CHAPTER 13

Configuring IEEE 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 Metro switch supports 802.1Q tunneling and Layer 2 protocol tunneling, as well as VLAN mapping (VLAN-ID translation).

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

This chapter contains these sections:

Note For information about VPNs used with multiprotocol label switching (MPLS), see Chapter 30, "Configuring MPLS and EoMPLS."

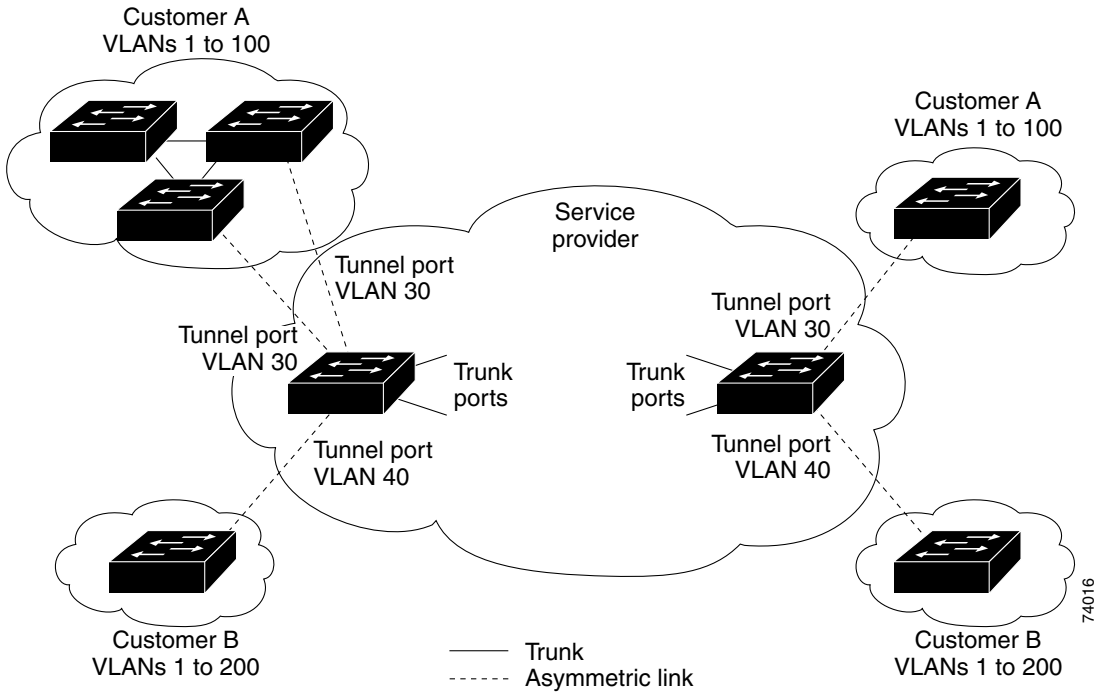
Understanding 802.1Q Tunneling

Service-provider business customers 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 infrastructure, even when they appear to be on the same VLAN. Using 802.1Q tunneling expands VLAN space by using a VLAN-in-VLAN hierarchy and tagging the tagged packets. A port configured to support 802.1Q tunneling is called a *tunnel port*. When you configure tunneling, you assign a tunnel port to a VLAN that is dedicated to tunneling. Each customer requires a separate 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 come from an 802.1Q trunk port on the customer device and into a tunnel port on the service-provider edge switch. The link between the customer device and the edge switch is an asymmetric link because one end is configured as an 802.1Q trunk port and the other end is configured as a tunnel port. You assign the tunnel port interface to an access VLAN ID that is unique to each customer. See Figure 13-1.

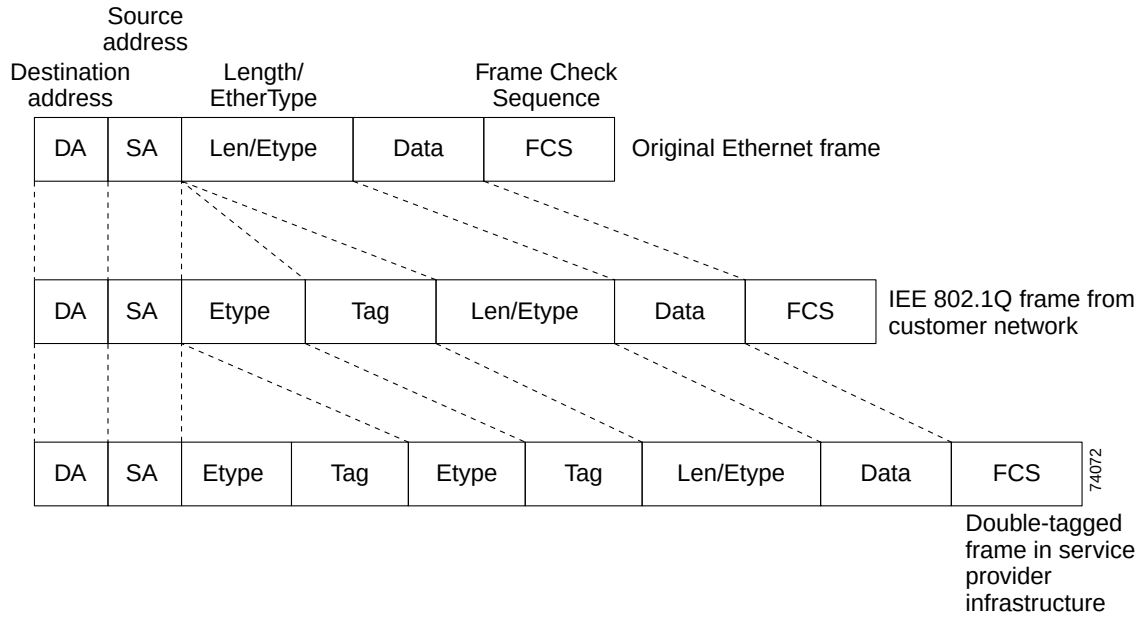
Figure 13-1 802.1Q Tunnel Ports in a Service-Provider Network



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 tagged packets remain intact inside the switch and, when they exit the trunk port into the service-provider network, 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 802.1Q tag from the customer is preserved in the encapsulated packet. Therefore, packets entering the service-provider infrastructure are double-tagged, with the outer tag containing the customer's access VLAN ID, and the inner VLAN ID being the VLAN 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 packet is processed inside the switch. When the packet exits another trunk port on the same core switch, the same metro tag is again added to the packet. Figure 13-2 shows the structure of the double-tagged packet.

Figure 13-2 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 packet is processed internally on the switch. However, the metro tag is not added when it is sent out the tunnel port on the edge switch into the customer network, and the packet is sent as a normal 802.1Q-tagged frame to preserve the original VLAN numbers in the customer network.

In Figure 13-1, 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. With 802.1Q tunneling, 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 number s 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 the traffic coming from a customer network is not tagged (native VLAN frames), these packets are bridged or routed as if they were 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).

The switch supports intelligent 802.1Q tunneling QoS, or the ability to copy the inner CoS value to the outer CoS value. For more information, see the "Configuring the Trust State on Ports Within the QoS Domain" section on page 26-42 and the "Configuring the CoS Value for an Interface" section on page 26-45.

Configuring 802.1Q Tunneling

This section includes this information about configuring 802.1Q tunneling:

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 maximum transmission units (MTUs) are explained in the next sections.

Native VLANs

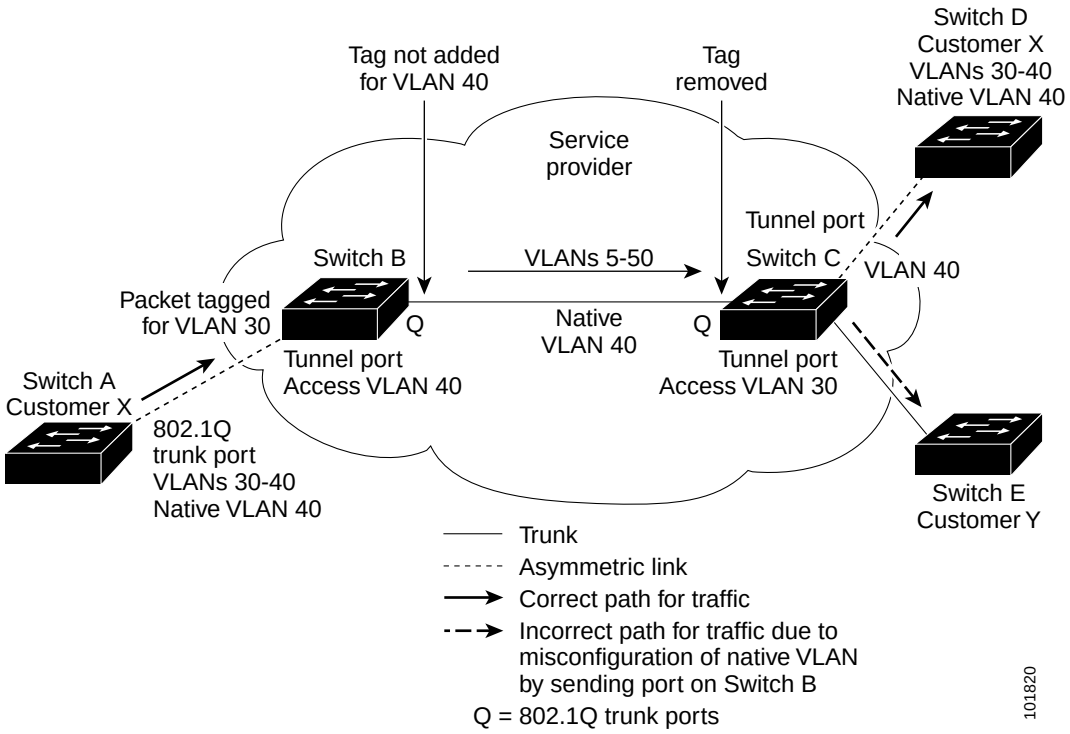
When configuring 802.1Q tunneling on an edge switch, you must use 802.1Q trunk ports for sending out packets into the service-provider network. However, packets going through the core of the service-provider network might be carried through 802.1Q trunks, ISL trunks, or nontrunking links. When 802.1Q 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 transmitting trunk port.

See Figure 13-3. 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 13-3 Potential Problem with 802.1Q Tunneling and Native VLANs



System MTU

The default system MTU for traffic on the Catalyst 3750 Metro 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 with some Layer 2 features and with 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 the switch virtual interface (SVI) that includes tunnel ports, untagged IP packets received from the tunnel port are recognized and routed by the switch. This allows the customer to access the Internet through its native VLAN. If this access is not required, 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 ACLs are 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** *interface-id*	Enter 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 12).
Step 3	**switchport access vlan** *vlan-id*	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.

	Command	Purpose
Step 5	**exit**	Return to global configuration mode.
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, if a customer VLAN ID is the same as the native VLAN, the trunk port does not apply a metro tag, and packets might be sent to the wrong destination.
Step 7	**end**	Return to privileged EXEC mode.
Step 8	**show dot1q-tunnel**	Display the tunnel ports on the switch.
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 auto. Use the **no vlan dot1q tag native** global configuration command to disable tagging of native VLAN packets.

This example shows how to configure aport 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 Fast Ethernet port 7 is VLAN 22.

```
Switch(config)# interface fastethernet1/0/7
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 fastethernet1/0/7
Port
-----
Fa1/0/7
-----
Switch# show vlan dot1q tag native
dot1q native vlan tagging is enabled
```

Configuring VLAN Mapping

VLAN mapping, or VLAN ID translation, is a feature that you can configure on the enhanced-services (ES) ports connected to the service-provider network to map the customer VLANs to service-provider VLANs. VLAN mapping acts as a filter on the ES ports without affecting the internal operation of the switch or the customer VLANs. A switch might also have a number of reserved VLANs or have a limited VLAN range. When customers want to use a VLAN number in the reserved range, you can use VLAN mapping to overlap customer VLANs by encapsulating the customer traffic in 802.1Q tunnels.

With VLAN mapping, the customer VLAN ID in the 802.1Q or ISL tag, or the inner and outer tag in an 802.1Q tunneled frame, are mapped (or translated) just before a packet is transmitted and just after a packet is received. The service-provider VLANs are not seen by the switch, so all configuration and statistics are done with the customer side-VLANs.

Note Do not configure VLAN mapping on an interface configured for MPLS or EoMPLS.

This section includes this configuration information:

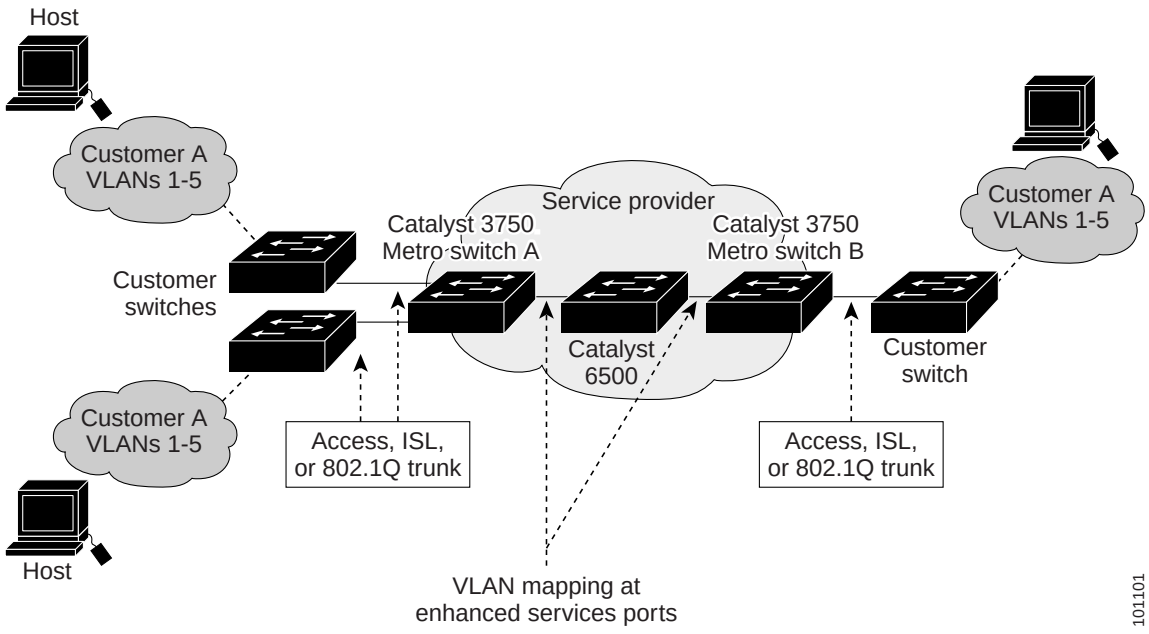
Default VLAN Mapping Configuration

By default, no VLAN mapping is configured.

Mapping Customer VLANs to Service-Provider VLANs

Figure 13-4 shows a topology where a customer uses the same VLANs in multiple sites at 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 an ES on each side of the service-provider network. See the example following the configuration steps.

Figure 13-4 Example of VLAN Mapping



Beginning in privileged EXEC mode, follow these steps on an ES port to map a customer VLAN ID to a service-provider VLAN ID:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode and the ES interface connected to the service-provider network.
Step 3	**switchport vlan mapping** *vlan-id translated-id*	Enter the VLAN IDs to be mapped: • *vlan-id*—the customer VLAN ID entering the switch from the customer network. The range is from 1 to 4094. • *translated-id*—the assigned service-provider VLAN ID. The range is from 1 to 4094.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show vlan mapping**	Verify the configuration.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no switchport vlan mapping** *vlan-id translated-id* command to remove the VLAN translation information.

This example shows how to map VLAN IDs 1, 2, 3, 4, and 5 in the customer network to VLANs 2001, 2002, 2003, and 2004 in the service-provider network as shown in Figure 13-4. You configure these same VLAN mapping commands for an ES port in Switch A and Switch B.

```
Switch(config)# interface gigabiethernet1/1/1
Switch(config-if)# switchport vlan mapping 1 1001
Switch(config-if)# switchport vlan mapping 2 1002
Switch(config-if)# switchport vlan mapping 3 1003
Switch(config-if)# switchport vlan mapping 4 1004
Switch(config-if)# switchport vlan mapping 5 1005
Switch(config-if)# exit
```

Mapping Customer 802.1Q Traffic with VLAN IDs

You can also use the **switchport vlan mapping** interface configuration command with the **dot1qtunnel** keyword to map 802.1Q traffic with a specified outer VLAN ID and inner VLAN ID to a single VLAN ID in the service-provider network. You can use **drop** keyword to specify that traffic on a specified outer VLAN ID is dropped unless the specified inner VLAN and outer VLAN ID combination is explicitly translated.

Beginning in privileged EXEC mode, follow these steps on an ES port to configure the switch to drop or translate 802.1Q tagged traffic:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode and the ES interface connected to the service-provider network.

	Command	Purpose
Step 3	**switchport vlan mapping dot1qtunnel** *outer-vlan-id* {*inner-vlan-id translated-vlan-id* | **drop**}	Specify the action to be taken if the outer VLAN ID of the original 802.1Q packet matches the specified value: • *outer-vlan-id* —Enter the outer VLAN ID of the original 802.1Q packet. • *inner-vlan-id translated-vlan-id* —Enter the inner VLAN ID of the original 802.1Q packet and the translated VLAN ID for the service-provider network. The VLAN ID range is from 1 to 4094. • **drop**—Specify that traffic with the specified outer VLAN ID is dropped unless the inner VLAN ID is matched in the existing VLAN mappings.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show vlan mapping**	Verify the configuration.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no switchport vlan mapping dot1qtunnel** *outer-vlan-id* {*inner-vlan-id translated-vlan-id* | **drop**} interface configuration command to remove the mapping information.

This example shows how to configure the ES port so that 802.1Q traffic with the outer VLAN ID 5 and the inner VLAN ID 6 would leave the switch with the single VLAN ID of 10. The 802.1Q traffic with the outer VLAN 5 and any inner VLAN ID other than 6 would be dropped. Other traffic on VLAN 5 would also be dropped:

```
Switch(config)# interface gigabiethernet1/1/1
Switch(config-if)# switchport vlan mapping dot1q-tunnel 5 6 10
Switch(config-if)# switchport vlan mapping dot1q-tunnel 5 drop
Switch(config-if)# exit
```

Understanding Layer 2 Protocol Tunneling

Customers at different sites connected across a service-provider network need to run various Layer 2 protocols to scale their topology to include all remote sites, as well as the local sites. STP must run properly, and every VLAN should build a proper spanning tree that includes the local site and all remote sites across the service-provider infrastructure. Cisco Discovery Protocol (CDP) must 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 infrastructure encapsulate Layer 2 protocol packets with a special MAC address and send them across the service-provider network. Core switches in the network do not process these packets but forward them as normal packets. Layer 2 protocol data units (PDUs) for CDP, STP, or VTP cross the service-provider infrastructure and are delivered to customer switches on the outbound side of the service-provider network. Identical packets are received by all customer ports on the same VLANs with these results:

- Users on each of a customer's sites are able to 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 through the service provider to all switches.

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 or trunk ports and enabling tunneling on the service-provider access or trunk port.

For example, in Figure 13-5, 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 13-6.

Figure 13-5 Layer 2 Protocol Tunneling

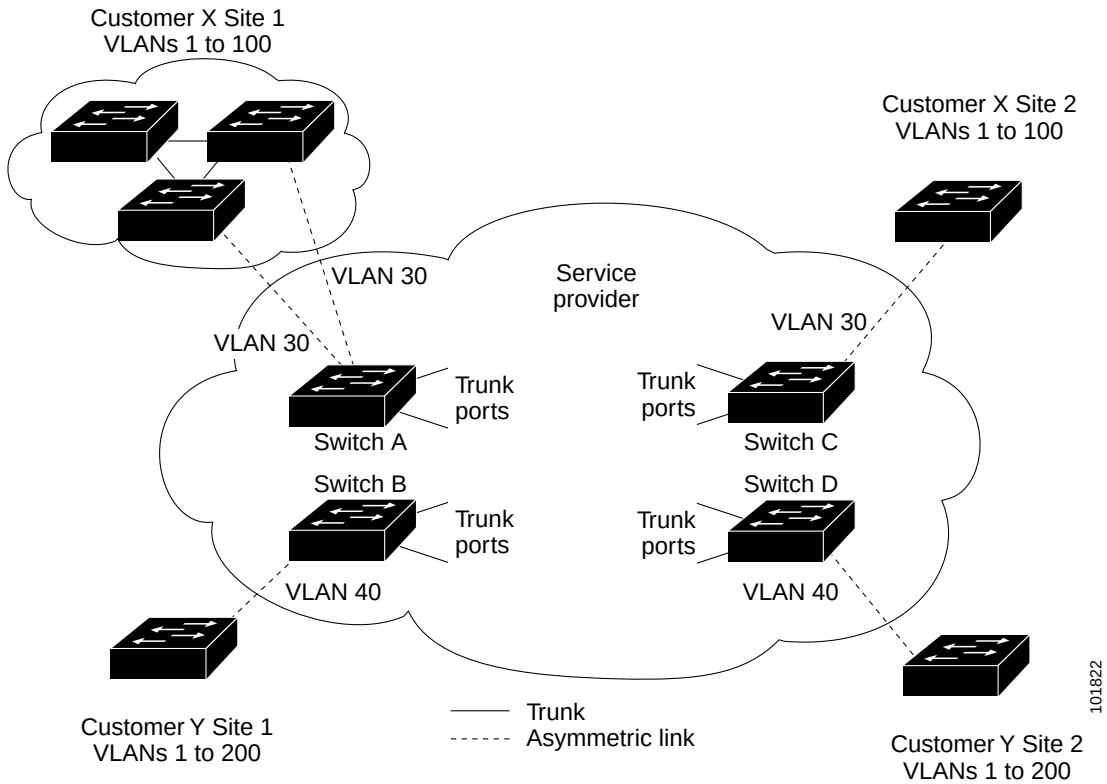
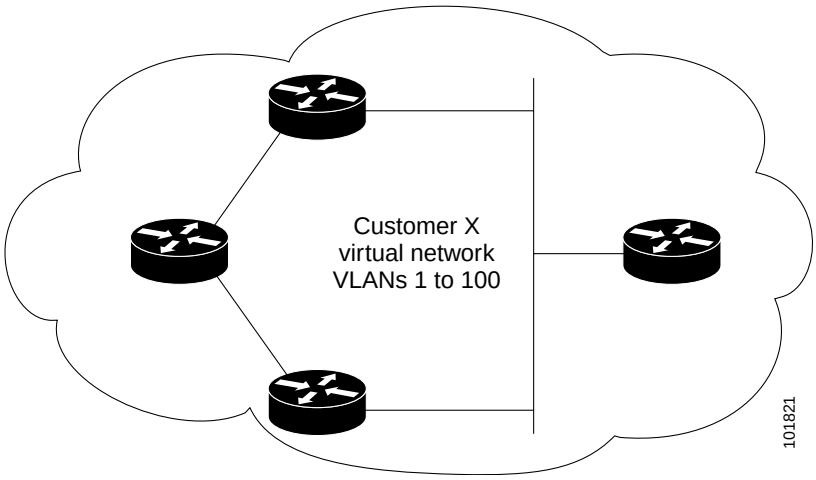


Figure 13-6 Virtual Network Topology without BPDU Tunneling



Configuring Layer 2 Protocol Tunneling

You 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. 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, tunnel ports, or trunk 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.

When the Layer 2 PDUs that entered the inbound edge switch through a Layer 2 protocol-enabled port exit the switch 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 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 ports, access ports, and Layer 2 protocol-enabled trunk ports in the same metro VLAN. Therefore, the Layer 2 PDUs are kept intact and delivered across the service-provider infrastructure to the other side of the customer network.

See Figure 13-5, 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 B from Customer Y in Site 1 are forwarded to the infrastructure as double-tagged packets with the well-known MAC address as the destination MAC address. These double-tagged packets have the outer VLAN tag of 40 as well as an inner VLAN tag (for example, VLAN 100). When the double-tagged packets reach 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 or trunk ports on the edge switch connected to access or trunk ports on the customer switch. In this case, the encapsulation and de-encapsulation behavior 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.

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

Default Layer 2 Protocol Tunneling Configuration

Table 13-1 shows the default Layer 2 protocol tunneling configuration.

Table 13-1 Default Layer 2 Ethernet Interface VLAN Configuration

Feature	Default Setting
Layer 2 protocol tunneling	Disabled for CDP, STP, and VTP.
Shutdown threshold	No threshold for packets-per-second of Layer 2 PDUs per port for the port to shut down.
Drop threshold	No threshold for packets-per-second of Layer 2 PDUs per port for the port to drop the PDUs.
Class of service (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.
- The switch does not support Layer 2 protocol tunneling on ports with switchport mode **dynamic auto** or **dynamic desirable**.
- 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.
- The edge switches on the outbound side restore the proper Layer 2 protocol and MAC address information and forward the packets to all tunnel ports, access ports and Layer 2 protocol-enabled trunk 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.
- EtherChannel port groups are compatible with tunnel ports as long as 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 access or trunk 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 re-enable the port (by entering a **shutdown** and **no shutdown** command sequence) or 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. No CDP packets are 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 rate-limit BPDUs 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 the port receives them is below the drop threshold.
- Because tunneled PDUs (especially STP BPDUs) must be delivered to all remote sites for the customer virtual network to operate 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 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** *interface-id*	Enter the 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 12).
Step 3	**switchport mode access** or **switchport mode dot1q-tunnel** or **switchport mode trunk**	Configure the interface as an access port, an 802.1Q tunnel port or a trunk 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**] *value*	(Optional) Configure the threshold in packets per second to be received for encapsulation before the interface shuts down. The port is disabled if the configured threshold is exceeded. If no protocol option is specified, the threshold is applied 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	**l2protocol-tunnel drop-threshold** [**cdp** | **stp** | **vtp**] *value*	(Optional) Configure the threshold in packets per second to be received for encapsulation before the interface drops packets. The port drops packets if the configured threshold is exceeded. If no protocol option is specified, the threshold is applied 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	**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 can be brought out of the disabled state and allowed to try again. You can also set the time interval. Errdisable recovery is disabled by default; when enabled, the default time interval is 300 seconds.
Step 9	**l2protocol-tunnel cos** *value*	(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 threshold, 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 of them. 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 gigabitethernet1/0/2
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
------- -------- --------- --------- ------------- ------------- -------------
Gi1/0/2 cdp           1500      1000             0             0             0
        stp           1500      1000             0             0             0
        vtp           1500      1000             0             0             0
```

Monitoring and Maintaining Tunneling and Mapping Status

Table 13-2 shows the privileged EXEC commands for monitoring and maintaining 802.1Q and Layer 2 protocol tunneling and VLAN mapping.

Table 13-2 Commands for Monitoring and Maintaining Tunneling and VLAN Mapping

Command	Purpose
clear l2protocol-tunnel counters	Clear the protocol counters on Layer 2 protocol tunneling ports.
show dot1q-tunnel	Display 802.1Q tunnel ports on the switch.
show dot1q-tunnel interface *interface-id*	Verify if a specific interface is a tunnel port.
show l2protocol-tunnel	Display information about Layer 2 protocol tunneling ports.
show errdisable recovery	Verify if the recovery timer from a Layer 2 protocol-tunnel error disable state is enabled.
show interface *interface-id* **vlan mapping**	Display VLAN mapping information for the specified interface. The interface can be an ES port or a VLAN.
show l2protocol-tunnel interface *interface-id*	Display information about a specific Layer 2 protocol tunneling port.
show l2protocol-tunnel summary	Display only Layer 2 protocol summary information.
show vlan dot1q native	Display the status of native VLAN tagging on the switch.
show vlan mapping	Display VLAN mapping information (contents of the VLAN mapping table) for the ES ports.

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