Configuring IEEE 802.1Q and Layer 2 Protocol Tunneling

Prerequisites for Configuring Tunneling

The following sections list prerequisites and considerations for configuring IEEE 802.1Q and Layer 2 protocol tunneling.

IEEE 802.1Q Tunneling

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 that includes IEEE 802.1Q tunnel ports. Packets received from a tunnel port are forwarded based only on Layer 2 information. If routing is enabled on a device virtual interface (SVI) that includes tunnel ports, untagged IP packets received from the tunnel port are recognized and routed by the device. Customers can access the Internet through its native VLAN. If this access is not needed, you should not configure SVIs on VLANs that include tunnel ports.
- Fallback bridging is not supported on tunnel ports. Because all IEEE 802.1Q-tagged packets received from a tunnel port are treated as non-IP packets, if fallback bridging is enabled on VLANs that have tunnel ports configured, IP packets would be improperly bridged across VLANs. Therefore, you must not enable fallback bridging on VLANs with 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. 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), Link Aggregation Control Protocol (LACP), and UniDirectional Link Detection (UDLD) are 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.
- Loopback detection is supported on IEEE 802.1Q tunnel ports.
- 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. Cisco Discovery Protocol (CDP) and the Layer Link Discovery Protocol (LLDP) are automatically disabled on the interface.

Information about Tunneling

IEEE 802.1Q and Layer 2 Protocol Overview

Virtual private networks (VPNs) provide enterprise-scale connectivity on a shared infrastructure, often Ethernet-based, with the same security, prioritization, reliability, and manageability requirements of private networks. Tunneling is a feature designed for service providers who carry traffic of multiple customers across their networks and are required to maintain the VLAN and Layer 2 protocol configurations of each customer without impacting the traffic of other customers.

For complete syntax and usage information for the commands used in this chapter, see the command reference for this release.

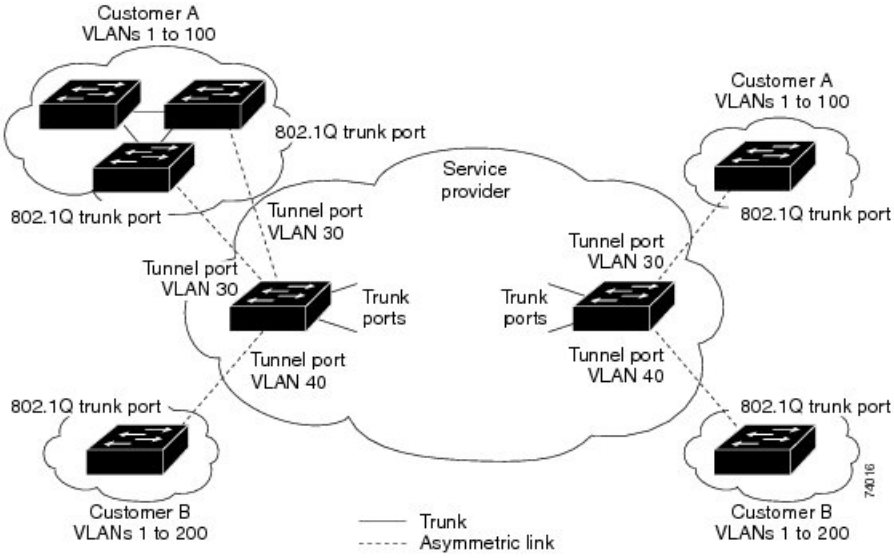
IEEE 802.1Q Tunneling

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. Using IEEE 802.1Q tunneling expands VLAN space by using a VLAN-in-VLAN hierarchy and retagging the tagged packets. A port configured to support IEEE 802.1Q tunneling is called a tunnel port. 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 device. The link between the customer device and the edge device 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 1: IEEE 802.1Q Tunnel Ports in a Service-Provider Network

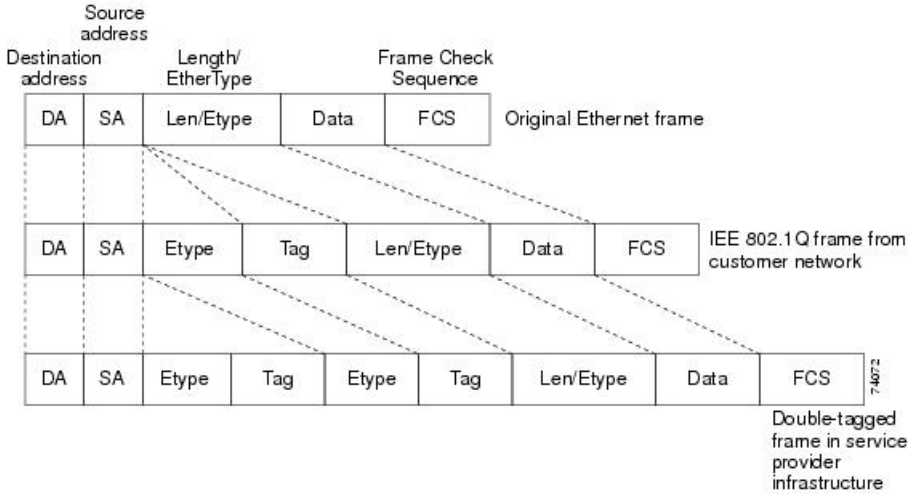


Packets coming from the customer trunk port into the tunnel port on the service-provider edge device are normally IEEE 802.1Q-tagged with the appropriate VLAN ID. The tagged packets remain intact inside the device and when they exit the trunk port into the service-provider network, they are encapsulated with another layer of an IEEE 802.1Q tag (called the metro tag) that contains the VLAN ID that is 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 2: 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.

In the above 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.

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.)

On device, 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 done on the active stack.

IEEE 802.1Q Tunneling Configuration Guidelines

When you configure IEEE 802.1Q tunneling, you should 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.

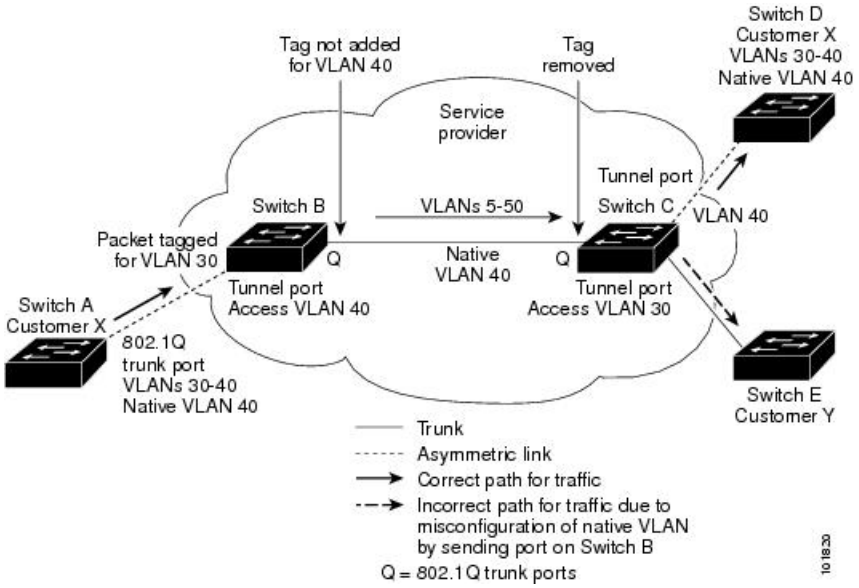
Configuration requirements for native VLANs and for and maximum transmission units (MTUs) are explained in these next sections.

Native VLANs

When configuring IEEE 802.1Q tunneling on an edge device, you must use IEEE 802.1Q trunk ports for sending packets into the service-provider network. However, packets going through the core of the service-provider network can be carried through IEEE 802.1Q trunks, ISL trunks, or nontrunking links. 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 because traffic on the native VLAN would not be tagged on the IEEE 802.1Q sending trunk port.

In the following network figure, VLAN 40 is configured 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, which belongs to access VLAN 40. Because the access VLAN of the tunnel port (VLAN 40) is the same as the native VLAN of the edge device trunk port (VLAN 40), the metro tag is not added to tagged packets received from the tunnel port. The packet carries only the VLAN 30 tag through the service-provider network to the trunk port of the egress-edgedevice (Device C) and is misdirected through the egress device tunnel port to Customer Y.

Figure 3: 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 devices so that all packets going out an IEEE 802.1Q trunk, including the native VLAN, are tagged. If the devices is configured to tag native VLAN packets on all IEEE 802.1Q trunks, the devices accepts untagged packets, but sends only tagged packets.

- Ensure that the native VLAN ID on the edge devices trunk port is not within the customer VLAN range. For example, if the trunk port carries traffic of VLANs 100 to 200, assign the native VLAN a number outside that range.

System MTU

The default system MTU for traffic on the device is 1500 bytes. You can configure Fast Ethernet ports on the device members in the mixed hardware device stack to support frames larger than 1500 bytes by using the **system mtu** global configuration command.

You can configure 10-Gigabit and Gigabit Ethernet ports to support frames larger than 1500 bytes by using the **system mtu jumbo** global configuration command.

The system MTU and system jumbo MTU values do not include the IEEE 802.1Q header. Because 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 be able to process maximum frames by adding 4 bytes to the system MTU and system jumbo MTU sizes.

For example, the device supports a maximum frame size of 1496 bytes with one of these configurations:

- The device has a system jumbo 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.
- The device member has a system MTU value of 1500 bytes, and the **switchport mode dot1q tunnel** interface configuration command is configured on a Fast Ethernet port of the member.

Default IEEE 802.1Q Tunneling Configuration

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.

How to Configure Tunneling

Configuring an IEEE 802.1Q Tunneling Port

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet2/0/1`	Enters interface configuration mode for the interface to be configured as a tunnel port. This 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 (port channels 1 to 48).
Step 4	**switchport access vlan** *vlan-id* **Example:** `Device(config-if)# switchport access vlan 2`	Specifies the default VLAN, which is used if the interface stops trunking. This VLAN ID is specific to the particular customer.
Step 5	**switchport mode dot1q-tunnel** **Example:** `Device(config-if)# switchport mode dot1q-tunnel`	Sets the interface as an IEEE 802.1Q tunnel port. **Note** Use the **no switchport mode dot1q-tunnel** interface configuration command to return the port to the default state of dynamic desirable.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Returns to privileged EXEC mode.
Step 7	**vlan dot1q tag native** **Example:** `Device(config)# vlan dot1q tag native`	(Optional) Sets the device to enable tagging of native VLAN packets on all IEEE 802.1Q trunk ports. When not set, and a customer VLAN ID is the same as the native VLAN, the trunk port does not apply a metro tag, and 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.
Step 8	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

	Command or Action	Purpose

Step 9	Use one of the following:
	• **show dot1q-tunnel**
	• **show running-config interface**
	Example:
	`Device# show dot1q-tunnel`
	or
	`Device# show running-config interface`
		Displays the ports configured for IEEE 802.1Q tunneling.
		Displays the ports that are in tunnel mode.

Step 10	**show vlan dot1q tag native** **Example:** `Device# show vlan dot1q native`	Displays IEEE 802.1Q native VLAN tagging status.
Step 11	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring the SP Edge Switch

Before you begin

For EtherChannels, you need to configure both the SP (service-provider) edge devices and the customer devices for Layer 2 protocol tunneling.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet1/0/1`	Specifies the interface connected to the phone, and enters interface configuration mode.
Step 4	**switchport mode dot1q-tunnel** **Example:** `Device(config-if)# switchport mode dot1q-tunnel`	Configures the interface as an IEEE 802.1Q tunnel port.
Step 5	**l2protocol-tunnel point-to-point** [**pagp** | **lacp** | **udld**] **Example:** `Device(config-if)# l2protocol-tunnel point-to-point pagp`	(Optional) Enables point-to-point protocol tunneling for the desired protocol. 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** [**pagp** | **lacp** | **udld**]] interface configuration command to disable point-to-point protocol tunneling for one of the Layer 2 protocols or for all three.
Step 6	**l2protocol-tunnel shutdown-threshold** [**point-to-point** [**pagp** | **lacp** | **udld**]] *value* **Example:** `Device(config-if)# l2protocol-tunnel shutdown-threshold point-to-point pagp 100`	(Optional) Configures the threshold for packets-per-second accepted for encapsulation. The interface is disabled if the configured threshold is exceeded. If no protocol option is specified, the threshold applies to each of the tunneled Layer 2 protocol types. 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.

	Command or Action	Purpose
		Note Use the **no l2protocol-tunnel shutdown-threshold** [**point-to-point** [**pagp** | **lacp** | **udld**]] and the **no l2protocol-tunnel drop-threshold** [[**point-to-point** [**pagp** | lacp | **udld**]] commands to return the shutdown and drop thresholds to the default settings.
Step 7	**l2protocol-tunnel drop-threshold** [**point-to-point** [**pagp** | **lacp** | **udld**]] *value* **Example:** `Device(config-if)# l2protocol-tunnel drop-threshold point-to-point pagp 500`	(Optional) Configures the threshold for packets-per-second accepted for encapsulation. The interface drops packets if the configured threshold is exceeded. If no protocol option is specified, the threshold applies to each of the tunneled Layer 2 protocol types. 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.
Step 8	**no cdp enable** **Example:** `Device(config-if)# no cdp enable`	Disables CDP on the interface.
Step 9	**spanning-tree bpdu filter enable** **Example:** `Device(config-if)# spanning-tree bpdu filter enable`	Enables BPDU filtering on the interface.
Step 10	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 11	**errdisable recovery cause l2ptguard** **Example:** `Device(config)# errdisable recovery`	(Optional) Configures the recovery mechanism from a Layer 2 maximum-rate error so that the interface is reenabled and can try again. Errdisable recovery is disabled by default; when enabled, the default time interval is 300 seconds.

	Command or Action	Purpose
	`cause l2ptguard`	
Step 12	**l2protocol-tunnel cos** *value* **Example:** `Device(config)# l2protocol-tunnel cos 2`	(Optional) Configures the CoS value for all tunneled Layer 2 PDUs. The range is 0 to 7; the default is the default CoS value for the interface. If none is configured, the default is 5.
Step 13	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 14	**show l2protocol** **Example:** `Device)# show l2protocol`	Displays the Layer 2 tunnel ports on the device, including the protocols configured, the thresholds, and the counters.
Step 15	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring the Customer Device

Before you begin

For EtherChannels, you need to configure both the SP edge device and the customer devices for Layer 2 protocol tunneling.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet1/0/1`	Specifies the interface connected to the phone, and enters interface configuration mode.
Step 4	**switchport trunk encapsulation dot1q** **Example:** `Device(config)# switchport trunk encapsulation dot1q`	Sets the trunking encapsulation format to IEEE 802.1Q.
Step 5	**switchport mode trunk** **Example:** `Device(config-if)# switchport mode trunk`	Enables trunking on the interface.
Step 6	**udld port** **Example:** `Device(config-if)# udld port`	Enables UDLD in normal mode on the interface.
Step 7	**channel-group** *channel-group-number* **mode desirable** **Example:** `Device(config-if)# channel-group 25 mode desirable`	Assigns the interface to a channel group, and specifies desirable for the PAgP mode.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 9	**interface port-channel port-channel** *number* **Example:** `Device(config)# interface port-channel port-channel 25`	Enters port-channel interface mode.

	Command or Action	Purpose
Step 10	**shutdown** **Example:** `Device(config)# shutdown`	Shuts down the interface.
Step 11	**no shutdown** **Example:** `Device(config)# no shutdown`	Enables the interface.
Step 12	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 13	**show l2protocol** **Example:** `Device# show l2protocol`	Displays the Layer 2 tunnel ports on the device, including the protocols configured, the thresholds, and the counters.
Step 14	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file. **Note** Use the **no switchport mode trunk**, the **no udld enable**, and the **no channel group** *channel-group-number* **mode desirable** interface configuration commands to return the interface to the default settings.

Configuration Examples for IEEE 802.1Q and Layer 2 Protocol Tunneling

Example: Configuring an IEEE 802.1Q Tunneling Port

The following 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
```

Examples: Configuring the SP Edge and Customer Switches

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:

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

SP edge switch 2 configuration:

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

```
Switch(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.

```
Switch(config)# interface gigabitethernet1/0/1
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# udld enable
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# exit
Switch(config)# interface gigabitethernet1/0/2
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# udld enable
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# exit
Switch(config)# interface gigabitethernet1/0/3
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# udld enable
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# exit
Switch(config)# interface gigabitethernet1/0/4
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# udld enable
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# exit
Switch(config)# interface port-channel 1
Switch(config-if)# shutdown
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Monitoring Tunneling Status

The following table describes the commands used to monitor tunneling status.

Table 1: 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.

Where to Go Next

You can configure the following:

- VTP
- VLANs
- VLAN Trunking
- VLAN Membership Policy Server (VMPS)
- Voice VLANs

Additional References

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Catalyst 2960-XR Switch VLAN Management Command Reference*

Standards and RFCs

Standard/RFC	Title
—	—

MIBs

MIB	MIBs Link
All the supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature History and Information for Tunneling

Release	Modification
Cisco IOS Release 15.0(2)EX	This feature was introduced.