Configuring 802.1Q Tunneling, VLAN Mapping, 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 who are required to maintain the VLAN and Layer 2 protocol configurations of each customer without impacting the traffic of other customers. This chapter describes how to configure 802.1Q, Layer 2 protocol tunneling, and VLAN mapping (or VLAN ID translation) on a Catalyst 4500 series switch.

This chapter contains these sections:

Note   For complete syntax and usage information for the switch commands used in this chapter, see the Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch.

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference* , you can locate it in the Cisco IOS Master Command List, All Releases.

About 802.1Q Tunneling

The VLAN ranges required by different customers in the same service provider network might overlap, and customer traffic through the infrastructure might be mixed. Assigning a unique range of VLAN IDs to each customer restricts customer configurations and could easily exceed the VLAN limit (4096) of the 802.1Q specification.

802.1Q tunneling enables service providers to use a single VLAN to support customers who have multiple VLANs, while preserving customer VLAN IDs and keeping traffic in different customer VLANs segregated.

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 service provider VLAN ID supports VLANs of all the customers.

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 (Figure 30-1).

Figure 30-1   802.1Q Tunnel Ports in a Service Provider Network
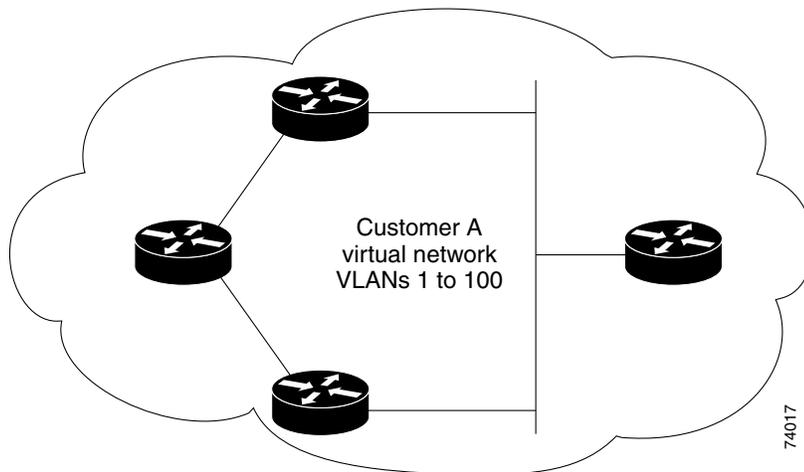
Packets coming from the customer trunk port into the tunnel port on the service provider edge switch are normally 802.1Q-tagged with the appropriate VLAN ID. When the tagged packets exit the trunk port into the service provider network, they are encapsulated with another layer of an 802.1Q tag (called 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. Packets entering the service provider network are double-tagged, with the 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 metro 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 30-2 shows the tag structures of the Ethernet packets starting with the original, or normal, frame.

Figure 30-2   Original (Normal), 802.1Q, and Double-Tagged Ethernet Packet Formats
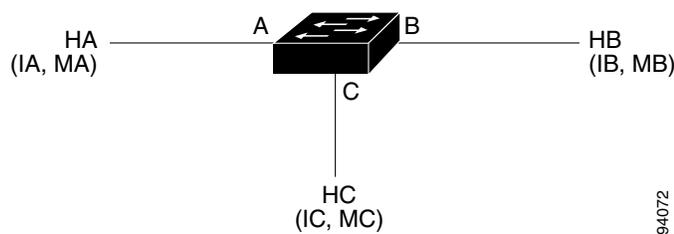
When the packet enters the trunk port of the service provider egress switch, the metro tag is again stripped as the switch processes the packet. However, the metro tag is not added when the packet is sent out the tunnel port on the edge switch into the customer network. The packet is sent as a normal 802.1Q-tagged frame to preserve the original VLAN numbers in the customer network.

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.)

In Figure 30-1, 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 metro tag containing VLAN ID 30 or 40, appropriately, and the inner tag containing the original customer VLAN number, for example, VLAN 100. Even if Customers A and B both have VLAN 100 in their networks, the traffic remains segregated within the service provider network because the metro 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.

Configuring 802.1Q Tunneling

These sections describe 802.1Q tunneling configuration:

- 802.1Q Tunneling Configuration Guidelines, page 30-3
- 802.1Q Tunneling and Other Features, page 30-5
- Configuring an 802.1Q Tunneling Port, page 30-5

Note   By default, 802.1Q tunneling is disabled because the default switch port 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 asymmetrical links for traffic going through a tunnel and should dedicate one VLAN for each tunnel. You should also be aware of configuration requirements for native VLANs and maximum transmission units (MTUs). For more information about MTUs, see the "System MTU" section on page 30-5.

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 is not tagged on the 802.1Q sending trunk port (Figure 30-3).

VLAN 40 is configured as the native VLAN for the 802.1Q trunk port from Customer A at the ingress edge switch in the service provider network (Switch 2). Switch 1 of Customer A sends a tagged packet on VLAN 30 to the ingress tunnel port of Switch 2 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 3) and is misdirected through the egress switch tunnel port to Customer B.

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 **switchport trunk native vlan tag** per-port command and 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 ensures that all packets exiting the trunk are tagged and prevents the reception of untagged packets on the trunk port.

- 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 30-3   Potential Problem with 802.1Q Tunneling and Native VLANs
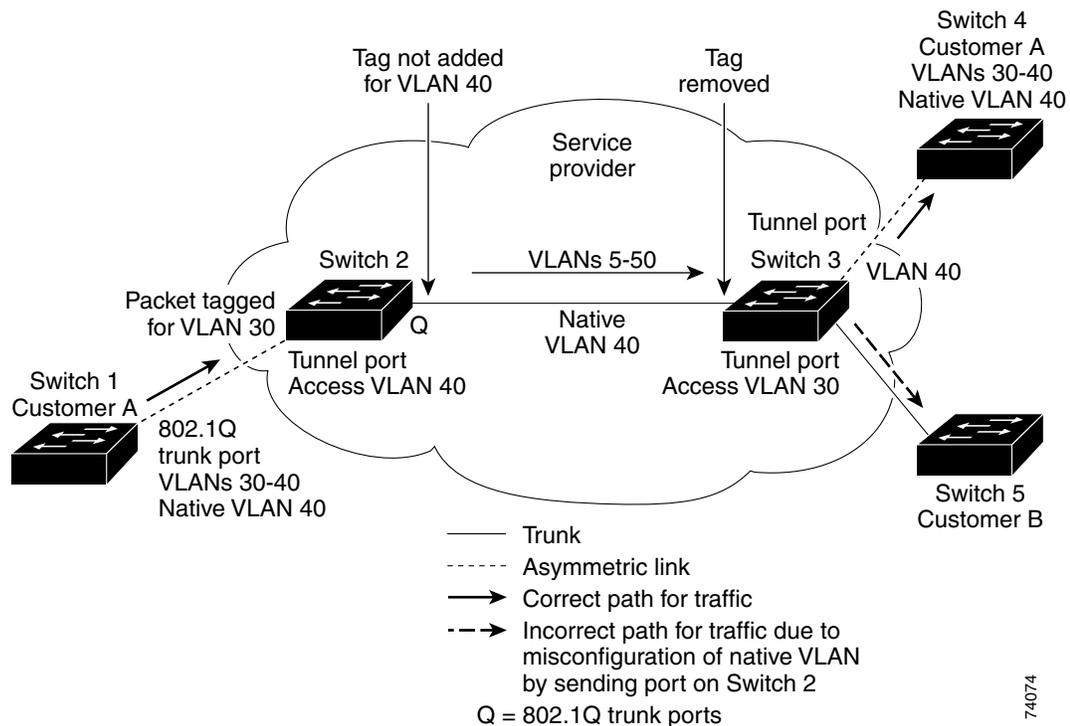
System MTU

The default system MTU for traffic on the switch is 1500 bytes. You can configure the switch to support larger frames by using the **system mtu** 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 larger frames by increasing the switch system MTU size to at least 1504 bytes. The maximum allowable system MTU for Catalyst 4500 Gigabit Ethernet switches is 9198 bytes; the maximum system MTU for Fast Ethernet switches is 1552 bytes.

802.1Q Tunneling and Other Features

Although 802.1Q tunneling works well for Layer 2 packet switching, incompatibilities exist 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. Customers can access the Internet through the native VLAN. If this access is not needed, you should not configure SVIs on VLANs that include 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.
- 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

To configure a port as an 802.1Q tunnel port, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters global configuration mode.
Step 2	Switch(config)# **interface** *interface-id*	Enters interface configuration mode and 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 64).
Step 3	Switch(config-if)# **switchport access vlan** *vlan-id*	Specifies the default VLAN, which is used if the interface stops trunking. This VLAN ID is specific to the particular customer.

	Command	Purpose
Step 4	Switch(config-if)# **switchport mode dot1q-tunnel**	Sets the interface as an 802.1Q tunnel port.
Step 5	Switch(config-if)# **exit**	Returns to global configuration mode.
Step 6	Switch(config)# **vlan dot1q tag native**	(Optional) Sets 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	Switch(config)# **end**	Returns to privileged EXEC mode.
Step 8	Switch# **show dot1q-tunnel**	Displays the tunnel ports on the switch.
Step 9	Switch# **show vlan dot1q tag native**	Displays 802.1Q native-VLAN tagging status.
Step 10	Switch# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Use the **no vlan dot1q tag native** global command and the **no switchport mode dot1q-tunnel** interface configuration command to return the port to the default state of dynamic auto. 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 2/7 is VLAN 22.

```
Switch(config)# interface gigabitethernet2/7
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 gigabitethernet2/7
Port
-----
LAN Port(s)
-----
Gi2/7
Switch# show vlan dot1q tag native
dot1q native vlan tagging is enabled globally
```

About VLAN Mapping

✎

Note   WS-C4948-10GE does not support VLAN mapping. VLAN mapping is only supported on Supervisor Engine 6-E and later engines.

In a typical deployment of VLAN mapping, you want the service provider to provide a transparent switching infrastructure that treats customers' switches at the remote location as a part of the local site. This allows customers to use the same VLAN ID space and run Layer 2 control protocols seamlessly across the provider network. In such scenarios, we recommend that service providers do not impose their VLAN IDs on their customers.

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

Service providers's internal assignments might conflict with a customer's VLAN. To isolate customer traffic, a service provider could decide to map a specific VLAN into another one while the traffic is in its cloud.

Deployment Example

In Figure 30-4, the service provider provides Layer 2 VPN service to two different customers, A and B. The service provider separates the data and control traffic between the two customers and from the providers' own control traffic. The service provider network must also be transparent to the customer edge devices.

Figure 30-4   Layer 2 VPN Deployment
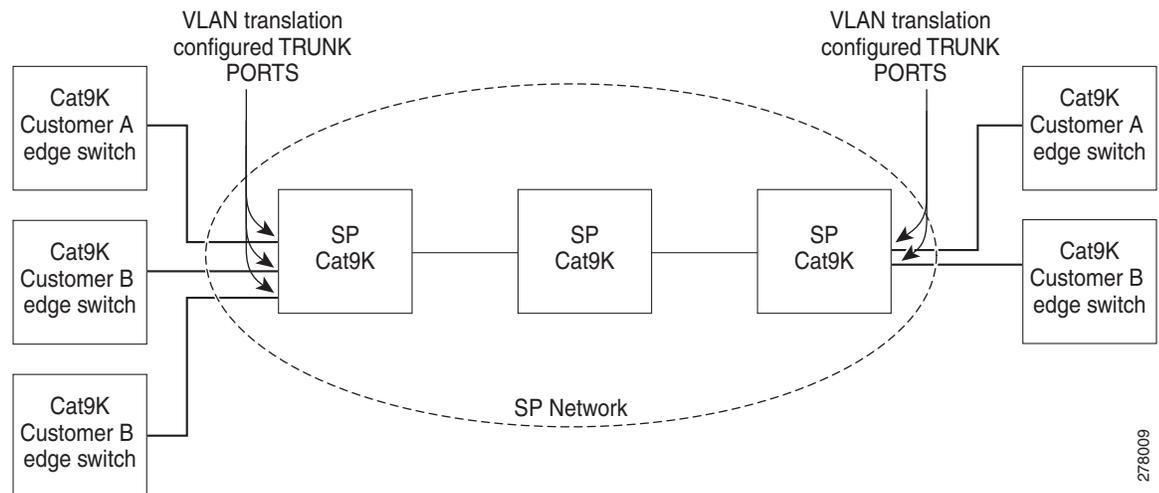
All forwarding operations on the Catalyst 4500 series switch are performed using S-VLAN and not C-VLAN information because the VLAN ID is mapped to the S-VLAN on ingress.

Note   When you configure features on a port configured for VLAN mapping, you always use the S-VLAN rather than the customer VLAN-ID (C-VLAN).

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

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

- One-to-one VLAN mapping occurs at the ingress and egress of the port and maps the customer C-VLAN ID in the 802.1Q tag to the service-provider S-VLAN ID. You can also specify that packets with all other Vlan Ids are dropped. See the "One-to-One Mapping" section on page 30-10.

- Traditional 802.1Q tunneling (QinQ) performs all-to-one bundling of C-VLAN IDs to a single S-VLAN ID for the port. The S-VLAN is added to the incoming unmodified C-VLAN. You can configure the UNI as an 802.1Q tunnel port for traditional QinQ, or you can configure selective QinQ on trunk ports for a more flexible implementation. Mapping takes place at ingress and egress of the port. All packets on the port are bundled into the specified S-VLAN. See the "Traditional Q-in-Q on a Trunk Port" section on page 30-11.

- Selective QinQ maps the specified customer VLANs entering the UNI to the specified S-VLAN ID. The S-VLAN is added to the incoming unmodified C-VLAN. You can also specify whether traffic carrying all other customer VLAN IDs should be dropped or not. See the "Selective Q-in-Q on a Trunk Port" section on page 30-12.

Note   Untagged packets enter the switch on the trunk native VLAN and are not mapped.

For quality of service (QoS), the switch supports flexible mapping between C-CoS or C-DSCP and S-CoS, and maps the inner CoS to the outer CoS for traffic with traditional QinQ or selective QinQ VLAN mapping.

Mapping Customer VLANs to Service-Provider VLANs

Figure 30-5 shows a topology where a customer uses the same VLANs in multiple sites on different sides of a service-provider network. You map the customer VLAN IDs to service-provider VLAN IDs for packet travel across the service-provider backbone. The customer VLAN IDs are retrieved at the other side of the service-provider backbone for use in the other customer site. Configure the same set of VLAN mappings at a customer-connected port on each side of the service-provider network.

The examples following the configuration steps illustrate how to use one-to-one mapping, traditional QinQ, or selective QinQ to map customer VLANs 1 to 5 to service-provider VLANs.

Figure 30-5   Mapping Customer VLANs
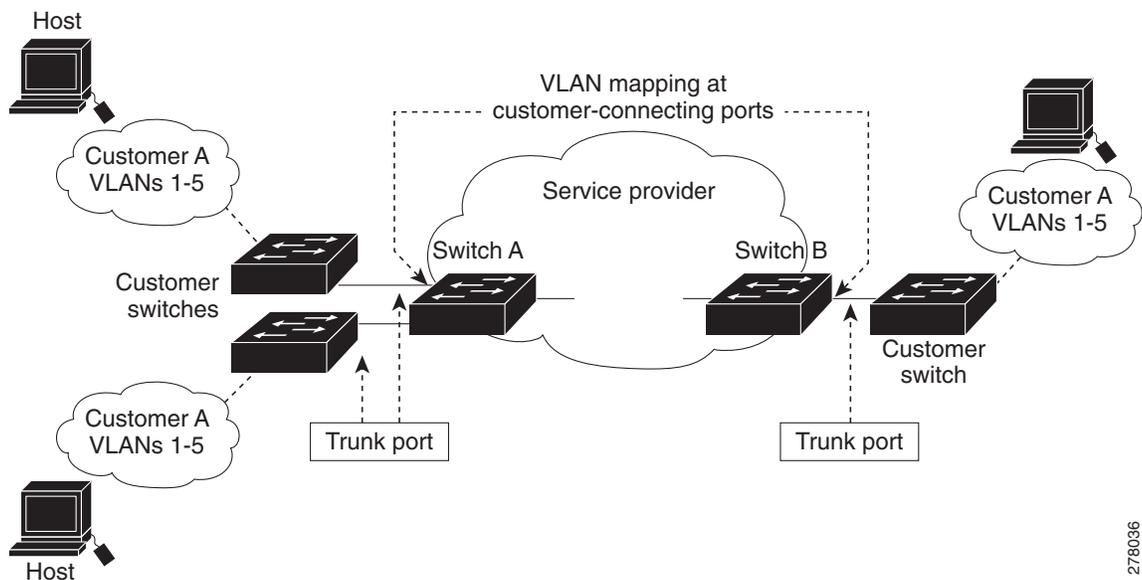
Configuring VLAN Mapping

Default VLAN Mapping Configuration

By default, no VLAN mapping is configured.

VLAN Mapping Configuration Guidelines

Guidelines include the following:

- Traditional QinQ uses 802.1Q tunnel ports; you configure one-to-one VLAN mapping and selective QinQ on 802.1Q trunk ports.

- To avoid mixing customer traffic, when you configure traditional Q-in-Q on a trunk port, you should configure the service provider S-VLAN ID as an allowed VLAN on the trunk port.

- When you configure VLAN mapping on an EtherChannel, the mapping applies to all ports in the port channel.

- You cannot configure encapsulation replicate on a SPAN destination port if the source port is configured as a tunnel port or has a 1-to-2 mapping configured. Encapsulation replicate is supported with 1-to-1 VLAN mapping.

- When configuring IEEE 802.1Q tunneling on an edge switch, 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 switches, the native VLANs of the IEEE 802.1Q trunks must not match any native VLAN of the nontrunking (tunneling) port on the same switch. it is because traffic on the native VLAN is not tagged on the IEEE 802.1Q sending trunk port.

- Ensure that the native VLAN for an IEEE 802.1Q trunk is the same on both ends of the trunk link. If the native VLAN on one end of the trunk is different from the native VLAN on the other end, spanning-tree loops might result.

- Layer 2 protocol tunneling must be configured for CDP, VTP, LLDP, or your switch detects the SP switches, which is not desirable:

```
interface GigabitEthernet1/23
switchport mode trunk
switchport vlan mapping 1 dot1q-tunnel 311
switchport vlan mapping 31 dot1q-tunnel 311
l2protocol-tunnel cdp
l2protocol-tunnel lldp
l2protocol-tunnel vtp
```

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

```
Current configuration : 153 bytes
!
interface FastEthernet9/1
```

```
                    switchport trunk native vlan 40
                    switchport mode trunk
                    switchport vlan mapping 10 20
                    spanning-tree bpdufilter enable
                  end
```

- If you need to merge CVLAN and SVLAN spanning tree topology, you do not need to configure **spanning-tree bpdufilter enable**.

- To ensure consistent operation, do not use a native VLAN for translation.

Configuring VLAN Mapping

The following procedures show how to configure each type of VLAN mapping on trunk ports. To verify your configuration, enter either the **show interfaces** *interface-id* **vlan mapping** or the **show vlan mapping** privileged EXEC command. See the "Monitoring and Maintaining Tunneling Status" section on page 30-23 for the syntax of these commands. For more information about all commands in this section, see the *Catalyst 4500 Series Switch Software Command Reference* for this release.

The following VLAN mapping types are discussed:

- One-to-One Mapping, page 30-10
- Traditional Q-in-Q on a Trunk Port, page 30-11
- Selective Q-in-Q on a Trunk Port, page 30-12

One-to-One Mapping

To configure one-to-one VLAN mapping to map a customer VLAN ID to a service-provider VLAN ID, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters global configuration mode.
Step 2	Switch(config)# **interface** *interface-id*	Enters interface configuration mode for the interface connected to the service-provider network. You can enter a physical interface or an EtherChannel port channel.
Step 3	Switch(config-if)# **switchport mode trunk**	Configures the interface as a trunk port.
Step 4	Switch(config-if)# **switchport vlan mapping** *vlan-id translated-id*	Enters the VLAN IDs to be mapped: • *vlan-id*—the customer VLAN ID (C-VLAN) entering the switch from the customer network. The range is from 1 to 4094. • *translated-id*—the assigned service-provider VLAN ID (S-VLAN). The range is from 1 to 4094. **Note** Packets with unconfigured vlan_ids are dropped.
Step 5	Switch# **end**	Returns to privileged EXEC mode.
Step 6	Switch# **show vlan mapping**	Verifies the configuration.
Step 7	Switch# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Use the **no switchport vlan mapping** *vlan-id translated-id* command to remove the VLAN mapping information. Entering the **no switchport vlan mapping all** command deletes all mapping configurations.

This example shows how to map VLAN IDs 1 to 5 in the customer network to VLANs 101 to 105 in the service-provider network (Figure 30-5). You configure the same VLAN mapping commands for a port in Switch A and Switch B; the traffic on all other VLAN IDs is dropped.

```
Switch(config)# interface gigabiethernet0/1
Switch(config-if)# switchport vlan mapping 1 101
Switch(config-if)# switchport vlan mapping 2 102
Switch(config-if)# switchport vlan mapping 3 103
Switch(config-if)# switchport vlan mapping 4 104
Switch(config-if)# switchport vlan mapping 4 105
Switch(config-if)# exit
```

In the previous example, at the ingress of the service-provider network, VLAN IDs 1 to 5 in the customer network are mapped to VLANs 101 to 105, in the service provider network. At the egress of the service provider network, VLANs 101 to 105 in the service provider network are mapped to VLAN IDs 1 to 5, in the customer network.

✎
Note   Packets with unconfigured vlan_ids are dropped.

Traditional Q-in-Q on a Trunk Port

To configure VLAN mapping for traditional Q-in-Q on a trunk port or tunneling by default, perform the following task:

✎
Note   Configuring tunneling by default bundles all packets on the port into the configured S-VLAN.

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters global configuration mode.
Step 2	Switch(config)# **interface** *interface-id*	Enters interface configuration mode for the interface connected to the service provider network. You can enter a physical interface or an EtherChannel port channel.
Step 3	Switch(config-if)# **switchport mode trunk**	Configures the interface as a trunk port.
Step 4	Switch(config-if)# **switchport trunk allowed vlan** *vlan-id*	Configures the outer VLAN of the service provider network (S-VLAN) to to be allowed on the interface. This should be the same outer VLAN ID entered in the next step.
Step 5	Switch(config-if)# **end**	Returns to privileged EXEC mode.
Step 6	Switch# **show interfaces** *interface-id* **vlan mapping**	Verifies the configuration.
Step 7	Switch# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Entering the **no switchport vlan mapping all** command deletes all mapping configurations.

Selective Q-in-Q on a Trunk Port

To configure VLAN mapping for selective Q-in-Q on a trunk port, perform this task:

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

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters global configuration mode.
Step 2	Switch(config)# **interface** *interface-id*	Enters interface configuration mode for the interface connected to the service provider network. You can enter a physical interface or an EtherChannel port channel.
Step 3	Switch(config-if)# **switchport mode trunk**	Configure the interface as a trunk port.
Step 4	Switch(config-if)# **switchport vlan mapping** *vlan-id* **dot1q-tunnel** *outer vlan-id*	Enters the VLAN IDs to be mapped: • *vlan-id*—The customer VLAN ID (C-VLAN) entering the switch from the customer network. The range is from 1 to 4094. You can enter a string of VLAN-IDs. • *outer-vlan-id*—The outer VLAN ID (S-VLAN) of the service provider network. The range is from 1 to 4094.
Step 5	Switch(config-if)# **switchport vlan mapping default drop**	Specifies that all packets on the port are dropped if they do not match the mapped VLANs (Step 4). By default, packets that do not match, are dropped. If the packets (that do not match) should not be dropped, enter the **no** version of the command.
Step 6	Switch(config-if)# **end**	Returns to privileged EXEC mode.
Step 7	Switch# **show interfaces** *interface-id* **vlan mapping**	Verifies the configuration.
Step 8	Switch# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Use the **no switchport vlan mapping** *vlan-id* **dot1q-tunnel** *outer vlan-id* command to remove the VLAN mapping configuration. Entering the **no switchport vlan mapping all** command deletes all mapping configurations.

This example shows how to configure selective QinQ mapping on the port so that traffic with a C-VLAN ID of 1 to 5 enters the switch with an S-VLAN ID of 100. The traffic of any other VLAN ID is dropped.

```
Switch(config)# interface gigabiethernet0/1
Switch(config-if)# switchport vlan mapping 1-5 dot1q-tunnel 100
Switch(config-if)# switchport vlan mapping default drop
Switch(config-if)# exit
```

This example shows how to configure selective QinQ mapping on the port so that traffic with a C-VLAN ID of 1 to 5 enters the switch with an S-VLAN ID of 100. The traffic of any other VLAN ID is allowed.

```
Switch(config)# interface gigabiethernet0/1
Switch(config-if)# switchport vlan mapping 1-5 dot1q-tunnel 100
Switch(config-if)# no switchport vlan mapping default drop
Switch(config-if)# exit
```

About Layer 2 Protocol Tunneling

Note   IPsec VPN is supported for control plane traffic protection on the management port, and must be used for management purposes only.

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 and 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.

Layer 2 protocol tunneling can enabled on trunk, access and tunnel ports. If protocol tunneling is not enabled, 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.

As an example, Customer A in Figure 30-6 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 A's Site 1 builds a spanning tree on the switches at that site without considering convergence parameters based on Customer A's switch in Site 2. Figure 30-6 shows one possible spanning tree topology.

Figure 30-6   Layer 2 Protocol Tunneling

Customer A Site 1
VLANs 1 to 100

Customer A Site 2
VLANs 1 to 100

VLAN 30

Service
provider

VLAN 30

VLAN 30

Switch 1

Trunk
ports

Trunk
ports

Switch 3

Switch 1

Switch 2

Switch 4

Trunk
ports

Trunk
ports

VLAN 40

VLAN 40

Customer B Site 1
VLANs 1 to 200

—— Trunk
----- Asymmetric link

Customer B Site 2
VLANs 1 to 200

74073
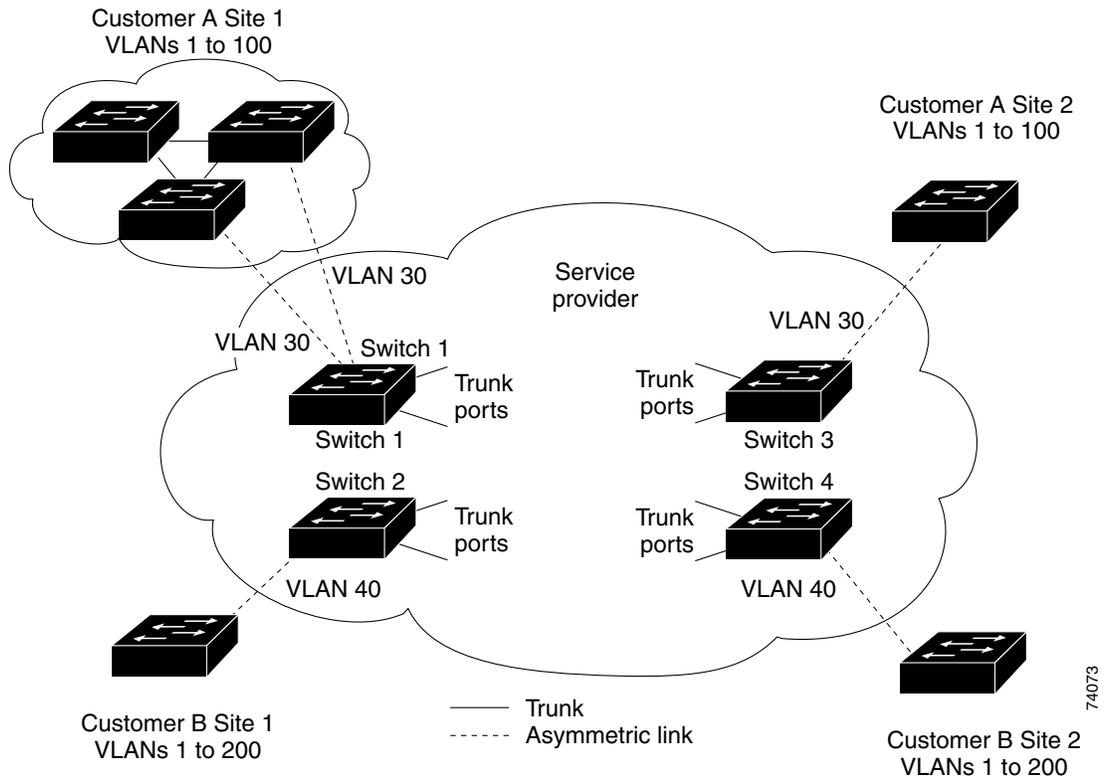
Figure 30-7   Layer 2 Network Topology without Proper Convergence

Customer A
virtual network
VLANs 1 to 100

74017
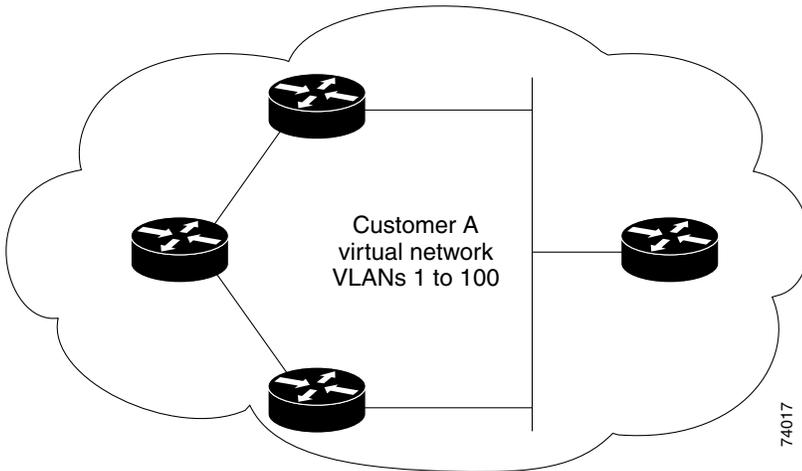
Configuring Layer 2 Protocol Tunneling

You can enable Layer 2 protocol tunneling (by protocol) on access ports, tunnel ports, or trunk 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 or normal trunk ports can be connected to customer 802.1Q trunk ports. Edge-switch access ports are connected to customer access ports.

When the Layer 2 PDUs that entered the service provider inbound edge switch through the tunnel port or the access port exit through its 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. This section includes these topics. The Layer 2 PDUs remain intact and are delivered across the service provider network to the other side of the customer network.

Figure 30-6 shows Customer A and Customer B in access VLANs 30 and 40. 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 B 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 4, the metro 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 B 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 ports on the customer switch. The encapsulation and de-encapsulation process is the same as described in the previous paragraph, except that the packets are not double-tagged in the service provider network. The single tag is the customer-specific access VLAN tag.

In an SP (service-provider) 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 the following figure (Layer 2 Protocol Tunneling for EtherChannels), 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.

Figure 30-8   *Layer 2 Protocol Tunneling for EtherChannels*
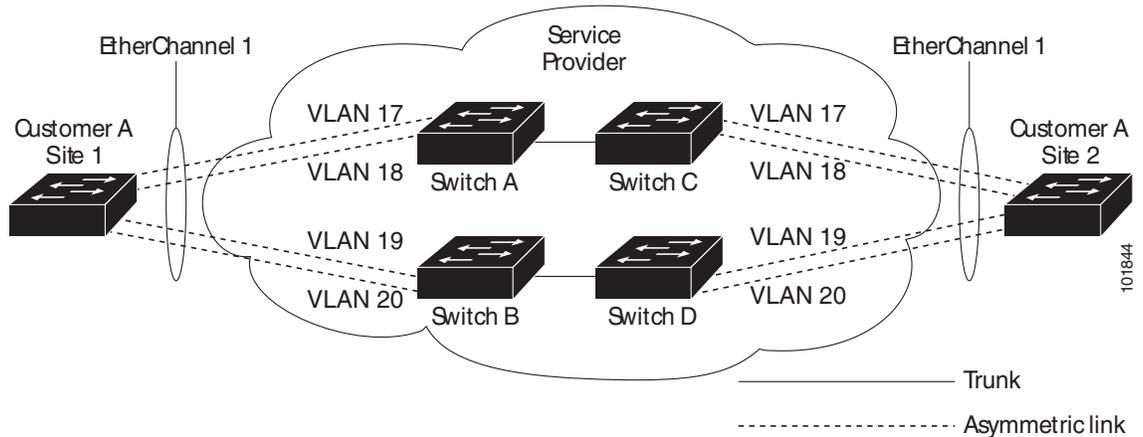
This section contains the following subsections:

Default Layer 2 Protocol Tunneling Configuration

Table 30-1 shows the default configuration for Layer 2 protocol tunneling.

Table 30-1   *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 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, access ports or trunk ports.

- Dynamic Trunking Protocol (DTP) is not compatible with Layer 2 protocol tunneling because you must manually configure asymmetric links with tunnel ports and trunk ports.

- 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 on a port with Layer 2 tunneling enabled, the 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 reenable 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 Layer 2 protocol tunneling ports. CDP packets are not forwarded from Layer 2 protocol tunneling 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 the BPDU rate by using QoS ACLs and policy maps on a Layer 2 protocol tunneling 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.

> **Note**   If Layer 2 protocol tunneling is not configured on a system, Layer 2 protocol tunneling packets are handled as data packets and this situation does not apply.

Configuring Layer 2 Tunneling

To configure a port for Layer 2 protocol tunneling, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters global configuration mode.
Step 2	Switch(config)# **interface** *interface-id*	Enters 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 64).
Step 3	Switch(config-if)# **switchport mode access** or Switch(config-if)# **switchport mode dot1q-tunnel** or Switch(config-if)# **switchport mode trunk**	Configures the interface as an access port, an 802.1Q tunnel port or a trunk port.
Step 4	Switch(config-if)# **l2protocol-tunnel** [**cdp** | **point-to-point** | **stp** | **stp** | **vtp**]	Enables protocol tunneling for the desired protocol. If no keyword is entered, tunneling is enabled for all three Layer 2 protocols.

	Command	Purpose
Step 5	`Switch(config-if)#` **l2protocol-tunnel shutdown-threshold** [**cdp** | **point-to-point** | **stp** | **vtp**] *value*	(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.
Step 6	`Switch(config-if)#` **l2protocol-tunnel drop-threshold** [**cdp** | **point-to-point** | **stp** | **vtp**] *value*	(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 7	`Switch(config-if)#` **exit**	Returns to global configuration mode.
Step 8	`Switch(config)#` **errdisable recovery cause l2ptguard**	(Optional) Configures the recovery method 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.
Step 9	`Switch(config)#` **l2protocol-tunnel cos** *value*	(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 10	`Switch(config)#` **end**	Returns to privileged EXEC mode.
Step 11	`Switch#` **show l2protocol**	Displays the Layer 2 tunnel ports on the switch, including the protocols configured, the thresholds, and the counters.
Step 12	`Switch#` **copy running-config startup-config**	(Optional) Saves 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 on an 802.1Q tunnel port for CDP, STP, VTP, and LLDP and how to verify the configuration:

```
Switch(config)# interface FastEthernet 1/1/11
Switch(config-if)# switchport mode dot1q-tunnel
Switch(config-if)# l2protocol-tunnel cdp
Switch(config-if)# l2protocol-tunnel stp
Switch(config-if)# l2protocol-tunnel vtp
Switch(config-if)# l2protocol-tunnel lldp
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
```

```
Drop Threshold for Encapsulated Packets: 0
```

Port	Protocol	Shutdown Threshold	Drop Threshold	Encaps Counter	Decaps Counter	Drop Counter
Gi1/1/11	cdp	1500	1000	2288	2282	0
	lldp	1500	1000	0	0	0
	stp	1500	1000	116	13	0
	vtp	1500	1000	3	67	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. This section includes the following tasks:

Configuring the SP Edge Switch

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters the global configuration mode.
Step 2	Switch(config)# **interface** *interface-id*	Enters interface configuration mode. 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.
Step 3	Switch(config-if)# **switchport mode dot1q-tunnel**	Configures the interface as an 802.1Q tunnel port
Step 4	Switch(config-if)# **l2protocol-tunnel point-to-point [pagp | lacp | udld]**	(Optional) Enables 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	Switch(config-if)# **l2protocol-tunnel shutdown-threshold [point-to-point [pagp | lacp | udld]]** *value*	(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	Purpose
Step 6	Switch(config-if)# **l2protocol-tunnel drop-threshold [point-to-point [pagp	lacp	udld]]** *value*	(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 7	Switch(config-if)# **no cdp enable**	Disables CDP on the interface.
Step 8	Switch(config-if)# **spanning-tree bpdufilter**	Enables BPDU filtering on the interface.
Step 9	Switch(config)# **exit**	Returns to global configuration mode.
Step 10	Switch(config)# **errdisable recovery cause l2ptguard**	(Optional) Configures 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	Switch(config)# **l2protocol-tunnel cos value**	(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 12	Switch(config)# **end**	Returns to privileged EXEC mode.
Step 13	Switch# **show l2protocol**	Displays the Layer 2 tunnel ports on the switch, including the protocols configured, the thresholds, and the counters.
Step 14	Switch# **copy running-config startup-config**	(Optional) Saves 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 their default settings.

This example shows how to configure the SP edge switch:

```
Switch(config)# interface gigabitEthernet 1/1/11
Switch(config-if)# switchport mode dot1q-tunnel
Switch(config-if)# switchport access vlan 10
Switch(config-if)# l2protocol-tunnel point-to-point
Switch(config-if)# l2protocol-tunnel shutdown-threshold point-to-point 3000
Switch(config-if)# l2protocol-tunnel drop-threshold point-to-point 2500
Switch(config-if)# exit
Switch# show l2protocol-tunnel
COS for Encapsulated Packets: 5
Drop Threshold for Encapsulated Packets: 0

Port                Protocol    Shutdown  Drop      Encaps    Decaps    Drop
                                Threshold Threshold Counter   Counter   Counter
------------------- ----------- --------- --------- --------- --------- ---------
Gi1/1/11            ---         ----      ----      ----      ----      ----
                    ---         ----      ----      ----      ----      ----
                    ---         ----      ----      ----      ----      ----
                    ---         ----      ----      ----      ----      ----
```

```
                              pagp        3000      2500          0           0           0
                              lacp        3000      2500          0           0           0
                              udld        3000      2500          0           0           0
```

Configuring the Customer Switch

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters the global configuration mode.
Step 2	Switch(config)# **interface** *interface-id*	Enters interface configuration mode. Enter the interface to be configured as a tunnel port.
Step 3	Switch(config-if)# **switchport mode trunk**	Enables trunking on the interface.
Step 4	Switch(config-if)# **udld enable**	Enables UDLD in normal mode on the interface
Step 5	Switch(config-if)# **channel-group** *channel-group-number* **mode desirable**	Assigns the interface to a channel group, and specifies desirable for the PAgP mode.
Step 6	Switch(config-if)# **exit**	Returns to global configuration mode.
Step 7	Switch(config)# **interface port-channel** port-channel number	Enters port-channel interface mode.
Step 8	Switch(config-if)# **shutdown**	Shuts down the interface.
Step 9	Switch(config-if)# **no shutdown**	Enables the interface.
Step 10	Switch(config-if)# **end**	Returns to global configuration mode.
Step 11	Switch# **show l2protocol**	Displays the Layer 2 tunnel ports on the switch, including the protocols configured, the thresholds, and the counters.
Step 12	Switch# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

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

This example shows you how to configure a customer switch:

```
Switch(config)# interface gigabitEthernet 2/14
Switch(config-if)# switchport mode trunk
Switch(config-if)# udld port
Switch(config-if)# channel-group 10 mode desirable
Switch(config-if)# exit
```

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, Gigabit Ethernet interfaces 1/1/11 and 1/1/12 are point-to-point tunnel ports with PAgP and UDLD enabled, the drop threshold is 1000, and Fast Ethernet interface 1/1/13 is a trunk port.

SP edge switch 1 configuration:

```
Switch(config)# interface gigabitEthernet 1/1/11
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 gigabitEthernet 1/1/12
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 gigabitEthernet 1/1/13
Switch(config-if)# switchport mode trunk
```

SP edge switch 2 configuration:

```
Switch(config)# interface gigabitEthernet 1/1/11
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 gigabitEthernet 1/1/12
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 gigabitEthernet 1/1/13
Switch(config-if)# switchport mode trunk
```

This example shows how to configure the customer switch at Site 1. Gigabit Ethernet interfaces 1/1, 1/2, 1/3, and 1/4 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 EtherChannel configuration. See also Figure 30-8.

```
Switch(config)# interface GigabitEthernet1/1
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 GigabitEthernet1/2
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 GigabitEthernet1/3
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 GigabitEthernet1/4
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 and Maintaining Tunneling Status

Table 30-2 shows the commands for monitoring and maintaining 802.1Q and Layer 2 protocol tunneling.

Table 30-2   Commands for Monitoring and Maintaining Tunneling

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

Note   With Cisco IOS Release 12.2(20)EW, the BPDU filtering configuration for both dot1q and Layer 2 protocol tunneling is no longer visible in the running configuration as spanning-tree bpdufilter enable. The configuration is visible in the output of the **show spanning tree int detail** command.

```
Switch# show spann int f6/1 detail
 Port 321 (FastEthernet6/1) of VLAN0001 is listening
   Port path cost 19, Port priority 128, Port Identifier 128.321.
   Designated root has priority 32768, address 0008.e341.4600
   Designated bridge has priority 32768, address 0008.e341.4600
   Designated port id is 128.321, designated path cost 0
   Timers: message age 0, forward delay 2, hold 0
   Number of transitions to forwarding state: 0
   Link type is point-to-point by default
   ** Bpdu filter is enabled internally **
   BPDU: sent 0, received 0
```