



Configuring 802.1Q and Layer 2 Protocol Tunneling

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. The Catalyst 3750 switch supports 802.1Q tunneling and Layer 2 protocol tunneling. Unless otherwise noted, the term *switch* refers to a standalone switch and to a switch stack.



Note

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

This chapter contains these sections:

- [Understanding 802.1Q Tunneling, page 17-1](#)
- [Configuring 802.1Q Tunneling, page 17-4](#)
- [Understanding Layer 2 Protocol Tunneling, page 17-7](#)
- [Configuring Layer 2 Protocol Tunneling, page 17-10](#)
- [Monitoring and Maintaining Tunneling Status, page 17-18](#)

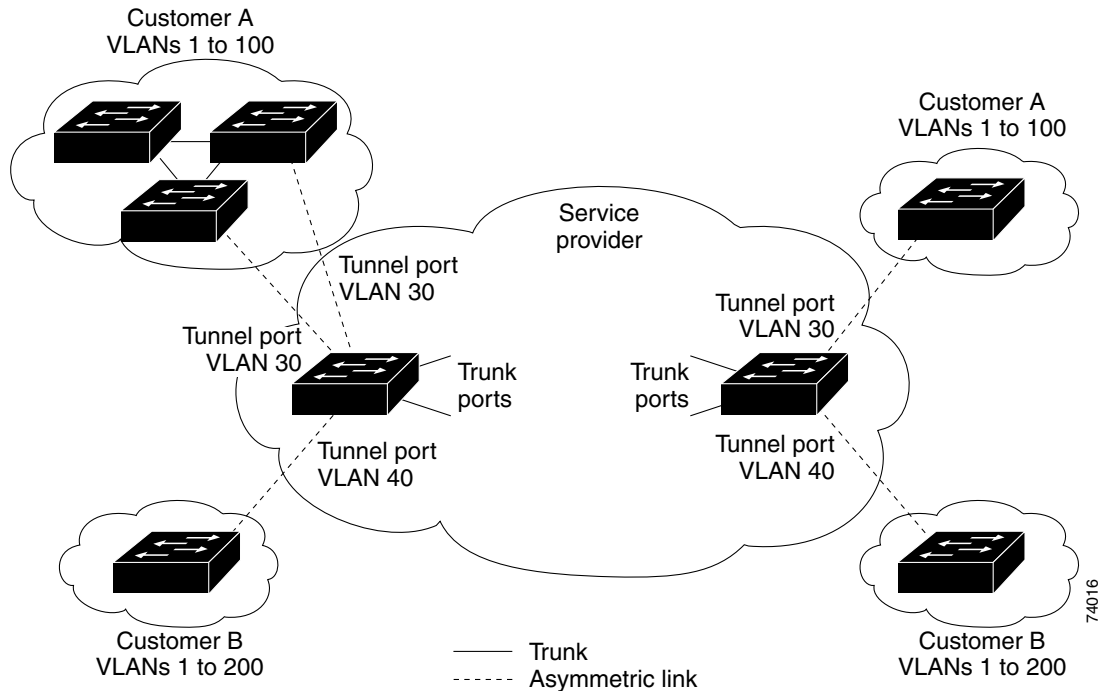
Understanding 802.1Q Tunneling

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 802.1Q specification.

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 network, even when they appear to be in the same VLAN. Using 802.1Q tunneling expands VLAN space by using a VLAN-in-VLAN hierarchy and retagging 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 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.

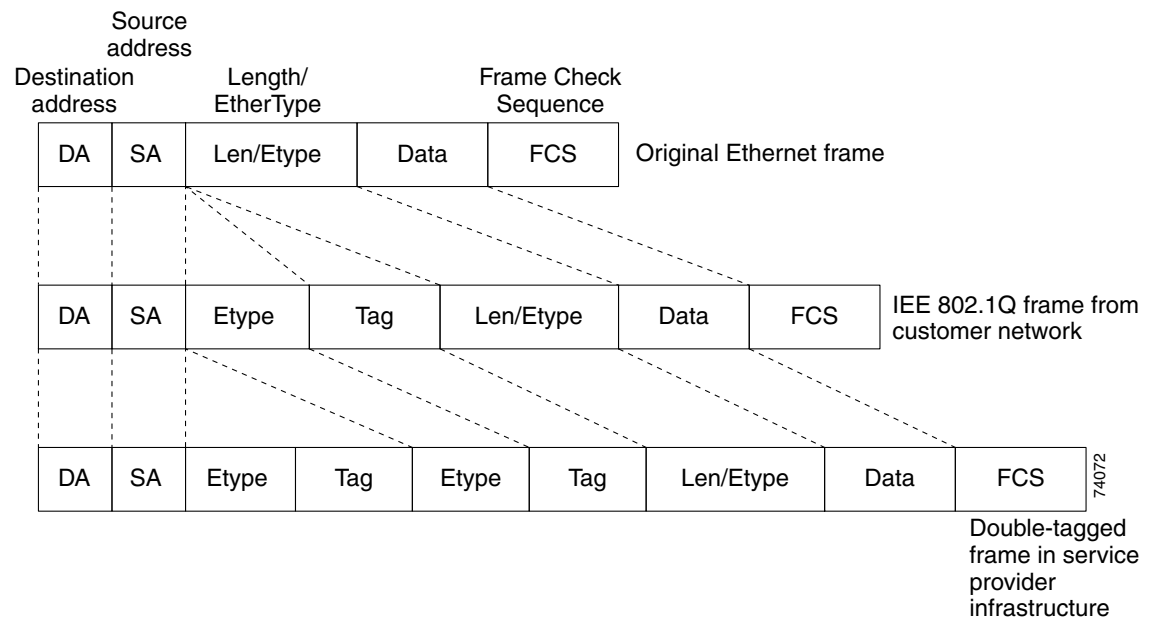
Customer traffic tagged in the normal way with appropriate VLAN IDs comes from an 802.1Q trunk port on the customer device and into a tunnel port on the service-provider edge switch. The link between the customer device and the edge switch is asymmetric because one end is configured as an 802.1Q trunk port, and the other end is configured as a tunnel port. You assign the tunnel port interface to an access VLAN ID that is unique to each customer. See [Figure 17-1](#).

Figure 17-1 802.1Q Tunnel Ports in a Service-Provider Network



Packets coming from the customer trunk port into the tunnel port on the service-provider edge switch are normally 802.1Q-tagged with the appropriate VLAN ID. The the tagged packets remain intact inside the switch and when they exit the trunk port into the service-provider network, they are encapsulated with another layer of an 802.1Q tag (called the *metro tag*) that contains the VLAN ID that is unique to the customer. The original customer 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 switch, the outer tag is stripped as the switch processes the packet. When the packet exits another trunk port on the same core switch, the same metro tag is again added to the packet. [Figure 17-2](#) shows the tag structures of the double-tagged packets.

Figure 17-2 Original (Normal), 802.1Q, and Double-Tagged Ethernet Packet Formats

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

In [Figure 17-1](#), Customer A was assigned VLAN 30, and Customer B was assigned VLAN 40. Packets entering the edge switch tunnel ports with 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 switch 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 switch are treated as untagged packets, whether they are untagged or already tagged with 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 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 switch is a standalone switch or a stack member. All configuration is done on the stack master.

Configuring 802.1Q Tunneling

This section includes this information about configuring 802.1Q tunneling:

- [Default 802.1Q Tunneling Configuration, page 17-4](#)
- [802.1Q Tunneling Configuration Guidelines, page 17-4](#)
- [802.1Q Tunneling and Other Features, page 17-6](#)
- [Configuring an 802.1Q Tunneling Port, page 17-6](#)

Default 802.1Q Tunneling Configuration

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

802.1Q Tunneling Configuration Guidelines

When you configure 802.1Q tunneling, you should always use an asymmetrical link between the customer device and the edge switch, with the customer device port configured as an 802.1Q trunk port and the edge switch 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

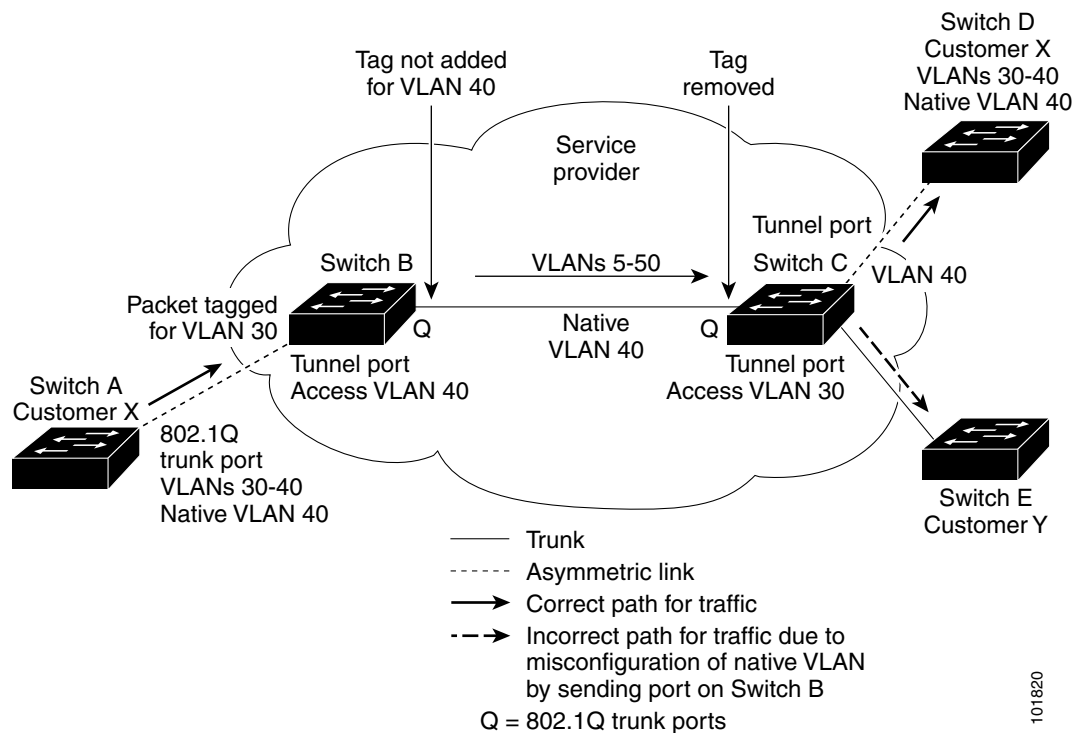
When configuring 802.1Q tunneling on an edge switch, you must use 802.1Q trunk ports for sending packets into the service-provider network. However, packets going through the core of the service-provider network can be carried through 802.1Q trunks, ISL trunks, or nontrunking links. When 802.1Q trunks are used in these core switches, the native VLANs of the 802.1Q trunks must not match any native VLAN of the nontrunking (tunneling) port on the same switch because traffic on the native VLAN would not be tagged on the 802.1Q sending trunk port.

See [Figure 17-3](#). 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, 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-switch 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-edge switch (Switch C) and is misdirected through the egress switch tunnel port to Customer Y.

These are some ways to solve this problem:

- Use ISL trunks between core switches in the service-provider network. Although customer interfaces connected to edge switches must be 802.1Q trunks, we recommend using ISL trunks for connecting switches in the core layer.
- Use the **vlan dot1q tag native** global configuration command to configure the edge switch so that all packets going out an 802.1Q trunk, including the native VLAN, are tagged. 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.
- 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.

Figure 17-3 Potential Problem with 802.1Q Tunneling and Native VLANs



System MTU

The default system MTU for traffic on the switch is 1500 bytes. You can configure Fast Ethernet ports to support frames larger than 1500 bytes by using the **system mtu** global configuration command. You can configure Gigabit Ethernet ports to support frames larger than 1500 bytes by using the **system mtu jumbo** global configuration command. Because the 802.1Q tunneling feature increases the frame size by 4 bytes when the metro tag is added, you must configure all switches in the service-provider network to be able to process maximum frames by increasing the switch system MTU size to at least 1504 bytes. The maximum allowable system MTU for Gigabit Ethernet interfaces is 9000 bytes; the maximum system MTU for Fast Ethernet interfaces is 1546 bytes.

802.1Q Tunneling and Other Features

Although 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 802.1Q ports. Packets received from a tunnel port are forwarded based only on Layer 2 information. If routing is enabled on a switch virtual interface (SVI) that includes tunnel ports, untagged IP packets received from the tunnel port are recognized and routed by the switch. Customer 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 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 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 802.1Q tunnel ports.
- Dynamic Trunking Protocol (DTP) is not compatible with 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 802.1Q tunnel ports.
- When a port is configured as an 802.1Q tunnel port, spanning-tree bridge protocol data unit (BPDU) filtering is automatically enabled on the interface. Cisco Discovery Protocol (CDP) is automatically disabled on the interface.

Configuring an 802.1Q Tunneling Port

Beginning in privileged EXEC mode, follow these steps to configure a port as an 802.1Q tunnel port:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	interface <i>interface-id</i>	Enter 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 switch. Valid interfaces include physical interfaces and port-channel logical interfaces (port channels 1 to 48).
Step 3	switchport access vlan <i>vlan-id</i>	Specify the default VLAN, which is used if the interface stops trunking. This VLAN ID is specific to the particular customer.
Step 4	switchport mode dot1q-tunnel	Set the interface as an 802.1Q tunnel port.
Step 5	exit	Return to global configuration mode.

	Command	Purpose
Step 6	vlan dot1q tag native	(Optional) Set the switch to enable tagging of native VLAN packets on all 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.
Step 7	end	Return to privileged EXEC mode.
Step 8	show running-config	Display the ports configured for 802.1Q tunneling.
	show dot1q-tunnel	Display the ports that are in tunnel mode.
Step 9	show vlan dot1q tag native	Display 802.1Q native VLAN tagging status.
Step 10	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Use the **no switchport mode dot1q-tunnel** interface configuration command to return the port to the default state of dynamic desirable. Use the **no vlan dot1q tag native** global configuration command to disable tagging of native VLAN packets.

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
```

Understanding Layer 2 Protocol Tunneling

Customers at different sites connected across a service-provider network need to use various Layer 2 protocols to scale their topologies to include all remote sites, as well as the local sites. 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 network. Cisco Discovery Protocol (CDP) must discover neighboring Cisco devices from local and remote sites. 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 network 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. Layer 2 protocol data units (PDUs) for CDP, STP, or VTP cross the service-provider network 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 with these results:

- 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 switches through the service provider.

**Note**

To provide interoperability with third-party vendors, you can use the Layer 2 protocol-tunnel bypass feature. Bypass mode transparently forwards control PDUs to vendor switches that have different ways of controlling protocol tunneling. You implement bypass mode by enabling Layer 2 protocol tunneling on the egress trunk port.

Layer 2 protocol tunneling can be used independently or can enhance 802.1Q tunneling. 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 PDUs and cannot properly run STP, CDP, 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. If 802.1Q tunneling is not used, you can still enable Layer 2 protocol tunneling by connecting to the customer switch through access ports and by enabling tunneling on the service-provider access port.

For example, in [Figure 17-4](#), 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 [Figure 17-5](#).

Figure 17-4 Layer 2 Protocol Tunneling

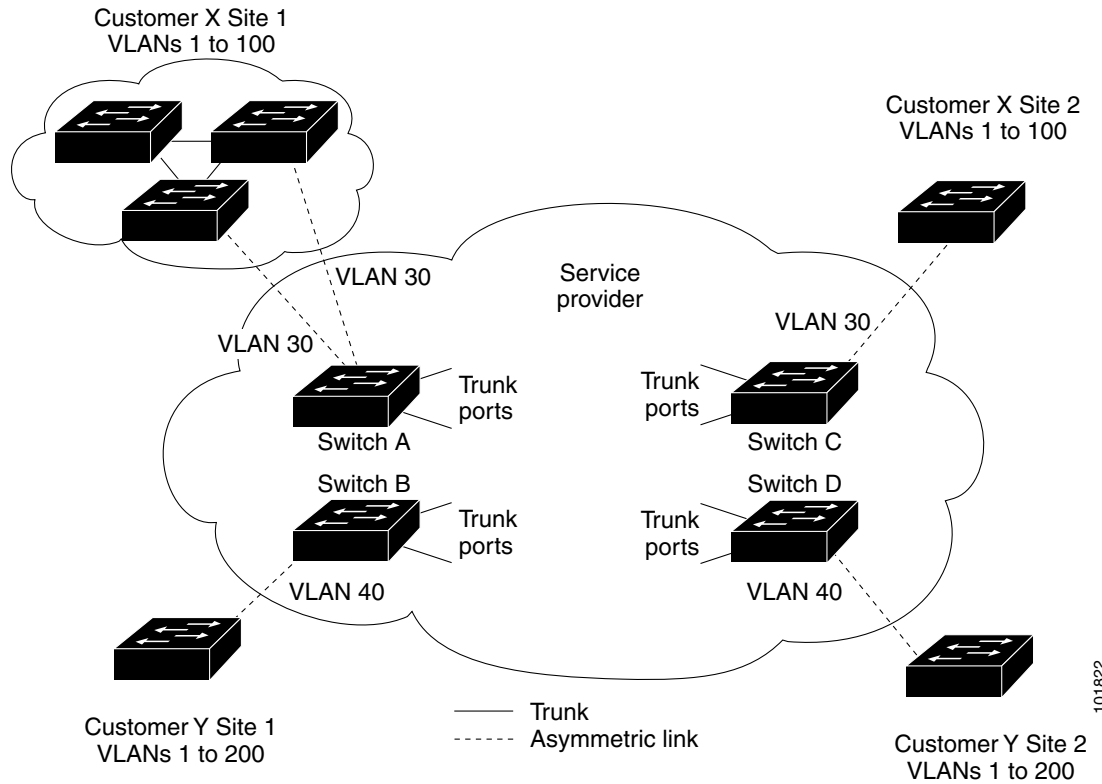
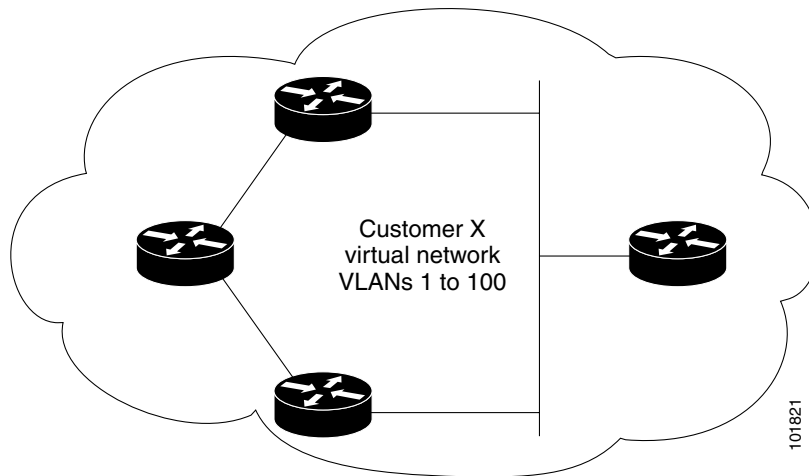


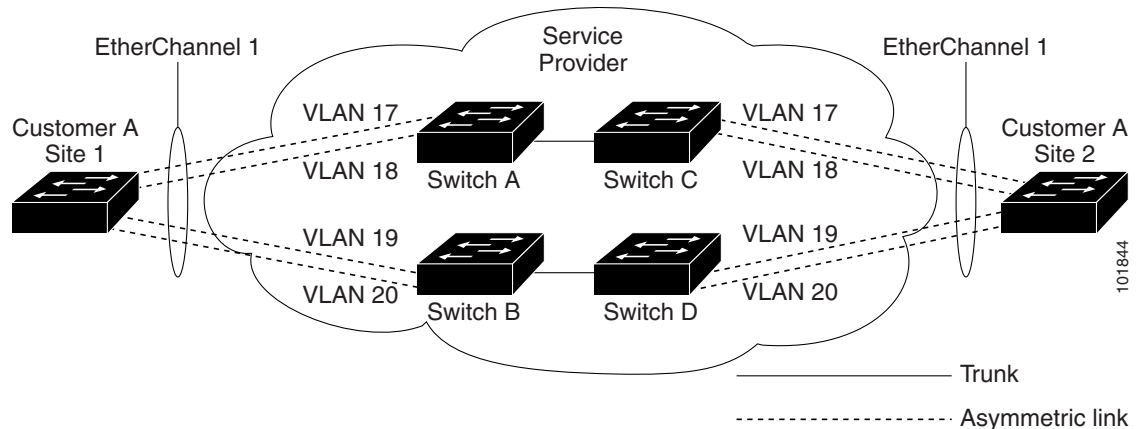
Figure 17-5 Layer 2 Network Topology without Proper Convergence



In an SP network, you can use Layer 2 protocol tunneling to enhance the creation of EtherChannels by emulating a point-to-point network topology. When you enable protocol tunneling (PAgP or LACP) on the SP switch, remote customer switches receive the PDUs and can negotiate the automatic creation of EtherChannels.

For example, in [Figure 17-6](#), 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. See the “[Configuring Layer 2 Tunneling for EtherChannels](#)” section on [page 17-14](#) for instructions.

Figure 17-6 Layer 2 Protocol Tunneling for EtherChannels



Configuring Layer 2 Protocol Tunneling

You can enable Layer 2 protocol tunneling (by protocol) on the ports that are connected to the customer in the edge switches of the service-provider network. The service-provider edge switches connected to the customer switch perform the tunneling process. Edge-switch tunnel ports are connected to customer 802.1Q trunk ports. Edge-switch access ports are connected to customer access ports. The edge switches connected to the customer switch perform the tunneling process.

You can enable Layer 2 protocol tunneling on ports that are configured as access ports or tunnel ports. You cannot enable Layer 2 protocol tunneling on ports configured with **switchport mode dynamic auto** or **dynamic desirable**.

The switch supports Layer 2 protocol tunneling for CDP, STP, and VTP. For emulated point-to-point network topologies, it also supports PAgP, LACP, and UDLD protocols.



Caution

PAgP, LACP, and UDLD protocol tunneling is only intended to emulate a point-to-point topology. An erroneous configuration that sends tunneled packets to many ports could lead to a network failure.

When the Layer 2 PDUs that entered the service-provider inbound edge switch through a Layer 2 protocol-enabled port exit through the trunk port into the service-provider network, the switch overwrites the customer PDU-destination MAC address with a well-known Cisco proprietary multicast address (01-00-0c-cd-cd-d0). If 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 switches ignore the inner tags and forward the packet to all trunk ports in the same metro VLAN. The edge switches 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. Therefore, the Layer 2 PDUs remain intact and are delivered across the service-provider infrastructure to the other side of the customer network.

See [Figure 17-4](#), with Customer X and Customer Y 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 2 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.

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.

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 and are in the same VLAN. All Layer 2 protocol tunneling configuration is handled by the stack master and distributed to all stack members.

This section has this information about configuring Layer 2 protocol tunneling:

- [Default Layer 2 Protocol Tunneling Configuration, page 17-11](#)
- [Layer 2 Protocol Tunneling Configuration Guidelines, page 17-11](#)
- [Configuring Layer 2 Protocol Tunneling, page 17-13](#)
- [Configuring Layer 2 Tunneling for EtherChannels, page 17-14](#)

Default Layer 2 Protocol Tunneling Configuration

Table 17-1 shows the default Layer 2 protocol tunneling configuration.

Table 17-1 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 for data packets, that value is the default used for Layer 2 PDUs. If none is configured, the default is 5.

Layer 2 Protocol Tunneling Configuration Guidelines

These are some configuration guidelines and operating characteristics of Layer 2 protocol tunneling:

- The switch supports PAgP, LACP, and UDLD tunneling for emulated point-to-point network topologies. Protocol tunneling is disabled by default but can be enabled for the individual protocols on 802.1Q tunnel ports or on access ports.
- If you enable PAgP or LACP tunneling, we recommend that you also enable UDLD on the interface for faster link-failure detection.

- Loopback detection is not supported on Layer 2 protocol tunneling of PAgP, LACP, or UDLD packets.
- EtherChannel port groups are compatible with tunnel ports when the 802.1Q configuration is consistent within an EtherChannel port group.
- If an encapsulated PDU (with the proprietary destination MAC address) is received from a tunnel port or an access port with Layer 2 tunneling enabled, the tunnel port shuts down to prevent loops. The port also shuts down when a configured shutdown threshold for the protocol is reached. You can manually re-enable the port (by entering a **shutdown** and a **no shutdown** command sequence). If error-disabled recovery is enabled, the operation is retried after a specified time interval.
- Only decapsulated PDUs are forwarded to the customer network. The spanning-tree instance running on the service-provider network does not forward BPDUs to tunnel ports. CDP packets are not forwarded from tunnel ports.
- The switch supports tunneling of CDP, STP, including multiple STP (MSTP), and VTP. Protocol tunneling is disabled by default but can be enabled for the individual protocols on 802.1Q tunnel ports or access ports.
- The switch does not support Layer 2 protocol tunneling on ports with **switchport mode dynamic auto** or **dynamic desirable**.
- DTP is not compatible with layer 2 protocol tunneling.
- The edge switches on the outbound side of the service-provider network restore the proper Layer 2 protocol and MAC address information and forward the packets to all tunnel and access ports in the same metro VLAN.
- For interoperability with third-party vendor switches, the switch supports a Layer 2 protocol-tunnel bypass feature. Bypass mode transparently forwards control PDUs to vendor switches that have different ways of controlling protocol tunneling. When Layer 2 protocol tunneling is enabled on ingress ports on a switch, egress trunk ports forward the tunneled packets with a special encapsulation. If you also enable Layer 2 protocol tunneling on the egress trunk port, this behavior is bypassed, and the switch forwards control PDUs without any processing or modification.
- The switch supports PAgP, LACP, and UDLD tunneling for emulated point-to-point network topologies. Protocol tunneling is disabled by default but can be enabled for the individual protocols on 802.1Q tunnel ports or on access ports.
- If you enable PAgP or LACP tunneling, we recommend that you also enable UDLD on the interface for faster link-failure detection.
- Loopback detection is not supported on Layer 2 protocol tunneling of PAgP, LACP, or UDLD packets.
- EtherChannel port groups are compatible with tunnel ports when the 802.1Q configuration is consistent within an EtherChannel port group.
- If an encapsulated PDU (with the proprietary destination MAC address) is received from a tunnel port or an access port with Layer 2 tunneling enabled, the tunnel port is shut down to prevent loops. The port also shuts down when a configured shutdown threshold for the protocol is reached. You can manually re-enable the port (by entering a **shutdown** and a **no shutdown** command sequence). If errdisable recovery is enabled, the operation is retried after a specified time interval.
- Only decapsulated PDUs are forwarded to the customer network. The spanning-tree instance running on the service-provider network does not forward BPDUs to tunnel ports. CDP packets are not forwarded from tunnel ports.
- When protocol tunneling is enabled on an interface, you can set a per-protocol, per-port, shutdown threshold for the PDUs generated by the customer network. If the limit is exceeded, the port shuts down. You can also limit BPDU rate by using QoS ACLs and policy maps on a tunnel port.

- When protocol tunneling is enabled on an interface, you can set a per-protocol, per-port, drop threshold for the PDUs generated by the customer network. If the limit is exceeded, the port drops PDUs until the rate at which it receives them is below the drop threshold.
- Because tunneled PDUs (especially STP BPDUs) must be delivered to all remote sites so that the customer virtual network operates properly, you can give PDUs higher priority within the service-provider network than data packets received from the same tunnel port. By default, the PDUs use the same CoS value as data packets.

Configuring Layer 2 Protocol Tunneling

Beginning in privileged EXEC mode, follow these steps to configure a port for Layer 2 protocol tunneling:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	interface <i>interface-id</i>	Enter interface configuration mode, and enter 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 switch. Valid interfaces can be physical interfaces and port-channel logical interfaces (port channels 1 to 48).
Step 3	switchport mode access or switchport mode dot1q-tunnel	Configure the interface as an access port or an 802.1Q tunnel port.
Step 4	l2protocol-tunnel [cdp stp vtp]	Enable protocol tunneling for the desired protocol. If no keyword is entered, tunneling is enabled for all three Layer 2 protocols.
Step 5	l2protocol-tunnel shutdown-threshold [cdp stp vtp] <i>value</i>	(Optional) Configure 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.
Step 6	l2protocol-tunnel drop-threshold [cdp stp vtp] <i>value</i>	(Optional) Configure 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. 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 7	exit	Return to global configuration mode.
Step 8	errdisable recovery cause l2ptguard	(Optional) Configure the recovery mechanism from a Layer 2 maximum-rate error so that the interface is re-enabled and can try again. Errdisable recovery is disabled by default; when enabled, the default time interval is 300 seconds.

	Command	Purpose
Step 9	l2protocol-tunnel cos <i>value</i>	(Optional) Configure 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 10	end	Return to privileged EXEC mode.
Step 11	show l2protocol	Display the Layer 2 tunnel ports on the switch, including the protocols configured, the thresholds, and the counters.
Step 12	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Use the **no l2protocol-tunnel [cdp | stp | vtp]** interface configuration command to disable protocol tunneling for one of the Layer 2 protocols or for all three. Use the **no l2protocol-tunnel shutdown-threshold [cdp | stp | vtp]** and the **no l2protocol-tunnel drop-threshold [cdp | stp | vtp]** commands to return the shutdown and drop thresholds to the default settings.

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


```
Switch(config)# interface fastethernet1/0/11
Switch(config-if)# l2protocol-tunnel cdp
Switch(config-if)# l2protocol-tunnel stp
Switch(config-if)# l2protocol-tunnel vtp
Switch(config-if)# l2protocol-tunnel shutdown-threshold 1500
Switch(config-if)# l2protocol-tunnel drop-threshold 1000
Switch(config-if)# exit
Switch(config)# l2protocol-tunnel cos 7
Switch(config)# end
Switch# show l2protocol
COS for Encapsulated Packets: 7
Port      Protocol Shutdown Drop      Encapsulation Decapsulation Drop
          Threshold Threshold Counter      Counter      Counter
-----
Fa1/0/11  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
```

Configuring Layer 2 Tunneling for EtherChannels

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

Configuring the SP Edge Switch

Beginning in privileged EXEC mode, follow these steps to configure a SP edge switch for Layer 2 protocol tunneling for EtherChannels:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	interface <i>interface-id</i>	Enter interface configuration mode, and enter the interface to be configured as a tunnel port. This should be the edge port in the SP network that connects to the customer switch. Valid interfaces are physical interfaces.
Step 3	switchport mode dot1q-tunnel	Configure the interface as an 802.1Q tunnel port.
Step 4	l2protocol-tunnel point-to-point [pagp lACP udld]	(Optional) Enable point-to-point protocol tunneling for the desired protocol. If no keyword is entered, tunneling is enabled for all three protocols.  Caution 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.
Step 5	l2protocol-tunnel shutdown-threshold [point-to-point [pagp lACP udld]] <i>value</i>	(Optional) Configure 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.
Step 6	l2protocol-tunnel drop-threshold [point-to-point [pagp lACP udld]] <i>value</i>	(Optional) Configure 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 7	no cdp enable	Disable CDP on the interface.
Step 8	spanning-tree bpduguard enable	Enable BPDU filtering on the interface.
Step 9	exit	Return to global configuration mode.
Step 10	errdisable recovery cause l2ptguard	(Optional) Configure the recovery mechanism from a Layer 2 maximum-rate error so that the interface is re-enabled and can try again. Errdisable recovery is disabled by default; when enabled, the default time interval is 300 seconds.
Step 11	l2protocol-tunnel cos <i>value</i>	(Optional) Configure 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 12	end	Return to privileged EXEC mode.

	Command	Purpose
Step 13	show l2protocol	Display the Layer 2 tunnel ports on the switch, including the protocols configured, the thresholds, and the counters.
Step 14	copy running-config startup-config	(Optional) Save your entries in the configuration file.

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. 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.

Configuring the Customer Switch

After configuring the SP edge switch, begin in privileged EXEC mode and follow these steps to configure a customer switch for Layer 2 protocol tunneling for EtherChannels:

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	interface <i>interface-id</i>	Enter the interface configuration mode. This should be the customer switch port.
Step 3	switchport trunk encapsulation dot1q	Set the trunking encapsulation format to 802.1Q.
Step 4	switchport mode trunk	Enable trunking on the interface.
Step 5	udld enable	Enable UDLD in normal mode on the interface.
Step 6	channel-group <i>channel-group-number</i> mode desirable	Assign the interface to a channel group, and specify desirable for the PAgP mode. For more information about configuring EtherChannels, see Chapter 34, “Configuring EtherChannels.”
Step 7	exit	Return to global configuration mode.
Step 8	interface port-channel <i>port-channel number</i>	Enter port-channel interface mode.
Step 9	shutdown	Shut down the interface.
Step 10	no shutdown	Enable the interface.
Step 11	end	Return to privileged EXEC mode.
Step 12	show l2protocol	Display the Layer 2 tunnel ports on the switch, including the protocols configured, the thresholds, and the counters.
Step 13	copy running-config startup-config	(Optional) Save your entries in the configuration file.

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

For EtherChannels, you need to configure both the SP edge switches and the customer switches for Layer 2 protocol tunneling. (See [Figure 17-6](#) on [page 17-10](#).)

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 fastethernet1/0/1
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 fastethernet1/0/2
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 fastethernet1/0/3
Switch(config-if)# switchport trunk encapsulation isl
Switch(config-if)# switchport mode trunk
```

SP edge switch 2 configuration:

```
Switch(config)# interface fastethernet1/0/1
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 fastethernet1/0/2
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 fastethernet1/0/3
Switch(config-if)# switchport trunk encapsulation isl
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 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 fastethernet1/0/1
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 fastethernet1/0/2
```

```

Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# uddld enable
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# exit
Switch(config)# interface fastethernet1/0/3
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# uddld enable
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# exit
Switch(config)# interface fastethernet1/0/4
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# uddld 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 and Maintaining Tunneling Status

Table 17-2 shows the privileged EXEC commands for monitoring and maintaining 802.1Q and Layer 2 protocol tunneling.

Table 17-2 Commands for Monitoring and Maintaining Tunneling

Command	Purpose
<code>clear l2protocol-tunnel counters</code>	Clear the protocol counters on Layer 2 protocol tunneling ports.
<code>show dot1q-tunnel</code>	Display 802.1Q tunnel ports on the switch.
<code>show dot1q-tunnel interface <i>interface-id</i></code>	Verify if a specific interface is a tunnel port.
<code>show l2protocol-tunnel</code>	Display information about Layer 2 protocol tunneling ports.
<code>show errdisable recovery</code>	Verify if the recovery timer from a Layer 2 protocol-tunnel error disable state is enabled.
<code>show l2protocol-tunnel interface <i>interface-id</i></code>	Display information about a specific Layer 2 protocol tunneling port.
<code>show l2protocol-tunnel summary</code>	Display only Layer 2 protocol summary information.
<code>show vlan dot1q tag native</code>	Display the status of native VLAN tagging on the switch.

For detailed information about these displays, see the command reference for this release.