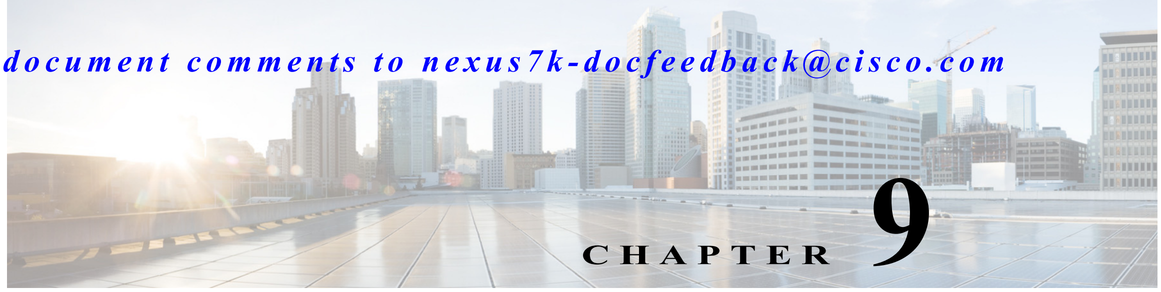


CHAPTER 9

Configuring Q-in-Q VLAN Tunnels

This chapter describes how to configure IEEE 802.1Q-in-Q (Q-in-Q) VLAN tunnels and Layer 2 protocol tunneling on Cisco NX-OS devices.

This chapter includes the following sections:

Information About Q-in-Q Tunnels

A Q-in-Q VLAN tunnel enables a service provider to segregate the traffic of different customers in their infrastructure, while still giving the customer a full range of VLANs for their internal use by adding a second 802.1Q tag to an already tagged frame.

This section includes the following topics:

Q-in-Q 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 of 4096 of the 802.1Q specification.

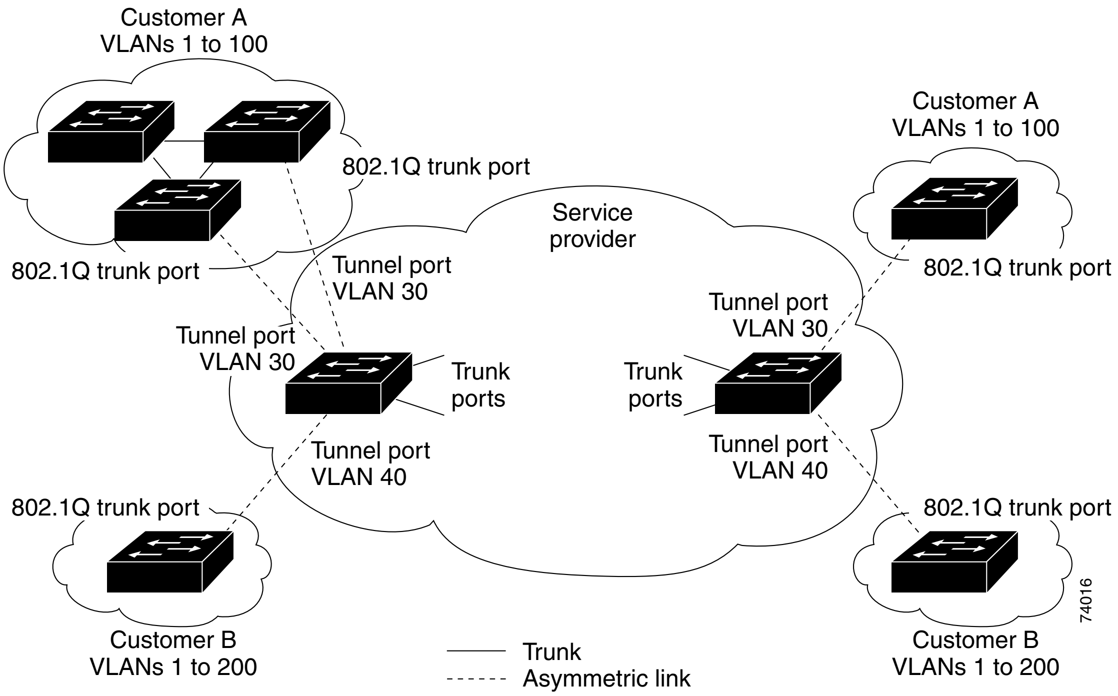
Note Q-in-Q is supported on port channels and vPC. To configure a port channel as an asymmetrical link, all ports in the port channel must have the same tunneling configuration.

Using the 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 infrastructure even when they appear to be on the same VLAN. The 802.1Q tunneling expands VLAN space by using a VLAN-in-VLAN hierarchy and tagging the tagged packets. A port configured to support 802.1Q tunneling is called a *tunnel port*. When you configure tunneling, you assign a tunnel port to a VLAN that is dedicated to tunneling. Each customer requires a separate VLAN, but that VLAN supports all of the customer's VLANs.

Customer traffic tagged in the normal way with appropriate VLAN IDs come from an 802.1Q trunk port on the customer device and into a tunnel port on the service-provider edge switch. The link between the customer device and the edge switch is an *asymmetric link* because one end is configured as an 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 unique to each customer. See Figure 9-1.

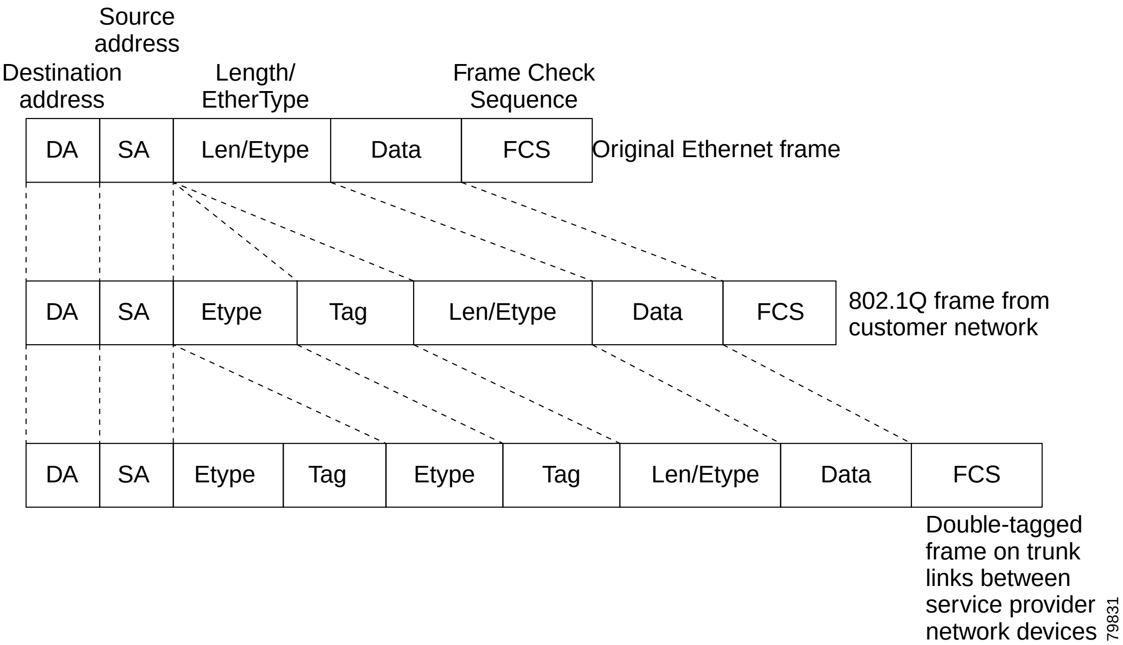
Note Selective Q-in-Q tunneling is not supported. All frames entering the tunnel port will be subject to Q-in-Q tagging.

Figure 9-1 802.1Q-in-Q Tunnel Ports



Packets entering the tunnel port on the service-provider edge switch, which are already 802.1Q-tagged with the appropriate VLAN IDs, are encapsulated with another layer of an 802.1Q tag that contains a VLAN ID unique to the customer. The original 802.1Q tag from the customer is preserved in the encapsulated packet. Therefore, packets that enter the service-provider infrastructure are double-tagged. The outer tag contains the customer's access VLAN ID (as assigned by the service provider), and the inner VLAN ID is the VLAN of the incoming traffic (as assigned by the customer). This double tagging is called tag stacking, Double-Q, or Q-in-Q as shown in Figure 9-2.

Figure 9-2 Untagged, 802.1Q-Tagged, and Double-Tagged Ethernet Frames



By using this method, the VLAN ID space of the outer tag is independent of the VLAN ID space of the inner tag. A single outer VLAN ID can represent the entire VLAN ID space for an individual customer. This technique allows the customer's Layer 2 network to extend across the service provider network, potentially creating a virtual LAN infrastructure over multiple sites.

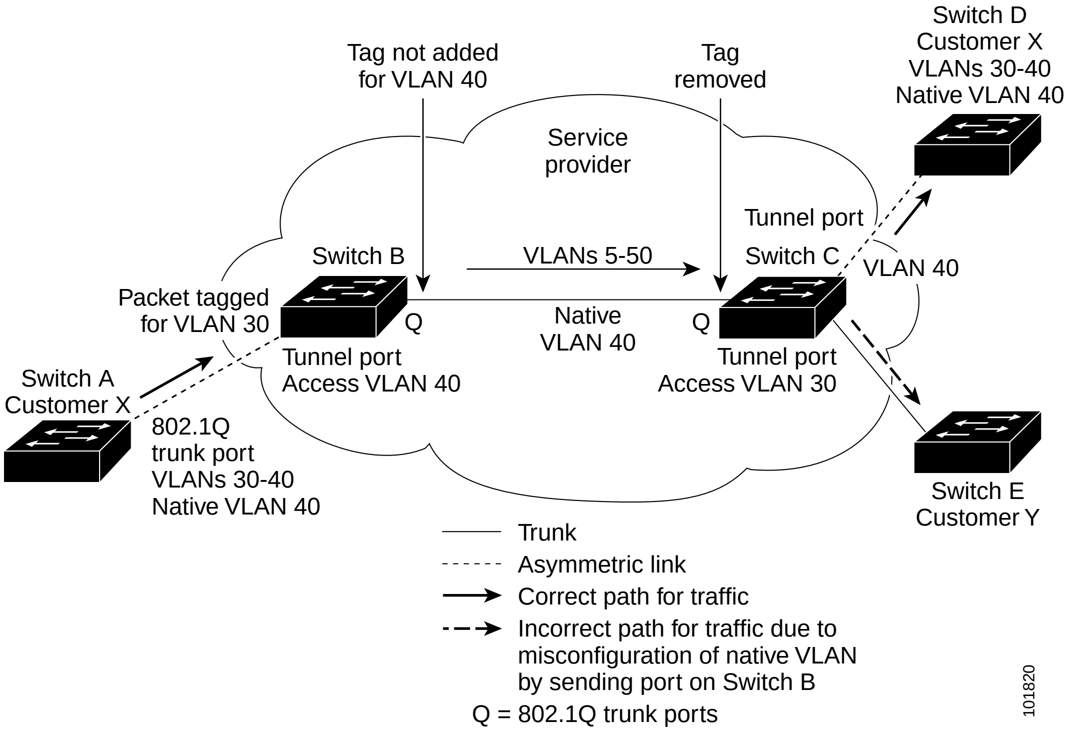
Note Hierarchical tagging, that is multi-level dot1q tagging Q-in-Q, is not supported.

Native VLAN Hazard

When configuring 802.1Q tunneling on an edge switch, you must use 802.1Q trunk ports for sending out packets into the service-provider network. However, packets that go through the core of the service-provider network might be carried through 802.1Q trunks, ISL trunks, or non-trunking links. When 802.1Q trunks are used in these core switches, the native VLANs of the 802.1Q trunks must not match any native VLAN of the dot1q-tunnel port on the same switch because traffic on the native VLAN is not tagged on the 802.1Q transmitting trunk port.

In Figure 9-3, VLAN 40 is configured as the native VLAN for the 802.1Q trunk port from Customer X at the ingress edge switch in the service-provider network (Switch B). Switch A of Customer X sends a tagged packet on VLAN 30 to the ingress tunnel port of Switch B in the service-provider network belonging to access VLAN 40. Because the access VLAN of the tunnel port (VLAN 40) is the same as the native VLAN of the edge-switch trunk port (VLAN 40), the 802.1Q tag is not added to tagged packets that are 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-edge switch (Switch C) and is misdirected through the egress switch tunnel port to Customer Y.

Figure 9-3 Native VLAN Hazard



These are a couple ways to solve the native VLAN problem:

- Configure the edge switch so that all packets going out an 802.1Q trunk, including the native VLAN, are tagged by using the **vlan dot1q tag native** command. If the switch is configured to tag native VLAN packets on all 802.1Q trunks, the switch accepts untagged packets but sends only tagged packets.

Note The **vlan dot1q tag native** command is a global command that affects the tagging behavior on all trunk ports.

- Ensure that the native VLAN ID on the edge switch 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.

Information About Layer 2 Protocol Tunneling

Customers at different sites connected across a service-provider network need to run various Layer 2 protocols to scale their topology to include all remote sites, as well as the local sites. The spanning Tree Protocol (STP) must run properly, and every VLAN should build a proper spanning tree that includes the local site and all remote sites across the service-provider infrastructure. Cisco Discovery Protocol (CDP) must be able to discover neighboring Cisco devices from local and remote sites, and the VLAN Trunking Protocol (VTP) must provide consistent VLAN configuration throughout all sites in the customer network.

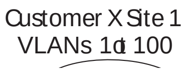
When protocol tunneling is enabled, edge switches on the inbound side of the service-provider infrastructure encapsulate Layer 2 protocol packets with a special MAC address and send them across the service-provider network. Core switches in the network do not process these packets, but forward them as normal packets. Bridge protocol data units (BPDUs) for CDP, STP, or VTP cross the service-provider infrastructure and are delivered to customer switches on the outbound side of the service-provider network. Identical packets are received by all customer ports on the same VLANs.

If protocol tunneling is not enabled on 802.1Q tunneling ports, remote switches at the receiving end of the service-provider network do not receive the BPDUs and cannot properly run STP, CDP, 802.1X, and VTP. When protocol tunneling is enabled, Layer 2 protocols within each customer's network are totally separate from those running within the service-provider network. Customer switches on different sites that send traffic through the service-provider network with 802.1Q tunneling achieve complete knowledge of the customer's VLAN.

Note Layer 2 protocol tunneling works by tunneling BPDUs in software. A large number of BPDUs coming into the SUP will cause the CPU load to go up. You may need to make use of hardware rate limiters to reduce the load on the SUP CPU. See the "Configuring the Rate Limit for Layer 2 Protocol Tunnel Ports" section on page 9-13.

For example, in Figure 9-4, Customer X has four switches in the same VLAN that are connected through the service-provider network. If the network does not tunnel BPDUs, switches on the far ends of the network cannot properly run the STP, CDP, 802.1X, and VTP protocols.

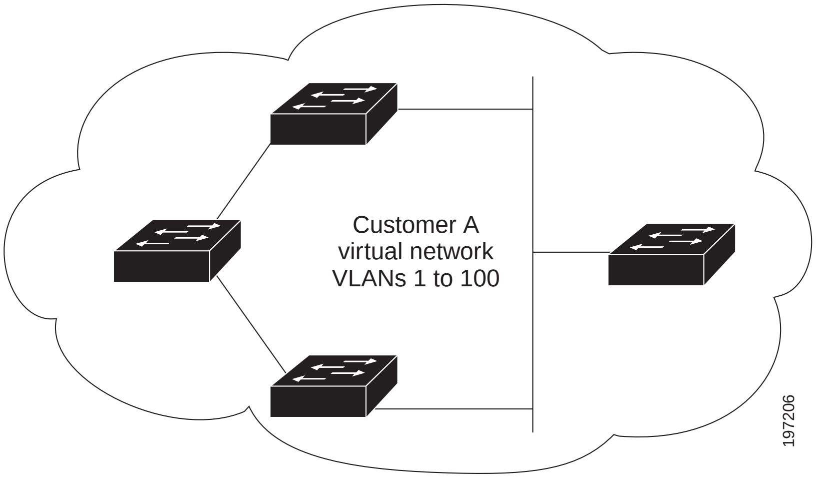
Figure 9-4 Layer 2 Protocol Tunneling



In the preceding 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.

Figure 9-5 shows the resulting topology on the customer's network when BPDU tunneling is not enabled.

Figure 9-5 Virtual Network Topology Without BPDU Tunneling



Licensing Requirements for Q-in-Q Tunnels

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	802.1Q-in-Q VLAN tunneling and L2 protocol tunneling require no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Guidelines and Limitations

Q-in-Q tunnels and Layer 2 tunneling have the following configuration guidelines and limitations:

- Switches in the service-provider network must be configured to handle the increase in MTU size due to Q-in-Q tagging.
- MAC address learning for Q-in-Q tagged packets is based on the outer VLAN (Service Provider VLAN) tag. Packet forwarding issues may occur in deployments where a single MAC address is used across multiple inner (customer) VLANs.
- Layer 3 and higher parameters cannot be identified in tunnel traffic (for example, Layer 3 destination and source addresses). Tunneled traffic cannot be routed.
- Cisco Nexus 7000 Series devices can provide only MAC-layer ACL/QoS for tunnel traffic (VLAN IDs and src/dest MAC addresses).
- You should use MAC address-based frame distribution.

- Asymmetrical links do not support the Dynamic Trunking Protocol (DTP) because only one port on the link is a trunk. You must configure the 802.1Q trunk port on an asymmetrical link to trunk unconditionally.
- You cannot configure the 802.1Q tunneling feature on ports that are configured to support private VLANs. Private VLAN are not required in these deployments.
- You must disable IGMP snooping on the tunnel VLANs.
- Control Plane Policing (CoPP) is not supported.
- You should run the **vlan dot1Q tag native** command to maintain the tagging on the native VLAN and drop untagged traffic. This will prevent native VLAN misconfigurations,
- You must manually configure the 802.1Q interfaces to be edge ports.
- Dot1x tunneling is not supported.
- You should perform an EPLD upgrade to newer versions in order for EtherType configuration to take effect on some Cisco Nexus devices.

Configuring Q-in-Q Tunnels and Layer 2 Protocol Tunneling

This section describes how to configure Q-in-Q tunnels and Layer 2 protocol tunneling on Cisco Nexus 7000 Series devices.

This section includes the following topics:

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Creating a 802.1Q Tunnel Port

You create the dot1q-tunnel port using the **switchport mode** command.

Note You need to set the 802.1Q tunnel port to an edge port with the **spanning-tree port type edge** command. The VLAN membership of the port is changed using the **switchport access vlan** *vlan-id* command.

You should disable IGMP snooping on the access VLAN allocated for the dot1q-tunnel port to allow multicast packets to traverse the Q-in-Q tunnel.

BEFORE YOU BEGIN

You must first configure the interface as a switchport.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port*
3. **switchport**
4. **switchport mode dot1q-tunnel**
5. **no switchport mode dot1q-tunnel**
6. **exit**
7. **show dot1q-tunnel [interface** *if-range*]
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`configure terminal`**
	Example:
	`switch# configure terminal`
		Enters global configuration mode.

Step 2	**`interface ethernet`** *`slot/port`*
	Example:
	`switch(config)# interface ethernet 7/1`
		Specifies an interface to configure, and enters interface configuration mode.

Step 3	**`switchport`**
	Example:
	`switch(config-if)# switchport`
		Sets the interface as a Layer 2 switching port.

Step 4	**`switchport mode dot1q-tunnel`**
	Example:
	`switch(config-if)# switchport mode dot1q-tunnel`
		Creates a 802.1Q tunnel on the port. The port will go down and reinitialize (port flap) when the interface mode is changed. BPDU filtering is enabled and CDP is disabled on tunnel interfaces.

Step 5	**`no switchport mode`**
	Example:
	`switch(config-if)# no switchport mode`
		(Optional) Disables the 802.1Q tunnel on the port.

Step 6	**`exit`**
	Example:
	`switch(config-if)# exit`
	`switch(config)# exit`
		Exits configuration mode.

	Command	Purpose
Step 7	**show dot1q-tunnel** [**interface** *if-range*] **Example:** `switch# show dot1q-tunnel`	(Optional) Displays all ports that are in dot1q-tunnel mode. Optionally you can specify an interface or range of interfaces to display.
Step 8	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

This example shows how to create an 802.1Q tunnel port:

```
switch# configure terminal
switch(config)# interface ethernet 7/1
switch(config-if)# switchport
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# exit
switch(config)# exit
switch# show dot1q-tunnel
```

Changing the EtherType for Q-in-Q

You can change the 802.1Q EtherType value to be used for Q-in-Q encapsulation.

Note You must set the EtherType only on the egress trunk interface that carries double tagged frames (the trunk interface that connects the service providers). If you change the EtherType on one side of the trunk, you must set the same value on the other end of the trunk (symmetrical configuration).

Caution The EtherType value you set will affect all the tagged packets going out on the interface (not just Q-in-Q packets).

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port*
3. **switchport**
4. **switchport dot1q ethertype** *value*
5. **no switchport dot1q ethertype**
6. **exit**
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`configure terminal`**
	
	Example:
	`switch# configure terminal`
		Enters global configuration mode.

Step 2	**`interface ethernet`** *`slot/port`*
	
	Example:
	`switch(config)#` **`interface ethernet 7/1`**
		Specifies an interface to configure, and enters interface configuration mode.

Step 3	**`switchport`**
	
	Example:
	`switch(config-if)# switchport`
		Sets the interface as a Layer 2 switching port.

Step 4	**`switchport dot1q ethertype`** *`value`*
	
	Example:
	`switch(config-if)# switchport dot1q`
	`ethertype 0x9100`
		Sets the EtherType for the Q-in-Q tunnel on the port.

Step 5	**`no switchport dot1q ethertype`**
	
	Example:
	`switch(config-if)# no switchport dot1q`
	`ethertype`
		(Optional) Resets the EtherType on the port to the default value of 0x8100.

Step 6	**`exit`**
	
	Example:
	`switch(config-if)# exit`
	`switch(config)# exit`
		Exits configuration mode.

Step 7	**`copy running-config startup-config`**
	
	Example:
	`switch# copy running-config startup-config`
		(Optional) Copies the running configuration to the startup configuration.

This example shows how to create an 802.1Q tunnel port:

```
switch# configure terminal
switch(config)# interface ethernet 7/1
switch(config-if)# switchport
switch(config-if)# switchport dot1q ethertype 0x9100
switch(config-if)# exit
switch(config)# exit
switch# show dot1q-tunnel
```

Enabling the Layer 2 Protocol Tunnel

You can enable protocol tunneling on the 802.1Q tunnel port.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*

3. **switchport**
4. **switchport mode dot1q-tunnel**
5. **l2protocol tunnel** [**cdp** | **stp** | **vtp**]
6. **no l2protocol tunnel** [**cdp** | **stp** | **vtp**]
7. **exit**
8. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`configure terminal`**
	`Example:`
	`switch# configure terminal`
		Enters global configuration mode.

Step 2	**`interface ethernet`** *`slot/port`*
	`Example:`
	`switch(config)# interface ethernet 7/1`
		Specifies an interface to configure, and enters interface configuration mode.

Step 3	**`switchport`**
	`Example:`
	`switch(config-if)# switchport`
		Sets the interface as a Layer 2 switching port.

Step 4	**`switchport mode dot1q-tunnel`**
	`Example:`
	`switch(config-if)# switchport mode dot1q-tunnel`
		Creates a 802.1Q tunnel on the port.

Step 5	**`l2protocol tunnel`** `[`**`cdp`** `|` **`stp`** `|` **`vtp`**`]`
	`Example:`
	`switch(config-if)# l2protocol tunnel stp`
		Enables Layer 2 protocol tunneling. Optionally, you can enable CDP, STP, or VTP tunneling.

Step 6	**`no l2protocol tunnel`** `[`**`cdp`** `|` **`stp`** `|` **`vtp`**`]`
	`Example:`
	`switch(config-if)# no l2protocol tunnel`
		(Optional) Disables protocol tunneling.

Step 7	**`exit`**
	`Example:`
	`switch(config-if)# exit`
	`switch(config)# exit`
		Exits configuration mode.

Step 8	**`copy running-config startup-config`**
	`Example:`
	`switch# copy running-config startup-config`
		(Optional) Copies the running configuration to the startup configuration.

This example shows how to enable protocol tunneling on an 802.1Q tunnel port:

```
switch# configure terminal
switch(config)# interface ethernet 7/1
switch(config-if)# switchport
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# l2protocol tunnel stp
switch(config-if)# exit
switch(config)# exit
```

Configuring Global CoS for L2 Protocol Tunnel Ports

You can specify a Class of Service (CoS) value globally so that ingress BPDUs on the tunnel ports are encapsulated with the specified class.

SUMMARY STEPS

1. **configure terminal**
2. **l2protocol tunnel cos** *value*
3. **no l2protocol tunnel cos**
4. **exit**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`configure terminal`**
	Example:
	`switch# configure terminal`
		Enters global configuration mode.

Step 2	**`l2protocol tunnel cos`** *`cos-value`*
	Example:
	`switch(config)# l2protocol tunnel cos 6`
		Specifies a global CoS value on all Layer 2 protocol tunneling ports. The default cos-value is 5.

Step 3	**`no l2protocol tunnel cos`**
	Example:
	`switch(config)# no l2protocol tunnel cos`
		(Optional) Sets the global CoS value to default.

Step 4	**`exit`**
	Example:
	`switch(config)# exit`
		Exits configuration mode.

Step 5	**`copy running-config startup-config`**
	Example:
	`switch# copy running-config startup-config`
		(Optional) Copies the running configuration to the startup configuration.

This example shows how to specify a global CoS value for the purpose of Layer 2 protocol tunneling:

```
switch# configure terminal
switch(config)# l2protocol tunnel cos 6
switch(config)# exit
```

Configuring the Rate Limit for Layer 2 Protocol Tunnel Ports

You can specify the hardware rate limiter configuration for Layer 2 protocol tunneling. The default is set to 500 packets per second. Depending on the load or the number of VLANs to be tunneled for a customer, you may need to adjust this value to prevent STP errors on the customer's network.

SUMMARY STEPS

1. **configure terminal**
2. **hardware rate-limiter layer-2 l2pt** *packets-per-sec*
3. **no hardware rate-limiter layer-2 l2pt**

DETAILED STEPS

	Command	Purpose

Step 1	**`configure terminal`**
	`Example:`
	`switch# configure terminal`
		Enters global configuration mode.

Step 2	**`hardware rate-limiter layer-2 l2pt`** *`packets-per-second`*
	`Example:`
	`switch(config)# hardware rate-limiter layer-2 l2pt 4096`
		Sets the threshold in packets per second above which incoming protocol packets from dot1q-tunnel ports are dropped in hardware. Valid values are from 0 to 30000.

Step 3	**`no hardware rate-limiter layer-2 l2pt`**
	`Example:`
	`switch(config)# no hardware rate-limiter layer-2 l2pt`
		(Optional) Resets the threshold values to the default of 500 packets per second.

Configuring Thresholds for Layer 2 Protocol Tunnel Ports

You can specify the port drop and shutdown value for a Layer 2 protocol tunneling port.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port*
3. **switchport**
4. **switchport mode dot1q-tunnel**
5. **l2protocol tunnel drop-threshold [cdp | stp | vtp]** *packets-per-sec*
6. **no l2protocol tunnel drop-threshold [cdp | stp | vtp]**
7. **l2protocol tunnel shutdown-threshold [cdp | stp | vtp]** *packets-per-sec*
8. **no l2protocol tunnel shutdown-threshold [cdp | stp | vtp]**
9. **exit**
10. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal	Enters global configuration mode.
Step 2	**interface ethernet** *slot/port* **Example:** switch(config)# interface ethernet 7/1	Specifies an interface to configure, and enters interface configuration mode.
Step 3	**switchport** **Example:** switch(config-if)# switchport	Sets the interface as a Layer 2 switching port.
Step 4	**switchport mode dot1q-tunnel** **Example:** switch(config-if)# switchport mode dot1q-tunnel	Creates a 802.1Q tunnel on the port.
Step 5	**l2protocol tunnel drop-threshold** [**cdp** | **stp** | **vtp**] *packets-per-sec* **Example:** switch(config)# l2protocol tunnel drop-threshold 3000	Specifies the maximum number of packets that can be processed on an interface before being dropped. Optionally, you can specify CDP, STP, or VTP. Valid values for the packets are from 1 to 4096.
Step 6	**no l2protocol tunnel drop-threshold** [**cdp** | **stp** | **vtp**] **Example:** switch(config)# no l2protocol tunnel drop-threshold	(Optional) Resets the threshold values to 0 and disables the drop threshold.
Step 7	**l2protocol tunnel shutdown-threshold** [**cdp** | **stp** | **vtp**] *packets-per-sec* **Example:** switch(config)# l2protocol tunnel shutdown-threshold 3000	Specifies the maximum number of packets that can be processed on an interface. When the number of packets is exceeded, the port is put in error-disabled state. Optionally, you can specify CDP, STP, or VTP. Valid values for the packets is from 1 to 4096.
Step 8	**no l2protocol tunnel shutdown-threshold** [**cdp** | **stp** | **vtp**] **Example:** switch(config)# no l2protocol tunnel shutdown-threshold	(Optional) Resets the threshold values to 0 and disables the shutdown threshold.
Step 9	**exit** **Example:** switch(config)# exit	Exits configuration mode.
Step 10	**copy running-config startup-config** **Example:** switch# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

Verifying the Q-in-Q Configuration

To display Q-in-Q tunnel and Layer 2 protocol tunneling configuration information, perform one of the following tasks:

Command	Purpose
`clear l2protocol tunnel counters [interface if-range]`	Clears all the statistics counters. If no interfaces are specified, the Layer 2 protocol tunnel statistics are cleared for all interfaces.
`show dot1q-tunnel [interface if-range]`	Displays a range of interfaces or all interfaces that are in dot1q-tunnel mode.
`show l2protocol tunnel [interface if-range | vlan vlan-id]`	Displays Layer 2 protocol tunnel information for a range of interfaces, for all dot1q-tunnel interfaces that are part of a specified VLAN or all interfaces.
`show l2protocol tunnel summary`	Displays a summary of all ports that have Layer 2 protocol tunnel configurations.
`show running-config l2pt`	Displays the current Layer 2 protocol tunnel running configuration.

Configuration Examples for Q-in-Q and Layer 2 Protocol Tunneling

This example shows a service provider switch that is configured to process Q-in-Q for traffic coming in on Ethernet 7/1. A Layer 2 protocol tunnel is enabled for STP BPDUs. The customer is allocated VLAN 10 (outer VLAN tag).

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# vlan 10
switch(config-vlan)# no shutdown
switch(config-vlan)# no ip igmp snooping
switch(config-vlan)# exit
switch(config)# interface ethernet 7/1
switch(config-if)# switchport
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport access vlan 10
switch(config-if)# spanning-tree port type edge
switch(config-if)# l2protocol tunnel stp
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# exit
switch#
```

Feature History for Q-in-Q Tunnels and Layer 2 Protocol Tunneling

Table 9-1 lists the release history for this feature.

Table 9-1 Feature History for Q-in-Q Tunnels and Layer 2 Protocol Tunneling

Feature Name	Releases	Feature Information
Q-in-Q VLAN Tunnels	5.0(2)	This feature was introduced.
L2 Protocol Tunneling	5.0(2)	This feature was introduced.