN) for network traffic monitoring and analysis on Cisco switches.

The following sections provide information on how to configure SPAN and RSPAN.

Create a local SPAN session

Configure a local SPAN session to specify the source (monitored) ports or VLANs and the destination (monitoring) ports.

Create a SPAN session to monitor network traffic by copying packets from source ports to destination ports for analysis.

Use this task to analyze network traffic or to troubleshoot connectivity issues by mirroring traffic from monitored interfaces to monitoring ports

Follow these steps to create a SPAN session and specify the source (monitored) ports or VLANs and the destination (monitoring) ports.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **no monitor session {session-number | all | local | remote}** command to clear the specified monitor session configuration.

   Example

   ```
   Device(config)# no monitor session all
   ```

   - For *session_number*, the range is 1 to 2.
   - **all**–Removes all SPAN sessions.
   - **local**–Removes all local sessions.
   - **remote**–Removes all remote SPAN sessions.

3. Use the **monitor session session-number source {interface interface-id} [, | -] [both | rx | tx]** command to specify the source traffic for the monitor session.

   Example

   ```
   Device(config)# monitor session 2 source interface gigabitethernet1/1 rx
   ```

   - For *session_number*, the range is 1 to 2.
   - For *interface-id*, specify the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel***port-channel-number*). Valid port-channel numbers are 1 to 6.
   - (Optional) [**,** | -] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.
   - (Optional) **both** | **rx** | **tx**–Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both received and sent traffic.
     - **both**–Monitors both received and sent traffic.
     - **rx**–Monitors received traffic.
     - **tx**–Monitors sent traffic.

> **Note**
>
> You can use the **monitor session***session_number***source** command multiple times to configure multiple source ports.

> **Note**
>
> A single session can include multiple sources (ports or VLANs), but you cannot combine source ports and source VLANs in the same session.

4. Use the **monitor session session-number destination {interface interface-id [, | -] [encapsulation replicate] [ingress {vlan vlan-id}]}** command to configure the destination interface for the monitor session.

   Example

```
Device(config)# monitor session 2 destination interface gigabitethernet1/2
encapsulation replicate ingress vlan 10
```

   > **Note**
   >
   > For local SPAN, you must use the same session number for the source and destination interfaces.

   - For *session_number*, specify the session number entered in step 4.
   - For *interface-id*, specify the destination port. The destination interface must be a physical port. It cannot be an EtherChannel or a VLAN.
   - (Optional) [, | -] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.
   - (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged).
   - **ingress** enables forwarding of incoming traffic on the destination port and to specify the encapsulation type:
     - **VLAN***VLAN-id*–Accepts incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN.

> **Note**
>
> You can use **monitor session***session_number***destination** command multiple times to configure multiple destination ports.

5. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# end
   ```

6. (Optional) Use the **show running-config** command to verify your entries.

   Example

   ```
   Device# show running-config
   ```

Create a local SPAN session and configure incoming traffic

Configure a SPAN session to specify source ports or VLANs and destination ports, and enable incoming traffic on the destination port for a network security device.

This task creates a SPAN session that enables monitoring of network traffic and allows incoming traffic on the destination port for network security devices such as a Cisco IDS Sensor Appliance.

Use this procedure when you need to monitor network traffic and enable bidirectional traffic flow for network security analysis.

Before you begin

Follow these steps to create a SPAN session, to specify the source ports or VLANs and the destination ports, and to enable incoming traffic on the destination port for a network security device (such as a Cisco IDS Sensor Appliance):

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **no monitor session {session-number | all | local | remote}** command to clear the specified monitor session configuration.

   Example

   ```
   Device(config)# no monitor session all
   ```

   - For *session_number*, the range is 1 to 2.

- **all**–Removes all SPAN sessions.
- **local**–Removes all local sessions.
- **remote**–Removes all remote SPAN sessions.

3. Use the **monitor session session-number source {interface interface-id} [, | -] [both | rx | tx]** command to specify the source traffic for the monitor session.

   Example

```
Device(config)# monitor session 2 source interface gigabitethernet1/1 rx
```

4. Use the **monitor session session-number destination {interface interface-id [, | -] [encapsulation replicate] [ingress {vlan vlan-id}]}** command to configure the destination interface for the monitor session.

   Example

```
Device(config)# monitor session 2 destination interface gigabitethernet1/2
encapsulation replicate ingress vlan 10
```

   - For *session_number*, specify the session number entered in Step 4.
   - For *interface-id*, specify the destination port. The destination interface must be a physical port; it cannot be an EtherChannel or a VLAN.
   - (Optional) **[, | -]**–Specifies a series or range of interfaces. Enter a space before and after the comma or hyphen.
   - (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If you do not select this option, the system sends packets in native form (untagged) by default.
   - **ingress** enables forwarding of incoming traffic on the destination port and to specify the encapsulation type:
     - **VLAN***VLAN-id*–Accepts incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN.

5. Use the **end** command to return to privileged EXEC mode.

   Example

```
Device(config)# end
```

6. (Optional) Use the **show running-config** command to verify your entries.

   Example

```
Device# show running-config
```

Specify VLANs to filter (SPAN)

Configure SPAN source traffic filtering to limit monitoring to specific VLANs.

This task allows you to limit SPAN source traffic to specific VLANs, providing more focused network monitoring and reducing unnecessary traffic analysis.

Use VLAN filtering with SPAN sessions to monitor only the traffic from specified VLANs, instead of monitoring all traffic on a trunk port. This approach optimizes monitoring resources and focuses analysis on relevant network segments.

Before you begin

The interface specified as the source port must already be configured as a trunk port.

Follow these steps to limit SPAN source traffic to specific VLANs:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **no monitor session {session-number | all | local | remote}** command to clear the specified monitor session configuration.

   Example

   ```
   Device(config)# no monitor session all
   ```

   - For *session_number*, the range is 1 to 2.
   - **all**–Removes all SPAN sessions.
   - **local**–Removes all local sessions.
   - **remote**–Removes all remote SPAN sessions.

3. Use the **monitor session session-number source {interface interface-id} [, | -] [both | rx | tx]** command to specify the source traffic for the monitor session.

   Example

   ```
   Device(config)# monitor session 2 source interface gigabitethernet1/2 rx
   ```

   - For *session_number*, the range is 1 to 2.
   - For *interface-id*, specify the source port to monitor. The interface specified must already be configured as a trunk port.

4. Use the **monitor session session-number filter vlan vlan-id [, | -]** command to configure the VLAN filter for the monitor session.

   Example

   ```
   Device(config)# monitor session 2 filter vlan 1 - 5 , 9
   ```

   - For *session_number*, enter the session number specified earlier.
   - For *VLAN-id*, the range is 1 to 4094.

- (Optional) Use a comma (,) to specify a series of VLANs, or use a hyphen (-) to specify a range of VLANs. Enter a space before and after the comma; enter a space before and after the hyphen.

5. Use the **monitor session session-number destination {interface interface-id [, | -] [encapsulation replicate] [ingress {vlan vlan-id}]}** command to configure the destination interface for the monitor session.

   Example

```
Device(config)# monitor session 2 destination interface gigabitethernet1/1
ingress vlan 6
```

   - For *session_number*, specify the session number entered in Step 4.
   - For *interface-id*, specify the destination port. The destination interface must be a physical port; it cannot be an EtherChannel, and it cannot be a VLAN.
   - (Optional) [, | -] Specifies a series or range of interfaces. Place a space before and after the comma, and before and after the hyphen.
   - (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged).
   - **ingress** enables forwarding of incoming traffic on the destination port and to specify the encapsulation type:
     - **VLAN***VLAN-id*–Accepts incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN.

6. Use the **end** command to return to privileged EXEC mode.

   Example

```
Device(config)# end
```

Configure a VLAN as an RSPAN VLAN

Configure a new VLAN and designate it as the RSPAN VLAN for the RSPAN session.

This task enables you to create a dedicated VLAN for Remote Switched Port Analyzer (RSPAN) monitoring sessions, allowing traffic monitoring across multiple switches in the network.

RSPAN requires a dedicated VLAN to carry monitored traffic between source and destination switches. This VLAN cannot be used for regular data traffic. You must configure this VLAN on all participating switches.

Before you begin

Create a new VLAN, and then configure it as the RSPAN VLAN for the RSPAN session.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

```
Device# configure terminal
```

2. Use the **vlan** *vlan-id* command to create or modify the VLAN and enter VLAN configuration mode.

   Example

   ```
   Device(config)# vlan 100
   ```

   The RSPAN VLAN cannot be VLAN 1 (the default VLAN) or VLAN IDs 1002 through 1005 (reserved for Token Ring and FDDI VLANs).

3. Use the **remote-span** command to configure the VLAN as a remote SPAN VLAN.

   Example

   ```
   Device(config-vlan)# remote-span
   ```

4. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config-vlan)# end
   ```

5. (Optional) Use the **show running-config** command to verify your entries.

   Example

   ```
   Device# show running-config
   ```

What to do next

You must create the RSPAN VLAN in all devices that will participate in RSPAN.

If the RSPAN VLAN-ID is in the normal range (lower than 1005) and VTP is enabled in the network, you can create the RSPAN VLAN in one device, and VTP propagates it to the other devices in the VTP domain.

For extended-range VLANs (greater than 1005), you must configure RSPAN VLAN on both source and destination devices and any intermediate devices

Use VTP pruning to ensure an efficient flow of RSPAN traffic. Alternatively, manually delete the RSPAN VLAN from all trunks that do not need to carry RSPAN traffic.

To remove the remote SPAN characteristic from a VLAN and convert it back to a normal VLAN, use the **no remote-SPAN** VLAN configuration command.

To remove a source port or VLAN from the SPAN session, use the **no monitor session***session_number***source {interface** *interface-ID* **| VLAN***VLAN-ID***}** global configuration command. To remove the RSPAN VLAN from the session, use the **no monitor session***session_number***destination remote VLAN***VLAN-ID*.

Create an RSPAN source session

Configure an RSPAN source session to monitor traffic and specify the destination RSPAN VLAN.

Creating an RSPAN source session allows you to monitor network traffic from source interfaces. It also forwards the monitored traffic to a remote SPAN VLAN for analysis.

Use RSPAN source sessions when you need to monitor traffic from interfaces on one switch and analyze that traffic on a different switch in the network. The source session captures the specified traffic and forwards it to an RSPAN VLAN for remote monitoring.

Before you begin

Follow these steps to create and start an RSPAN source session and to specify the monitored source and the destination RSPAN VLAN.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **no monitor session {session-number | all | local | remote}** command to clear the specified monitor session configuration.

   Example

   ```
   Device(config)# no monitor session 1
   ```

   - For *session_number*, the range is 1 to 2.
   - **all**–Removes all SPAN sessions.
   - **local**–Removes all local sessions.
   - **remote**–Removes all remote SPAN sessions.

3. Use the **monitor session session-number source {interface interface-id} [, | -] [both | rx | tx]** command to specify the source traffic for the monitor session.

   Example

   ```
   Device(config)# monitor session 1 source interface gigabitethernet1/1 tx
   ```

   - For *session_number*, the range is 1 to 2.
   - Enter a source port or source VLAN for the RSPAN session:
     - For *interface-id*, specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel***port-channel-number*). Valid port-channel numbers are 1 to 6.
     - A single session can include multiple sources, such as ports or VLANs, defined in a series of commands. However, you cannot combine source ports and source VLANs in one session.
   - (Optional) **[, | -]**–Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.
   - (Optional) **both | rx | tx**–Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both sent and received traffic.
     - **both**–Monitors both received and sent traffic.
     - **rx**–Monitors received traffic.

- **tx**–Monitors sent traffic.

4. Use the **monitor session session-number destination remote vlan vlan-id** command to configure the remote VLAN destination for the monitor session.

   Example

   ```
   Device(config)# monitor session 1 destination remote vlan 100
   ```

   - For *session_number*, enter the number defined in Step 4.
   - For *VLAN-id*, specify the source RSPAN VLAN to monitor.

5. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# end
   ```

6. (Optional) Use the **show running-config** command to verify your entries.

   Example

   ```
   Device# show running-config
   ```

Specify VLANs to filter (RSPAN)

Configure the RSPAN source session to limit RSPAN source traffic to specific VLANs.

Configure RSPAN source sessions to monitor only the VLANs that require analysis. This reduces unnecessary network overhead and improves monitoring efficiency.

Use this procedure to monitor only specific VLAN traffic instead of all traffic from the source interface. This approach reduces monitored traffic volume and focuses analysis on relevant network segments.

Before you begin

The source interface must already be configured as a trunk port.

Follow these steps to configure the RSPAN source session to limit RSPAN source traffic to specific VLANs:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **no monitor session {session-number | all | local | remote}** command to clear the specified monitor session configuration.

Example

```
Device(config)# no monitor session 2
```

- For *session_number*, the range is 1 to 2.
- **all**–Removes all SPAN sessions.
- **local**–Removes all local sessions.
- **remote**–Removes all remote SPAN sessions.

3. Use the **monitor session session-number source {interface interface-id} [, | -] [both | rx | tx]** command to specify the source traffic for the monitor session.

   Example

   ```
   Device(config)# monitor session 2 source interface gigabitethernet1/2 rx
   ```

   - For *session_number*, the range is 1 to 2.
   - For *interface-id*, specify the source port to monitor. The interface specified must already be configured as a trunk port.
   - (Optional) **[, | -]**–Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.
   - (Optional) **both | rx | tx**–Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both sent and received traffic.
     - **both**–Monitors both received and sent traffic.
     - **rx**–Monitors received traffic.
     - **tx**–Monitors sent traffic.

4. Use the **monitor session session-number filter vlan vlan-id [, | -]** command to configure the VLAN filter for the monitor session.

   Example

   ```
   Device(config)# monitor session 2 filter vlan 1 - 5 , 9
   ```

   - For *session_number*, enter the session number specified earlier
   - For *VLAN-id*, the range is 1 to 4094.
   - (Optional) **, | -** Use a comma (,) to specify a series of VLANs or use a hyphen (-) to specify a range of VLANs. Enter a space before and after the comma; enter a space before and after the hyphen.

5. Use the **monitor session session-number destination remote vlan vlan-id** command to configure the remote VLAN destination for the monitor session.

   Example

   ```
   Device(config)# monitor session 2 destination remote vlan 902
   ```

   - For *session_number*, enter the session number specified in Step 4.

- For *VLAN-id*, specify the RSPAN VLAN to carry the monitored traffic to the destination port.

6. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# end
   ```

7. (Optional) Use the **show running-config** command to verify your entries.

   Example

   ```
   Device# show running-config
   ```

Create an RSPAN destination session

Configure an RSPAN destination session on a different device or device stack from the source session to define the RSPAN VLAN and specify the source RSPAN VLAN and destination port.

Create an RSPAN destination session to monitor traffic from a remote source VLAN on a different device or device stack.

You configure an RSPAN destination session on a device or device stack that is different from the one used for the source session.

Before you begin

Define the RSPAN VLAN on that device, create an RSPAN destination session, and specify the source RSPAN VLAN and the destination port.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **vlan vlan-id** command to create or modify the VLAN and enter VLAN configuration mode.

   Example

   ```
   Device(config)# vlan 901
   ```

   If both devices are participating in VTP and the RSPAN VLAN ID is from 2 to 1005, Steps 3 through 5 are not required because the RSPAN VLAN ID is propagated through the VTP network.

3. Use the **remote-span** command to configure the VLAN as a remote SPAN VLAN.

   Example

   ```
   Device(config-vlan)# remote-span
   ```

4. Use the **exit** command to return to global configuration mode.

Example

```
Device(config-vlan)# exit
```

5. Use the **no monitor session {session-number | all | local | remote}** command to clear the specified monitor session configuration.

   Example

   ```
   Device(config)# no monitor session 1
   ```

   - For *session_number*, the range is 1 to 2.
   - **all**–Removes all SPAN sessions.
   - **local**–Removes all local sessions.
   - **remote**–Removes all remote SPAN sessions.

6. Use the **monitor session session-number source remote vlan vlan-id** command to specify the RSPAN session and the source RSPAN VLAN.

   Example

   ```
   Device(config)# monitor session 1 source remote vlan 901
   ```

   - For *session_number*, the range is 1 to 2.
   - For *VLAN-ID*, specify the source RSPAN VLAN to monitor.

7. Use the **monitor session session-number destination {interface interface-id [, | -] [encapsulation replicate] [ingress {vlan vlan-id}]}** command to configure the destination interface for the monitor session.

   Example

   ```
   Device(config)# monitor session 1 destination interface gigabitethernet1/2
   encapsulation replicate ingress vlan 10
   ```

   - For *session_number*, enter the number defined in Step 7.

     In an RSPAN destination session, you must use the same session number for the source RSPAN VLAN and the destination port.
   - For *interface-ID*, specify the destination interface. The destination interface must be a physical interface.
   - (Optional) [, | -] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.
   - (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged).
   - **ingress** enables forwarding of incoming traffic on the destination port and to specify the encapsulation type:
     - **VLAN***VLAN-ID*–Accepts incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN.
   - You can use **monitor session***session_number***destination** command multiple times to configure multiple destination ports.

8. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# end
   ```

9. (Optional) Use the **show running-config** command to verify your entries.

   Example

   ```
   Device# show running-config
   ```

Create an RSPAN destination session and configure incoming traffic

Configure an RSPAN destination session with a specified source RSPAN VLAN and destination port, and enable incoming traffic on the destination port for a network security device.

Create an RSPAN destination session to monitor traffic from a remote SPAN VLAN and enable bidirectional traffic flow for network security devices such as Cisco IDS Sensor Appliances.

Use this procedure to set up remote monitoring of network traffic, enabling the monitoring device to inject traffic back into the network for security analysis or response.

Before you begin

To create an RSPAN destination session, specify the source RSPAN VLAN and the destination port, and enable incoming traffic on the destination port for a network security device such as a Cisco IDS Sensor Appliance.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **no monitor session {session-number | all | local | remote}** command to clear the specified monitor session configuration.

   Example

   ```
   Device(config)# no monitor session all
   ```

   - For *session_number*, the range is 1 to 2.
   - **all**–Removes all SPAN sessions.
   - **local**–Removes all local sessions.
   - **remote**–Removes all remote SPAN sessions.

3. Use the **monitor session session-number source remote vlan vlan-id** command to specify the remote VLAN source for the monitor session.

Example

```
Device(config)# monitor session 2 source remote vlan 901
```

- For *session_number*, the range is 1 to 2.
- For *VLAN-ID*, specify the source RSPAN VLAN to monitor.

4. Use the **monitor session session-number destination {interface interface-id [, | -] [encapsulation replicate] [ingress {vlan vlan-id}]}** command to configure the destination interface for the monitor session.

Example

```
Device(config)# monitor session 2 destination interface gigabitethernet1/2
encapsulation replicate ingress vlan 10
```

- For *session_number*, enter the number defined in Step 5.

  In an RSPAN destination session, you must use the same session number for the source RSPAN VLAN and the destination port.
- For *interface-ID*, specify the destination interface. The destination interface must be a physical interface.
- Encapsulation replicate appears in the command-line help string, but it is not supported for RSPAN. The system overwrites the original VLAN ID with the RSPAN VLAN ID, so all packets appear on the destination port as untagged.
- (Optional) [, | -] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.
- Enter **ingress** with additional keywords to enable forwarding of incoming traffic on the destination port and to specify the encapsulation type:
  - **VLAN***VLAN-ID*–Forwards incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN.

5. Use the **end** command to return to privileged EXEC mode.

Example

```
Device(config)# end
```

6. (Optional) Use the **show running-config** command to verify your entries.

Example

```
Device# show running-config
```

Configure an FSPAN session

Configure a Flexible Switch Port Analyzer (FSPAN) session to monitor network traffic by specifying source and destination ports with filtering capabilities.

Configure FSPAN to monitor network traffic. Use FSPAN for troubleshooting, security analysis, or performance optimization by creating sessions that capture and forward packets from source ports to destination ports with optional filtering.

FSPAN sessions allow you to monitor traffic on specific ports or VLANs by copying packets to a destination port where they can be analyzed. Use FSPAN to inspect network traffic without disrupting normal operations.

Before you begin

To create a SPAN session, specify the source (monitored) ports or VLANs and the destination (monitoring) ports. Configure FSPAN for the session.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **no monitor session {session-number | all | local | remote}** command to clear the specified monitor session configuration.

   Example

   ```
   Device(config)# no monitor session all
   ```

   - For *session_number*, the range is 1 to 2.
   - **all**–Removes all SPAN sessions.
   - **local**–Removes all local sessions.
   - **remote**–Removes all remote SPAN sessions.

3. Use the **monitor session session-number source {interface interface-id} [, | -] [both | rx | tx]** command to specify the source traffic for the monitor session.

   Example

   ```
   Device(config)# monitor session 2 source interface gigabitethernet1/1 rx
   ```

   - For *session_number*, the range is 1 to 2.
   - For *interface-id*, specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel***port-channel-number*). Valid port-channel numbers are 1 to 6.
   - (Optional) [**,** | -]–Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.
   - (Optional) [**both** | **rx** | **tx**]–Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the SPAN monitors both sent and received traffic.
     - **both**–Monitors both sent and received traffic. This is the default.
     - **rx**–Monitors received traffic.
     - **tx**–Monitors sent traffic.

> **Note**
>
> You can use the **monitor session***session_number***source** command multiple times to configure multiple source ports.

4. Use the **monitor session session-number destination {interface interface-id [, | -] [encapsulation replicate] [ingress {vlan vlan-id}]}** command to configure the destination interface for the monitor session.

   Example

   ```
   Device(config)# monitor session 2 destination interface gigabitethernet1/2
   encapsulation replicate ingress vlan 10
   ```

   - For *session_number*, specify the session number used earlier.
   - For **destination**, specify these parameters:
     - For *interface-id*, specify the destination port. The destination interface must be a physical port; it cannot be an EtherChannel, and it cannot be a VLAN.
     - (Optional) [, | -] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.
     - (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged).
     - **ingress** enables forwarding of incoming traffic on the destination port and to specify the encapsulation type:
       - **VLAN***VLAN-id*–Accepts incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN.

> **Note**
>
> For local SPAN, you must use the same session number for the source and destination interfaces.
>
> You can use **monitor session***session_number***destination** command multiple times to configure multiple destination ports.

5. Use the **monitor session session-number filter {ip | ipv6 | mac} access-group {access-list-number | name}** command to apply an access group filter to the monitor session.

   Example

   ```
   Device(config)# monitor session 2 filter ip access-group 7
   ```

   - For *session_number*, specify the session number entered in Step 4.
   - For *access-list-number*, specify the ACL number that you want to use to filter traffic.

- For *name*, specify the ACL name that you want to use to filter traffic.

6. Use the **end** command to return to privileged EXEC mode.

Example

```
Device(config)# end
```

7. (Optional) Use the **show running-config** command to verify your entries.

Example

```
Device# show running-config
```

Configure an FRSPAN session

Configure an FRSPAN session to start an RSPAN source session, specify the monitored source and destination RSPAN VLAN.

An FRSPAN session allows you to monitor network traffic by specifying source ports and destination RSPAN VLANs with filtering capabilities using access control lists.

Use FRSPAN to monitor specific traffic patterns on source ports by filtering packets based on IP, IPv6, or MAC access control lists before sending them to the destination RSPAN VLAN.

Before you begin

To start an RSPAN source session, specify the monitored source and destination RSPAN VLAN, and configure FRSPAN for the session.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

Example

```
Device# configure terminal
```

2. Use the **no monitor session {session-number | all | local | remote}** command to clear the specified monitor session configuration.

Example

```
Device(config)# no monitor session all
```

- For *session_number*, the range is 1 to 2.
- **all**–Removes all SPAN sessions.
- **local**–Removes all local sessions.
- **remote**–Removes all remote SPAN sessions.

3. Use the **monitor session session-number source {interface interface-id} [, | -] [both | rx | tx]** command to specify the source traffic for the monitor session.

   Example

   ```
   Device(config)# monitor session 2 source interface gigabitethernet1/1 rx
   ```

   - For *session_number*, the range is 1 to 2.
   - For *interface-id*, specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel***port-channel-number*). Valid port-channel numbers are 1 to 6.
   - Specify the source VLAN to monitor for *VLAN-id*. The range is 1 to 4094, excluding the RSPAN VLAN.

     Note

     You can include multiple sources (ports or VLANs) in a single session by defining them in a series of commands, but cannot combine source ports and source VLANs in one session.

   - (Optional) [, | -]–Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.
   - (Optional) [**both** | **rx** | **tx**]–Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the SPAN monitors both sent and received traffic.
   - **both**–Monitors both sent and received traffic. This is the default.
   - **rx**–Monitors received traffic.
   - **tx**–Monitors sent traffic.

     Note

     You can use the **monitor session***session_number***source** command multiple times to configure multiple source ports.

4. Use the **monitor session session-number destination remote vlan vlan-id** command to configure the remote VLAN destination for the monitor session.

   Example

   ```
   Device(config)# monitor session 2 destination remote vlan 902
   ```

   - For *session_number*, enter the number defined earlier.
   - For *VLAN-id*, specify the destination RSPAN VLAN to monitor.

5. Use the **vlan vlan-id** command to create or modify the VLAN and enter VLAN configuration mode.

Example

```
Device(config)# vlan 901
```

6. Use the **remote-span** command to configure the VLAN as a remote SPAN VLAN.

   Example

   ```
   Device(config-vlan)# remote-span
   ```

7. Use the **exit** command to return to global configuration mode.

   Example

   ```
   Device(config-vlan)# exit
   ```

8. Use the **monitor session session-number filter {ip | ipv6 | mac} access-group {access-list-number | name}** command to apply an access group filter to the monitor session.

   Example

   ```
   Device(config)# monitor session 2 filter ip access-group 7
   ```

   - For *session_number*, specify the session number entered in Step 4.
   - For *access-list-number*, specify the ACL number that you want to use to filter traffic.
   - For *name*, specify the ACL name that you want to use to filter traffic.

9. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# end
   ```

Monitor SPAN and RSPAN operations

Lists commands for displaying SPAN and RSPAN operations configuration and results to monitor operations.

This reference provides commands used to display SPAN and RSPAN operations configuration and results to monitor operations, as well as global troubleshooting commands for SPAN debugging.

Table 4: Monitoring SPAN and RSPAN operations

Command	Purpose
show monitor session all	Displays the current SPAN, RSPAN, FSPAN, or FRSPAN configuration.

Table 5: Global troubleshooting commands

Command	Purpose
debug all	Displays all SPAN debug messages.
debug capture	Displays capture tracing.
debug errors	Displays SPAN error details.
debug ERSPAN	Displays ERSPAN tracing.
debug IDB-update	Display SPAN interface description block (IDB) update-trace debug messages.
debug info	Display SPAN informational-tracing debug messages.
debug list	Displays SPAN port and VLAN list tracing.
debug notification	Display SPAN notification debug messages.
debug platform	Display SPAN platform-tracing debug messages.
debug requests	Display SPAN request debug messages.
debug SCP	Displays SPAN and SCP tracing.
debug SNMP	Display SPAN and Simple Network Management Protocol (SNMP) tracing debug messages

Configuration examples for SPAN and RSPAN

Describes configuration examples for implementing Switched Port Analyzer (SPAN) and Remote SPAN (RSPAN) features. Provides practical scenarios and command syntax for monitoring network traffic across local and remote switches.

The following sections provide configuration examples for SPAN and RSPAN

Example: configure local SPAN

Provides configuration examples for local SPAN sessions including bidirectional traffic monitoring, removing source ports, and VLAN filtering.

These examples demonstrate various local SPAN configuration scenarios including setting up monitoring sessions, removing existing configurations, configuring source and destination ports, and implementing VLAN filtering for traffic monitoring.

This example shows how to set up SPAN session 1 for monitoring source port traffic to a destination port. First, any existing SPAN configuration for session 1 is deleted, and then bidirectional traffic is mirrored from source Gigabit Ethernet port 1 to destination Gigabit Ethernet port 2, retaining the encapsulation method.

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1
Device(config)# monitor session 1 source interface gigabitethernet1/1
Device(config)# monitor session 1 destination interface gigabitethernet1/2
encapsulation replicate ingress vlan 7
Device(config)# end
```

This example shows how to remove port 1 as a SPAN source for SPAN session 1:

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1 source interface gigabitethernet1/1
Device(config)# end
```

This example shows how to disable received traffic monitoring on port 1, which was configured for bidirectional monitoring:

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1 source interface gigabitethernet1/1 rx
```

The monitoring of traffic received on port 1 is disabled, but traffic sent from this port continues to be monitored.

This example shows how to remove any existing configuration on SPAN session 2, configure SPAN session 2 to monitor received traffic on all ports belonging to VLANs 1 through 3, and send it to destination Gigabit Ethernet port 2. The configuration is then modified to also monitor all traffic on all ports belonging to VLAN 10.

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source vlan 1 - 3 rx

Device(config)# monitor session 2 destination interface gigabitethernet1/2
Device(config)# monitor session 2 source vlan 10
Device(config)# end
```

This example shows how to remove any existing configuration on SPAN session 2, configure SPAN session 2 to monitor received traffic on Gigabit Ethernet source port 1, and send it to destination Gigabit Ethernet port 2 with the same egress encapsulation type as the source port, and to enable ingress forwarding with VLAN 6 as the default ingress VLAN:

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source gigabitethernet0/1 rx
Device(config)# monitor session 2 destination interface gigabitethernet0/2
encapsulation replicate ingress vlan 6
Device(config)# end
```

This example shows how to remove any existing configuration on SPAN session 2, configure SPAN session 2 to monitor traffic received on Gigabit Ethernet trunk port 2, and send traffic for only VLANs 1 through 5 and VLAN 9 to destination Gigabit Ethernet port 1:

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source interface gigabitethernet1/2 rx
Device(config)# monitor session 2 filter vlan 1 - 5 , 9
Device(config)# monitor session 2 destination interface gigabitethernet1/1
ingress vlan 10
Device(config)# end
```

Examples: create an RSPAN VLAN

Provides examples of creating RSPAN VLAN configurations for network monitoring.

This reference provides configuration examples for creating Remote Switched Port Analyzer (RSPAN) VLANs. These examples demonstrate how to create RSPAN VLANs and configure RSPAN sessions for monitoring network traffic across switches.

This example shows how to create the RSPAN VLAN 901:

```
Device> enable
Device# configure terminal
Device(config)# vlan 901
Device(config-vlan)# remote span
Device(config-vlan)# end
```

This example shows how to remove any existing RSPAN configuration for session 1, configure RSPAN session 1 to monitor multiple source interfaces, and configure the destination as RSPAN VLAN 901:

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1
Device(config)# monitor session 1 source interface gigabitethernet1/1 tx
Device(config)# monitor session 1 source interface gigabitethernet1/2 rx
Device(config)# monitor session 1 source interface port-channel 2
Device(config)# monitor session 1 destination remote vlan 901
Device(config)# end
```

This example shows how to remove any existing configuration on RSPAN session 2, configure RSPAN session 2 to monitor traffic received on trunk port 2, and send traffic for only VLANs 1 through 5 and 9 to destination RSPAN VLAN 902:

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source interface gigabitethernet1/2 rx
Device(config)# monitor session 2 filter vlan 1 - 5 , 9
Device(config)# monitor session 2 destination remote vlan 902
Device(config)# end
```

This example shows how to configure VLAN 901 as the source remote VLAN and port 1 as the destination interface:

```
Device> enable
Device# configure terminal
Device(config)# monitor session 1 source remote vlan 901
Device(config)# monitor session 1 destination interface gigabitethernet2/1
ingress vlan 10
Device(config)# end
```

This example shows how to configure VLAN 901 as the source remote VLAN in RSPAN session 2, to configure Gigabit Ethernet source port 2 as the destination interface, and to enable forwarding of incoming traffic on the interface with VLAN 6 as the default receiving VLAN:

```
Device> enable
Device# configure terminal
Device(config)# monitor session 2 source remote vlan 901
```

```
Device(config)# monitor session 2 destination interface gigabitethernet1/2
ingress vlan 6
Device(config)# end
```

3 Configuring IEEE 802.1Q Tunneling

Topics:

IEEE 802.1Q tunneling

Explains IEEE 802.1Q tunneling for service providers carrying multiple customer traffic.

IEEE 802.1Q tunneling is a network feature that

- enables service providers to carry traffic of multiple customers across their networks,
- maintains the VLAN and Layer 2 protocol configurations of each customer, and
- prevents impact on the traffic of other customers.

IEEE 802.1Q tunnel ports

Describes IEEE 802.1Q tunnel ports that enable service providers to support multiple customer VLANs using VLAN-in-VLAN hierarchy.

IEEE 802.1Q tunnel ports are service provider network interfaces that support customers with multiple VLANs using a single service provider VLAN. It preserves customer VLAN IDs while segregating traffic from different customers and expands VLAN space using VLAN-in-VLAN hierarchy and retagging of tagged packets.

Service provider network requirements

Business customers of service providers often have specific requirements for VLAN IDs and the number of VLANs to be supported. The VLAN ranges required by different customers in the same service-provider network might overlap, and traffic of customers through the infrastructure might be mixed."

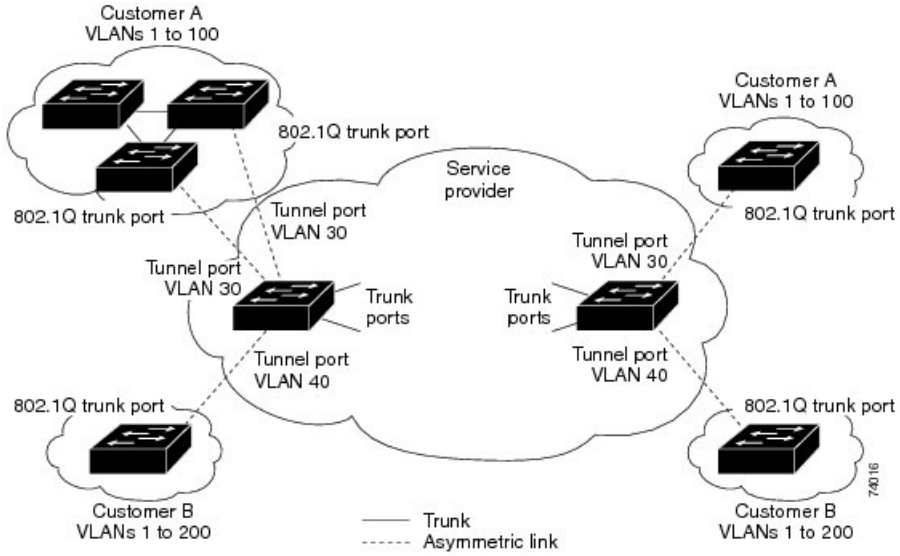
Assigning a unique range of VLAN IDs to each customer would restrict customer configurations and could easily exceed the VLAN limit (4096) of the IEEE 802.1Q specification.

Using the IEEE 802.1Q tunneling feature, service providers can use a single VLAN to support customers who have multiple VLANs. Customer VLAN IDs are preserved, and traffic from different customers is segregated within the service-provider network, even when they appear to be in the same VLAN.

When you configure tunneling, you assign a tunnel port to a VLAN ID that is dedicated to tunneling. Each customer requires a separate service-provider VLAN ID, but that VLAN ID supports all of the customer's VLANs.

Customer traffic tagged in the normal way with appropriate VLAN IDs comes from an IEEE 802.1Q trunk port on the customer device and into a tunnel port on the service-provider edge. The link between the customer device and the edge is asymmetric because one end is configured as an IEEE 802.1Q trunk port, and the other end is configured as a tunnel port. You assign the tunnel port interface to an access VLAN ID that is unique to each customer.

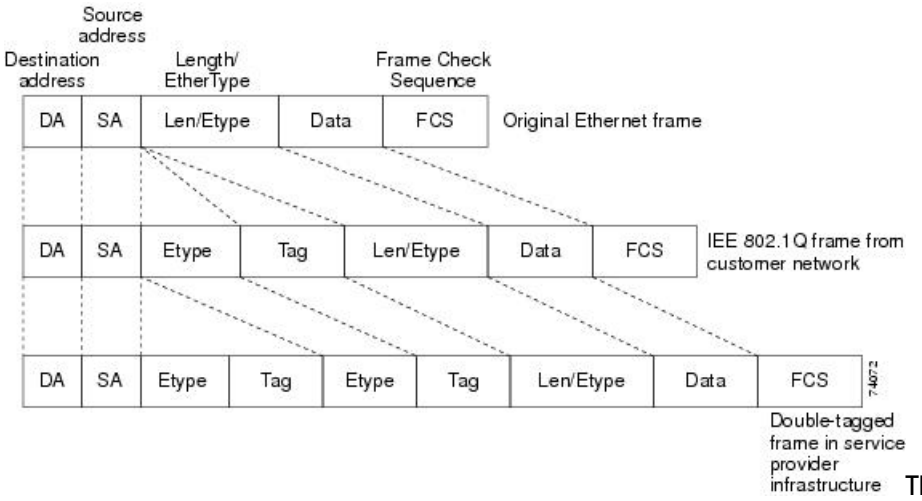
Figure 6: IEEE 802.1Q tunnel ports in a service-provider network



Packets coming from the customer trunk port into the tunnel port on the service-provider edge are normally IEEE 802.1Q-tagged with the appropriate VLAN ID. The tagged packets remain intact inside the network. When they exit the trunk port into the service-provider network, they are encapsulated with an additional IEEE 802.1Q tag (called the metro tag) that contains the VLAN ID unique to the customer. The original customer IEEE 802.1Q tag is preserved in the encapsulated packet. Therefore, packets entering the service-provider network are double-tagged, with the outer (metro) tag containing the customer's access VLAN ID, and the inner VLAN ID being that of the incoming traffic.

When the double-tagged packet enters another trunk port in a service-provider core device, the outer tag is stripped as the device processes the packet. When the packet exits another trunk port on the same core device, the same metro tag is again added to the packet.

Figure 7: Original (normal), IEEE 802.1Q, and double-tagged ethernet packet formats



This figure shows the tag structures of the double-tagged packets.

When the packet enters the trunk port of the service-provider egress device, the outer tag is again stripped as the device internally processes the packet. However, the metro tag is not added when the packet is sent out the tunnel port on the edge device into the customer network. The packet is sent as a normal IEEE 802.1Q-tagged frame to preserve the original VLAN numbers in the customer network.

If traffic coming from a customer network is not tagged (native VLAN frames), these packets are bridged or routed as normal packets. All packets entering the service-provider network through a tunnel port on an edge device are treated as untagged packets, whether they are untagged or already tagged with IEEE 802.1Q headers. The packets are encapsulated with the metro tag VLAN ID (set to the access VLAN of the tunnel port) when they are sent through the

service-provider network on an IEEE 802.1Q trunk port. The priority field on the metro tag is set to the interface class of service (CoS) priority configured on the tunnel port. (The default is zero if none is configured.)

Because 802.1Q tunneling is configured on a per-port basis, it does not matter whether the device is a standalone device or a stack member. All configuration is performed on the stack master.

Customer traffic segregation example

In the network figure, Customer A was assigned VLAN 30, and Customer B was assigned VLAN 40. Packets entering the edge device tunnel ports with IEEE 802.1Q tags are double-tagged when they enter the service-provider network, with the outer tag containing VLAN ID 30 or 40, appropriately, and the inner tag containing the original VLAN number, for example, VLAN 100. Even if both Customers A and B have VLAN 100 in their networks, the traffic remains segregated within the service-provider network because the outer tag is different. Each customer controls its own VLAN numbering space, which is independent of the VLAN numbering space used by other customers and the VLAN numbering space used by the service-provider network.

At the outbound tunnel port, the original VLAN numbers on the customer's network are recovered. It is possible to have multiple levels of tunneling and tagging, but the device supports only one level in this release.

Native VLANs

Explains native VLANs in IEEE 802.1Q tunneling configurations on edge devices.

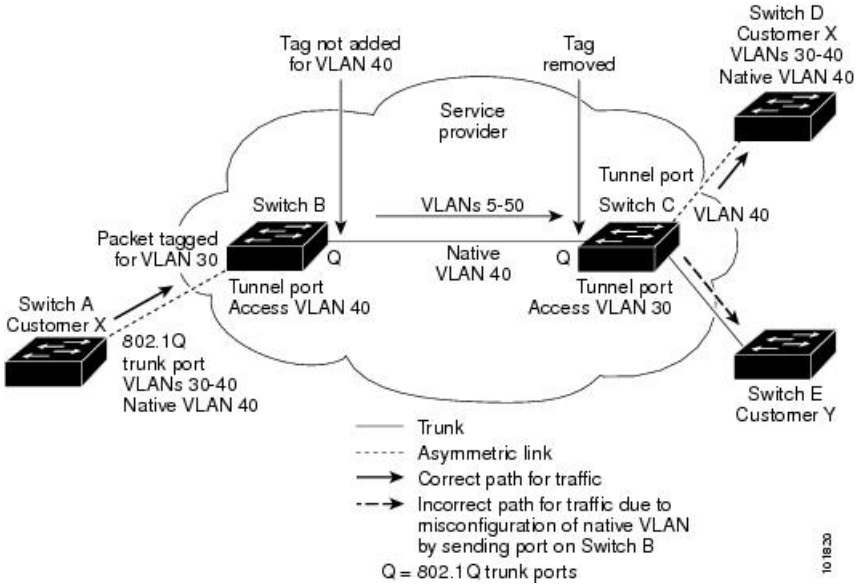
A native VLAN is a VLAN configuration that carries untagged traffic on IEEE 802.1Q trunk ports. Service-provider networks must carefully manage native VLANs to prevent traffic misdirection.

IEEE 802.1Q tunneling configuration requirements

When IEEE 802.1Q trunks are used in these core devices, the native VLANs of the IEEE 802.1Q trunks must not match any native VLAN of the nontrunking (tunneling) port on the same device. If these VLANs match, traffic on the native VLAN will not be tagged on the IEEE 802.1Q sending trunk port, which can cause traffic misdirection.

In this configuration, VLAN 40 is assigned as the native VLAN for the IEEE 802.1Q trunk port from Customer X at the ingress edge device in the service-provider network (device B). Device A of Customer X sends a tagged packet on VLAN 30 to the ingress tunnel port of device B in the service-provider network. This port is configured for access VLAN 40. Because the tunnel port’s access VLAN (VLAN 40) is the same as the native VLAN of the edge device’s trunk port (VLAN 40), the metro tag is not added to tagged packets received from the tunnel port. As a result, the packet carries only the VLAN 30 tag through the service-provider network to the trunk port of the egress edge device (Device C) and is then misdirected through the egress device tunnel port to Customer Y.

Figure 8: Potential problems with IEEE 802.1Q tunneling and native VLANs



These are some ways to solve this problem:

- Use the **VLAN dot1q tag native** global configuration command to configure the edge device so that all packets going out an IEEE 802.1Q trunk, including the native VLAN, are tagged. If the device is configured to tag native VLAN packets on all IEEE 802.1Q trunks, the device accepts untagged packets, but sends only tagged packets.
- Ensure that the native VLAN ID on the edge device trunk port is not within the customer VLAN range. For example, if the trunk port carries traffic for VLANs 100 to 200, assign the native VLAN a number outside that range.

System MTU

Describes the system MTU configuration for traffic on network devices.

A system MTU is a network configuration setting that defines the maximum transmission unit size for traffic on the device. By default, this value is 1500 bytes. You can configure the value on 10-Gigabit and Gigabit Ethernet ports to support frames larger than 1500 bytes using the **system mtu** global configuration command. The calculation does not include the IEEE 802.1Q header.

IEEE 802.1Q tunneling considerations

The IEEE 802.1Q tunneling feature increases the frame size by 4 bytes when the metro tag is added. You must configure all devices in the service-provider network to process maximum frames by adding 4 bytes to the system MTU size.

System MTU configuration example

The device supports a maximum frame size of 1496 bytes with this configuration: The device has a system MTU value of 1500 bytes, and the **switchport mode dot1q tunnel** interface configuration command is configured on a 10-Gigabit or Gigabit Ethernet device port.

Restrictions for IEEE 802.1Q tunneling and other features

Consider the incompatibilities between IEEE 802.1Q tunneling and Layer 3 switching features when implementing Layer 2 packet switching.

Although IEEE 802.1Q tunneling works well for Layer 2 packet switching, there are incompatibilities between some Layer 2 features and Layer 3 switching.

- A tunnel port cannot be a routed port.

- IP routing is not supported on a VLAN with IEEE 802.1Q tunnel ports. Packets from the tunnel port are forwarded based on Layer 2 information only. If you enable routing on a switch virtual interface (SVI) that includes tunnel ports, the switch routes untagged IP packets received from the tunnel port. Customers can access the Internet through their native VLAN. If Internet access is not needed, do not configure SVIs on VLANs that include tunnel ports.
- Fallback bridging is not supported on tunnel ports. All IEEE 802.1Q-tagged packets from a tunnel port are treated as non-IP packets. If you enable fallback bridging on VLANs with tunnel ports, IP packets will be bridged incorrectly across VLANs. Do not enable fallback bridging on VLANs that have tunnel ports.
- Tunnel ports do not support IP access control lists (ACLs).
- Layer 3 quality of service (QoS) ACLs and other QoS features related to Layer 3 information are not supported on tunnel ports. However, MAC-based QoS is supported on tunnel ports.
- EtherChannel port groups are compatible with tunnel ports as long as the IEEE 802.1Q configuration is consistent within an EtherChannel port group.
- Port Aggregation Protocol (PAgP) and Link Aggregation Control Protocol (LACP) are supported on IEEE 802.1Q tunnel ports.

  UniDirectional Link Detection (UDLD) is not supported on IEEE 802.1Q tunnel ports.
- Dynamic Trunking Protocol (DTP) is not compatible with IEEE 802.1Q tunneling because you must manually configure asymmetric links with tunnel ports and trunk ports.
- VLAN Trunking Protocol (VTP) does not work between devices that are connected by an asymmetrical link or devices that communicate through a tunnel.
- When a port is configured as an IEEE 802.1Q tunnel port, spanning-tree bridge protocol data unit (BPDU) filtering is automatically enabled on the interface, while Cisco Discovery Protocol (CDP) and the Link Layer Discovery Protocol (LLDP) are automatically disabled.
- When an IEEE 802.1Q tunnel port is configured as SPAN source, SPAN filter must be applied for SVLAN to avoid packet loss.
- IGMP/MLD packet forwarding can be enabled on IEEE 802.1Q tunnels. This can be done by disabling IGMP/MLD snooping on the service provider network.

Default IEEE 802.1Q tunneling configuration

Lists the default IEEE 802.1Q tunneling configuration settings.

By default, IEEE 802.1Q tunneling is disabled because the default switchport mode is dynamic auto. Tagging of IEEE 802.1Q native VLAN packets on all IEEE 802.1Q trunk ports is also disabled.

Configure IEEE 802.1Q tunneling

Configure an IEEE 802.1Q tunneling port on an edge device.

Configure an IEEE 802.1Q tunneling port to enable service provider networks to carry customer VLAN traffic transparently across the provider network infrastructure.

IEEE 802.1Q tunneling allows service providers to transport customer VLAN traffic through their network while maintaining VLAN tag separation between different customers.

Before you begin

- Always use an asymmetrical link between the customer device and the edge device, with the customer device port configured as an IEEE 802.1Q trunk port and the edge device port configured as a tunnel port.
- Assign tunnel ports only to VLANs that are used for tunneling.

- Observe configuration requirements for native VLANs and maximum transmission units (MTUs).

Follow these steps to configure a port as an IEEE 802.1Q tunnel port:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **interface** *interface-id* command to enter interface configuration mode for the tunnel port.

   Example

   ```
   Device(config)# interface gigabitethernet2/0/1
   ```

   This interface should be the edge port in the service-provider network that connects to the customer device. Valid interfaces include physical interfaces and port-channel logical interfaces. Valid port-channel numbers are 1 to 48.

3. Use the **switchport access vlan** *vlan-id* command to specify the default VLAN to use if the interface stops trunking.

   Example

   ```
   Device(config-if)# switchport access vlan 2
   ```

   This VLAN ID is specific to the particular customer.

4. Use the **switchport mode dot1q-tunnel** command to set the interface as an IEEE 802.1Q tunnel port.

   Example

   ```
   Device(config-if)# switchport mode dot1q-tunnel
   ```

   Note

   Use the **no switchport mode dot1q-tunnel** interface configuration command to return the port to the default state of dynamic desirable.

5. Use the **exit** command to return to global configuration mode.

   Example

   ```
   Device(config-if)# exit
   ```

6. (Optional) Use the **vlan dot1q tag native** command to enable tagging of native VLAN packets on all IEEE 802.1Q trunk ports.

Example

```
Device(config)# vlan dot1q tag native
```

If this command is not configured and a customer VLAN ID matches the native VLAN, the trunk port does not apply a metro tag. As a result, packets could be sent to the wrong destination.

> **Note**
>
> Use the **no vlan dot1q tag native** global configuration command to disable tagging of native VLAN packets.

7. Use the **end** command to return to privileged EXEC mode.

Example

```
Device(config)# end
```

8. (Optional) Use the **show dot1q-tunnel** or **show running-config interface** command to verify the IEEE 802.1Q tunneling port configuration.

Example

```
Device# show dot1q-tunnel
```

or

```
Device# show running-config interface
```

9. (Optional) Use the **show vlan dot1q tag native** command to display IEEE 802.1Q native VLAN tagging status.

Example

```
Device# show vlan dot1q tag native
```

Monitor tunneling status

Lists commands used to monitor tunneling status on network interfaces.

This table describes the commands used to monitor tunneling status.

Table 6: Commands for monitoring tunneling

Command	Purpose
show dot1q-tunnel	Displays IEEE 802.1Q tunnel ports on the device.
show dot1q-tunnel interface *interface-id*	Verifies if a specific interface is a tunnel port.
show VLAN dot1q tag native	Displays the status of native VLAN tagging on the device.

Example: configure an IEEE 802.1Q tunneling port

Provides an example of configuring an interface as a tunnel port, enabling tagging of native VLAN packets, and verifying the configuration.

This example shows how to configure an interface as a tunnel port, enable tagging of native VLAN packets, and verify the configuration. In this configuration, the VLAN ID for the customer connected to Gigabit Ethernet interface 7 on stack member 1 is VLAN 22.

```
Switch(config)# interface gigabitethernet1/0/7
Switch(config-if)# switchport access vlan 22
% Access VLAN does not exist. Creating vlan 22
Switch(config-if)# switchport mode dot1q-tunnel
Switch(config-if)# exit
Switch(config)# vlan dot1q tag native
Switch(config)# end
Switch# show dot1q-tunnel interface gigabitethernet1/0/7
Port
-----
Gi1/0/1Port
-----
Switch# show vlan dot1q tag native
dot1q native vlan tagging is enabled
```

4 Configuring VLAN Mapping

Topics:

VLAN mapping

Describes a service that allows service providers to establish translated VLAN IDs by mapping customer VLANs to service-provider VLANs on trunk ports.

VLAN mapping is a service that

- maps customer VLANs to service-provider VLANs (called VLAN ID translation) on trunk ports connected to a customer network,
- allows service providers to provide a transparent switching infrastructure that includes customers' switches at the remote location as a part of the local site, and
- enables customers to use the same VLAN ID space and run Layer 2 control protocols seamlessly across the provider network.

Mapping operations and features

In a typical deployment of VLAN mapping, the service provider should provide a transparent switching infrastructure that includes customers' switches at the remote location as part of the local site. This setup enables customers to use the same VLAN ID space and run Layer 2 control protocols seamlessly across the provider network.

One way to establish translated VLAN IDs (S-VLANs) is to map customer VLANs to service-provider VLANs (called VLAN ID translation) on trunk ports connected to a customer network. Packets entering the port are mapped to a service provider VLAN (S-VLAN) based on the port number and the packet's original customer VLAN-ID (C-VLAN).

Service providers' internal assignments might conflict with a customer's VLAN. To isolate customer traffic, a service provider could choose to map a specific VLAN to another VLAN during the traffic's transit through its cloud.

All forwarding operations on the switch are performed using S-VLAN and not C-VLAN information because the VLAN ID is mapped to the S-VLAN on ingress.

Note

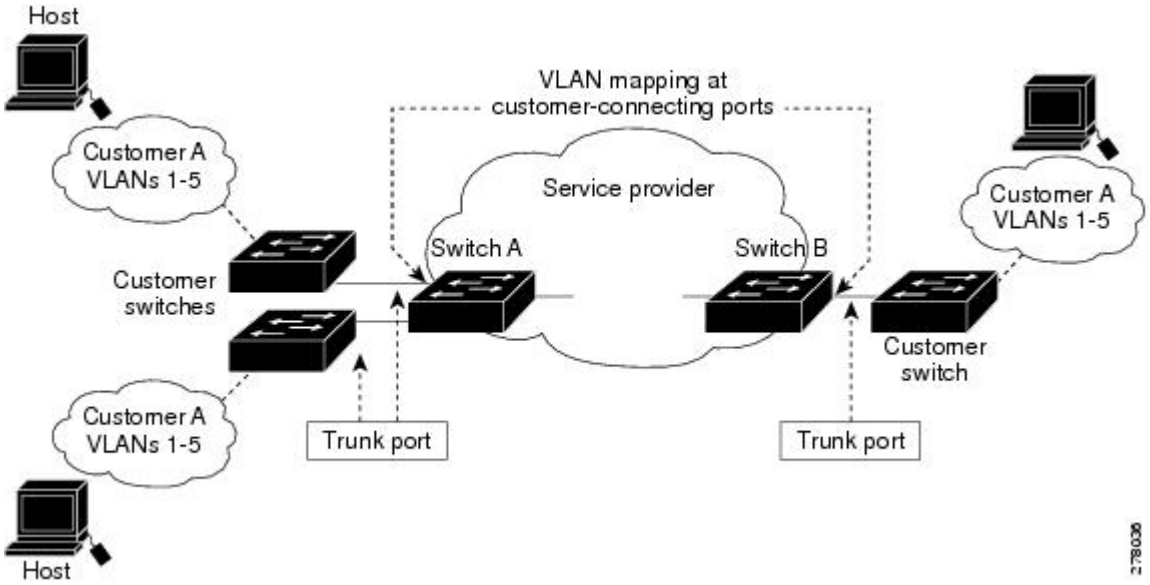
When you configure features on a port configured for VLAN mapping, you always use the S-VLAN rather than the customer VLAN-ID (C-VLAN). One-to-one VLAN mapping is not supported at this time.

On an interface configured for VLAN mapping, the specified C-VLAN packets are mapped to the specified S-VLAN when they enter the port. When packets exit the port, the switch applies symmetrical mapping to the customer C-VLAN.

The switch supports these types of VLAN mapping on trunk ports:

- Selective QinQ.
- QinQ on a trunk port.

Mapping customer VLANs to service-provider VLANs



The figure here shows a topology where a customer uses the same VLANs in multiple sites on different sides of a service-provider network. You map the customer VLAN IDs to service-provider VLAN IDs for packet travel across the service-provider backbone. The customer VLAN IDs are retrieved at the other side of the service-provider backbone for use in the other customer site. Configure the same set of VLAN mappings at a customer-connected port on each side of the service-provider network.

Selective Q-in-Q

Describes a VLAN mapping technique that maps specified customer VLANs to designated S-VLAN IDs for service provider network transport.

Selective Q-in-Q is a VLAN mapping technique that

- maps specified customer VLANs entering the UNI to the specified S-VLAN ID,
- adds the S-VLAN ID to the incoming unmodified C-VLAN so the packet travels the service provider network double-tagged, and
- removes the S-VLAN ID at egress while retaining the customer VLAN-ID on the packet.

Selective Q-in-Q behavior

By default, packets that do not match the specified customer VLANs are dropped.

Q-in-Q on a trunk port

Describes a QinQ configuration that maps all customer VLANs entering the UNI to a specified S-VLAN ID on trunk ports.

Q-in-Q on a trunk port is a VLAN tunneling configuration. This configuration maps all customer VLANs entering the UNI to a specified S-VLAN ID, creates double-tagged packets as in Selective QinQ, and removes the S-VLAN ID at egress.

Configuration guidelines for VLAN mapping

Outlines configuration guidelines and restrictions for VLAN mapping implementation.

Follow these configuration guidelines when implementing VLAN mapping on your network devices.

- By default, no VLAN mapping is configured.

EtherChannel configuration guidelines

When configuring VLAN mapping with EtherChannel interfaces, consider these requirements to maintain proper bundle operation.

- If the VLAN mapping is enabled on an EtherChannel, the configuration does not apply to all member ports of the EtherChannel bundle and applies only to the EtherChannel interface.
- If the VLAN mapping is enabled on an EtherChannel and a conflicting mapping/translation is enabled on a member port, then the port is removed from the EtherChannel.
- If a port belonging to an EtherChannel is configured with a VLAN mapping and the EtherChannel is configured with a conflicting VLAN mapping, then the port is removed from the EtherChannel.
- The member port of an EtherChannel is removed from the EtherChannel bundle if the mode of the port is changed to anything other than 'trunk' mode.

Control traffic processing guidelines

To process control traffic consistently, either enable Layer 2 protocol tunneling (recommended):

```
!
Device(config)# interface Gig 1/1
Device(config-if)# switchport mode access
Device(config-if)# l2protocol-tunnel stp
Device(config-if)# end
```

Or insert a BPDU filter for spanning tree:

```
Current configuration : 153 bytes
!
Device(config)# interface Gig 1/1
Device(config-if)# switchport mode trunk
Device(config-if)# switchport vlan mapping 10 20
Device(config-if)# spanning-tree bpdufilter enable
Device(config-if)# end
```

VLAN restrictions

Be aware of VLAN limitations when configuring VLAN mapping.

- Default native VLANs, user-configured native VLANs, and reserved VLANs (range 1002-1005) cannot be used for VLAN mapping.
- PVLAN support is not available when VLAN mapping is configured.

Guidelines for selective Q-in-Q configuration

Outlines key requirements and limitations when configuring Selective Q-in-Q on network devices.

Follow these guidelines when configuring Selective Q-in-Q:

- S-VLAN should be created and present in the allowed VLAN list of the trunk port where Selective Q-in-Q is configured.
- When Selective Q-in-Q is configured, the device supports Layer 2 protocol tunneling for CDP, STP, LLDP, and VTP.
- IP routing is not supported on Selective Q-in-Q enabled ports.
- IPSG is not supported on Selective Q-in-Q enabled ports.

Guidelines for Q-in-Q configuration on trunk ports

Outlines the configuration requirements and supported features for Q-in-Q implementation on trunk ports.

Follow these guidelines when configuring Q-in-Q on trunk ports:

- Ensure that the S-VLAN is created and included in the allowed VLAN list of the trunk port where Q-in-Q is configured..
- When Q-in-Q on a trunk port is configured, the device supports Layer 2 protocol tunneling for CDP, STP, LLDP, and VTP.
- Ingress SPAN, egress SPAN, and RSPAN are supported on trunk ports with Q-in-Q enabled.
- When Q-in-Q is enabled, the SPAN filtering can be enabled to monitor only the traffic on the mapped VLAN, i.e. S-VLANs.
- IGMP snooping is not supported on the C-VLAN.

How to configure VLAN mapping

Describes how to configure VLAN mapping on Cisco devices. Provides step-by-step procedures and configuration examples for mapping VLANs between different network segments or interfaces.

The following sections provide information about configuring VLAN mapping:

Configure selective Q-in-Q on a trunk port

Configure VLAN mapping for selective Q-in-Q on a trunk port to enable customer VLAN traffic encapsulation within service provider VLANs.

This task configures VLAN mapping for selective Q-in-Q on a trunk port, allowing specific customer VLANs to be mapped to service provider VLANs while providing options for handling unmapped traffic.

Use selective Q-in-Q configuration to map specific customer VLANs to service provider VLANs on trunk interfaces that connect to service provider networks.

> **Note**
>
> You cannot configure one-to-one mapping and selective Q-in-Q on the same interface.

Before you begin

Follow these steps to configure VLAN mapping for selective Q-in-Q on a trunk port:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **interface** *interface-id* command to enter interface configuration mode for the interface connected to the service-provider network.

Example

```
Device(config)# interface gigabitethernet1/1
```

Enter a physical interface or an EtherChannel port channel.

3. Use the **switchport mode trunk** command to configure the interface as a trunk port.

   Example

   ```
   Device(config-if)# switchport mode trunk
   ```

4. Use the **switchport vlan mapping** *vlan-id* **dot1q-tunnel** *outer-vlan-id* command to map the customer VLAN ID to the service-provider VLAN ID.

   Example

   ```
   Device(config-if)# switchport vlan mapping 16 dot1q-tunnel 64
   ```

   - vlan-id is the customer VLAN ID (C-VLAN) entering the device from the customer network. The range is 1 to 4094. You can enter a string of VLAN IDs.
   - outer-vlan-id is the outer VLAN ID (S-VLAN) of the service provider network. The range is 1 to 4094.

   Use the **no** form of this command to remove the VLAN mapping configuration. Entering the **no switchport vlan mapping all** command deletes all mapping configurations.

5. (Optional) Use the **switchport vlan mapping default dot1q-tunnel** *vlan-id* command to forward all unmapped packets on the port with the specified S-VLAN.

   Example

   ```
   Device(config-if)# switchport vlan mapping default dot1q-tunnel 22
   ```

   By default, packets that do not match the mapped VLANs are dropped. The device forwards untagged traffic.

6. Use the **exit** command to return to global configuration mode.

   Example

   ```
   Device(config-if)# exit
   ```

7. Use the **spanning-tree bpdufilter enable** command to insert a BPDU filter for spanning tree.

   Example

   ```
   Device(config)# spanning-tree bpdufilter enable
   ```

> **Note**
>
> To process control traffic consistently, either enable Layer 2 protocol tunneling (recommended) or insert a BPDU filter for spanning tree.

8. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# end
   ```

9. (Optional) Use the **show interfaces** *interface-id* **vlan mapping** command to verify the VLAN mapping configuration.

   Example

   ```
   Device# show interfaces gigabitethernet1/1 vlan mapping
   ```

Configure Q-in-Q on a trunk port

Configure VLAN mapping for Q-in-Q on a trunk port so that customer VLAN traffic is forwarded with a service provider VLAN.

Configure VLAN mapping for Q-in-Q on a trunk port to forward all unmapped C-VLAN packets with a specified S-VLAN.

Use this procedure to configure VLAN mapping for Q-in-Q on a trunk port connected to the service-provider network.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **interface** *interface-id* command to enter interface configuration mode for the network interface connected to the service provider.

   Example

   ```
   Device(config)# interface gigabitethernet1/1
   ```

   You can enter a physical interface or an EtherChannel port channel.

3. Use the **switchport mode trunk** command to configure the interface as a trunk port.

   Example

   ```
   Device(config-if)# switchport mode trunk
   ```

4. Use the **switchport vlan mapping default dot1q-tunnel** *vlan-id* command to forward all unmapped C-VLAN packets on the port with the specified S-VLAN.

   Example

   ```
   Device(config-if)# switchport vlan mapping default dot1q-tunnel 16
   ```

5. Use the **exit** command to return to global configuration mode.

   Example

   ```
   Device(config-if)# exit
   ```

6. Use the **spanning-tree bpdufilter enable** command to insert a BPDU filter for spanning tree.

   Example

   ```
   Device(config)# spanning-tree bpdufilter enable
   ```

   Note

   To process control traffic consistently, either enable Layer 2 protocol tunneling (recommended) or insert a BPDU filter for spanning tree.

7. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# end
   ```

8. (Optional) Use the **show interfaces** *interface-id* **vlan mapping** command to verify the VLAN mapping configuration.

   Example

   ```
   Device# show interfaces gigabitethernet1/1 vlan mapping
   ```

9. (Optional) Use the **copy running-config startup-config** command to save your entries in the configuration file.

   Example

   ```
   Device# copy running-config startup-config
   ```

This example shows how to configure Q-in-Q mapping on the port so that traffic of any VLAN ID is forwarded with the S-VLAN ID of 200.

```
Device(config)# interface gigabitethernet0/1
Device(config-if)# switchport vlan mapping default dot1q-tunnel 200
Device(config-if)# exit
```

5 Loop Detection Guard

Topics:

Loop Detection Guard

Explains how Loop Detection Guard detects Layer 2 loops and helps maintain network stability.

A computer network can experience a network loop where there is more than one Layer 2 path between two endpoints. This is possible when there are multiple connections between two switches in a network or two ports on the same switch are connected to each other.

Need for Loop Detection Guard

While Spanning Tree Protocol (STP) is typically used to prevent network loops, loop detection guard is essential in environments with unmanaged switches that do not support STP or where STP is not configured. In such scenarios—especially when edge switches are connected to an unmanaged switch and STP is disabled, a dedicated loop detection mechanism is essential to ensure network stability.

Loop detection mechanism

To detect loops, the system sends loop-detect frames from the interface, at preconfigured intervals. When a loop is detected, the configured action is taken. Loop detection guard is enabled at the interface level.

Supported devices

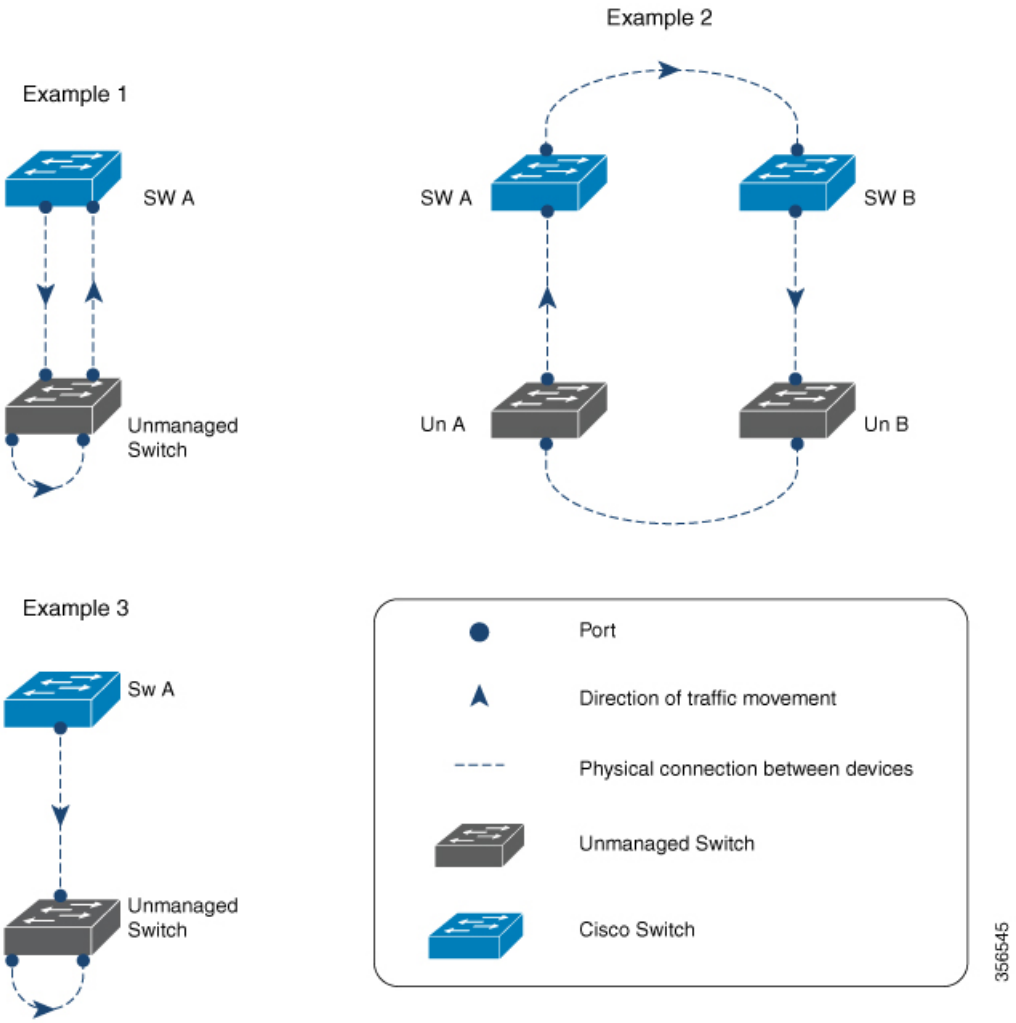
This feature is available with Network Essential license and is supported on these switches:

- Cisco Catalyst IE3100 Series Switches
- Cisco Catalyst IE3200 Series Switches
- Cisco Catalyst IE3300 Series Switches
- Cisco Catalyst IE3400 Series Switches

Types of loops

The figure given here illustrates how loops occur.

Figure 9: Types of loops



- **Example 1**: A managed switch (SW A) is sending traffic to an unmanaged switch on one port and receiving traffic from the same unmanaged switch, on another port. On the unmanaged switch, the port receiving traffic is connected to the port sending the traffic back to the network. This creates a loop between the managed and unmanaged switches.
- **Example 2**: Two managed switches (SW A and SW B) are sending traffic to two unmanaged switches (Un A and Un B). Traffic is moving from SW A to SW B to Un A to Un B and back to SW A, resulting in a network loop.
- **Example 3**: A cable connects two ports on the unmanaged switch, resulting in a network loop.

Enable Loop Detection Guard using CLI

Guides how to enable Loop Detection Guard on an interface using CLI commands.

This task allows you to enable or disable the feature on a specific interface and configure the interval at which loop-detect frames are sent.

Before you begin

This feature is disabled by default and can be enabled in interface configuration mode for physical interfaces only.

Procedure

1. Use the **configure terminal** command to enter the global configuration mode.

Example

```
Switch# configure terminal
```

2. Use the **interface** *interface-name* command to enter the interface configuration mode.

   Example

   ```
   Switch#(config)# interface Gigabitethernet 1/6
   ```

   Specify only a physical interface to configure loop detection guard on the device.

3. Use the **loopdetect** command to enable loop detection guard.

   Example

   ```
   Switch#(config-if)# loopdetect
   ```

   Use the **no** form of the command to disable loop detection guard.

4. Use the **loopdetect** *time* command to speccify the frequency at which loop-detect frames are sent.

   Example

   ```
   Switch#(config)# loopdetect 7
   ```

   The *time* is the interval to send loop-detect frame, in seconds. The range is from 1 to 10. The default is 5.

5. Use the **loopdetect action syslog** command to specify the action to be performed after detecting a loop.

   Example

   ```
   Switch#(config)# loopdetect action syslog
   ```

   In this instance, a Syslog is displayed after detecting a loop. If you do not specify an action, the destination port is error-disabled by default. Use the **no** form of the command to disable the action.

6. Use the **loopdetect sourceport** command to error-disable the source port.

   Example

   ```
   Switch#(config)# loopdetect sourceport
   ```

   Use the **no** form of the command to disable the option.

Verify Loop Detect

Shows how to verify Loop Detection Guard status and confirm loop detection behavior.

To monitor and troubleshoot loop detection behavior on a switch, use these commands.

Procedure

1. (Optional) Use the **show loopdetect** command to see all the interfaces where loop detection guard is enabled, the frequency at which loop-detect packets are sent, and the status of the physical interface.

   Example

```
Switch# show loopdetect

Loopdetect is enabled

Interface      Interval Elapsed-Time Port-to-Errdisbale     ACTION
-------------- -------- ------------ ---------------------  ---------
Gi1/5           7        2            errdisable Dest Port   ERRDISABLE
Gi1/6           7        2            errdisable Dest Port   ERRDISABLE
```

2. (Optional) Use the **show interfaces status err-disabled** command to see all the interfaces that are currently in an error-disabled (err-disabled) state.

   Example

```
Switch# show interfaces status err-disabled

 Port          Name          Status       Reason               Err-disabled
Vlans

Gi1/5                      err-disabled loopdetect
```

   To recover an interface from the err-disabled state, you can run the **shutdown** and **no shutdown** commands under the interface configuration mode, or enable recovery using the **errdisable recovery cause loopdetect** command

3. (Optional) Use the **show running-config interface** *interface-name* | **i loop** command to see where the loop detect is enabled on the interface.

   Example

```
Switch# show running-config interface Gi1/6 | i loop

loopdetect
loopdetect 7
loopdetect source-port
```

Feature History for Loop Detection Guard

Lists release support and history for the Loop Detection Guard feature.

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE 17.18.1	Loop Detection Guard	Loop detection guard prevents loops in networks without STP or with unmanaged devices.