Configuring IEEE 802.1Q Tunneling

This chapter describes how to configure IEEE 802.1Q tunneling on the Cisco Industrial Ethernet 2000U Series (IE 2000U) and Connected Grid Switches, hereafter referred to as *switch*.

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 switch supports IEEE 802.1Q tunneling and Layer 2 protocol tunneling; Layer 2 protocol tunneling is described in Chapter 10, "Configuring Layer 2 Protocol Tunneling."

Note For complete syntax and usage information for the commands used in this chapter, see the documents listed in the "Related Documents" section on page 8-8.

This chapter includes the following sections:

Information About 802.1Q Tunneling

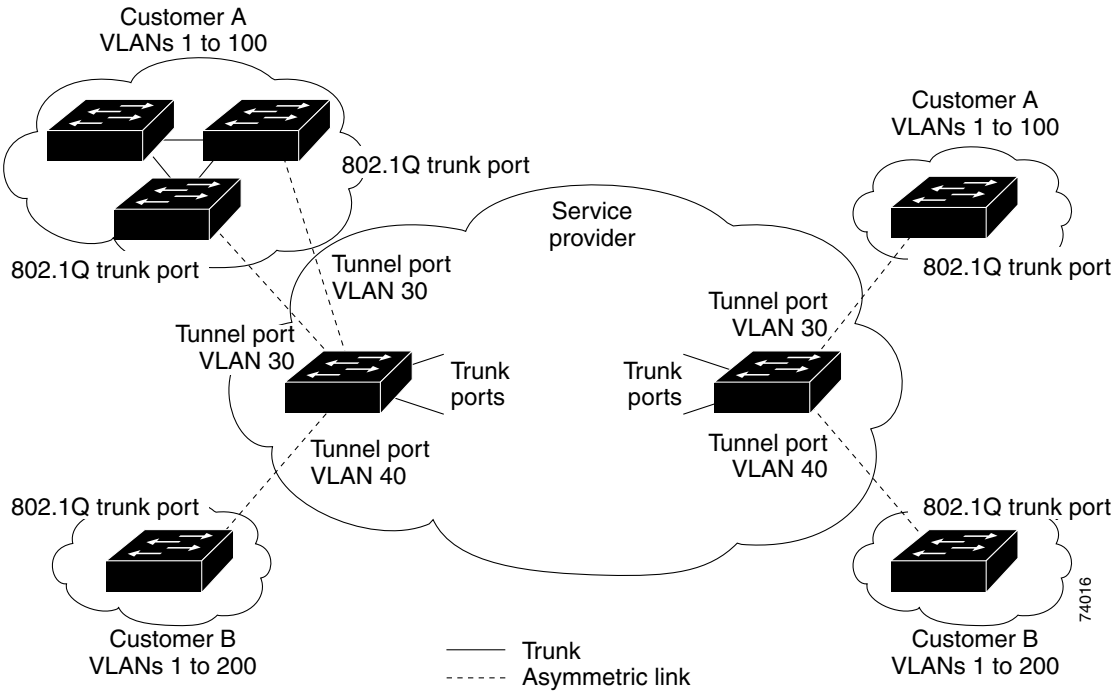
Business customers of service providers often have specific requirements for VLAN IDs and the number of VLANs to be supported. The VLAN ranges required by different customers in the same service-provider network might overlap, and traffic of customers through the infrastructure might be mixed. Assigning a unique range of VLAN IDs to each customer would restrict customer configurations and could easily exceed the VLAN limit (4096) of the 802.1Q specification.

Using the 802.1Q tunneling (QinQ) feature, service providers can use a single VLAN to support customers who have multiple VLANs. Customer VLAN IDs (C-VLANs) are preserved, and traffic from different customers is segregated within the service-provider network, even when they appear to be in the same VLAN. Using 802.1Q tunneling expands VLAN space by using a VLAN-in-VLAN hierarchy and retagging the tagged packets. A port configured to support 802.1Q tunneling is called a *tunnel port*. When you configure tunneling, you assign a tunnel port to a VLAN ID that is dedicated to tunneling. Each customer requires a separate service-provider VLAN ID (S-VLAN), but that VLAN ID supports all of the customer's VLANs. Configuring 802.1Q tunneling on a tunnel port is referred to as *traditional QinQ*.

Customer traffic tagged in the normal way with appropriate VLAN IDs comes from an 802.1Q trunk port on the customer device and into a tunnel port on the service-provider edge switch. The link between the customer device and the edge switch is asymmetric because one end is configured as an 802.1Q trunk port, and the other end is configured as a tunnel port. You assign the tunnel port interface to an access VLAN ID that is unique to each customer. See Figure 8-1.

Note By default, VLANs configured on the switch are user network interface-enhanced network interface (UNI-ENI) isolated VLANs. In a UNI-ENI isolated VLAN, 802.1Q tunneled access ports on the switch are isolated from each other. If you use the **uni-vlan community** VLAN configuration command to change a VLAN to a UNI-ENI community VLAN, local switching occurs between these ports. For more information about UNI-ENI VLANs, see Chapter 3, "Configuring VLANs."

Figure 8-1 802.1Q Tunnel Ports in a Service-Provider Network



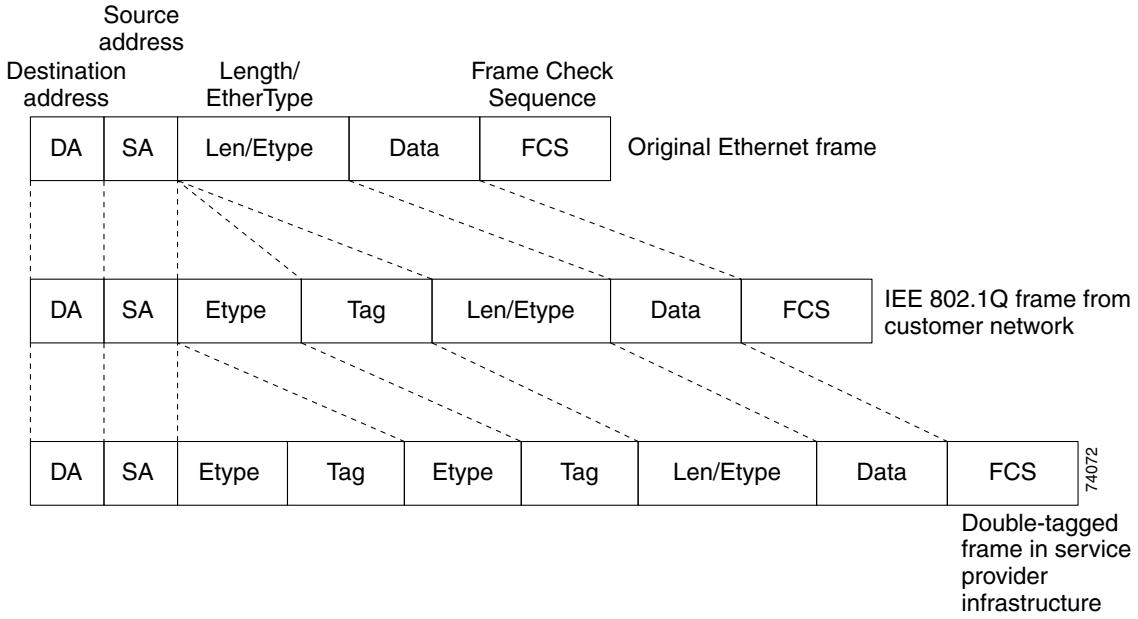
Packets coming from the customer trunk port into the tunnel port on the service-provider edge switch are normally 802.1Q-tagged with the appropriate VLAN ID. The tagged packets remain intact inside the switch and when they exit the trunk port into the service-provider network, they are encapsulated with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

customer. The original customer 802.1Q tag is preserved in the encapsulated packet. Therefore, packets entering the service-provider network are double-tagged, with the outer (metro) tag containing the customer's access VLAN ID, and the inner VLAN ID being that of the incoming traffic.

When the double-tagged packet enters another trunk port in a service-provider core switch, the outer tag is stripped as the switch processes the packet. When the packet exits another trunk port on the same core switch, the same metro tag is again added to the packet. Figure 8-2 shows the tag structures of the double-tagged packets.

Note Remove the Layer 2 protocol configuration from a trunk port because incoming encapsulated packets change that trunk port to error disabled. The outgoing encapsulated VTP (CDP and STP) packets are dropped on that trunk.

Figure 8-2 Original (Normal), 802.1Q, and Double-Tagged Ethernet Packet Formats



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

In Figure 8-1, Customer A was assigned VLAN 30, and Customer B was assigned VLAN 40. Packets entering the edge switch tunnel ports with 802.1Q tags are double-tagged when they enter the service-provider network, with the outer tag containing VLAN ID 30 or 40, appropriately, and the inner tag containing the original VLAN number, for example, VLAN 100. Even if both Customers A and B have VLAN 100 in their networks, the traffic remains segregated within the service-provider network because the outer tag is different. Each customer controls its own VLAN numbering space, which is independent of the VLAN numbering space used by other customers and the VLAN numbering space used by the service-provider network.

At the outbound tunnel port, the original VLAN numbers on the customer's network are recovered. It is possible to have multiple levels of tunneling and tagging, but the switch supports only one level in this release.

If traffic coming from a customer network is not tagged (native VLAN frames), these packets are bridged or routed as normal packets. All packets entering the service-provider network through a tunnel port on an edge switch are treated as untagged packets, whether they are untagged or already tagged with 802.1Q headers. The packets are encapsulated with the metro tag VLAN ID (set to the access VLAN of the tunnel port) when they are sent through the service-provider network on an 802.1Q trunk port. The priority field on the metro tag is set to the interface class of service (CoS) priority configured on the tunnel port. (The default is zero if none is configured.)

Prerequisites

- Be familiar with the information in the "Information About 802.1Q Tunneling" section on page 8-1 and "Guidelines and Limitations" section on page 8-4.
- Ensure that your network strategy and planning for your network are complete.

Guidelines and Limitations

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, maximum transmission units (MTUs), and 802.1Q tunneling interactions with other features 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 packets into the service-provider network. However, packets going through the core of the service-provider network can be carried through 802.1Q trunks, ISL trunks, or nontrunking links. When 802.1Q trunks are used in these core switches, the native VLANs of the 802.1Q trunks must not match any native VLAN of the nontrunking (tunneling) port on the same switch because traffic on the native VLAN would not be tagged on the 802.1Q sending trunk port.

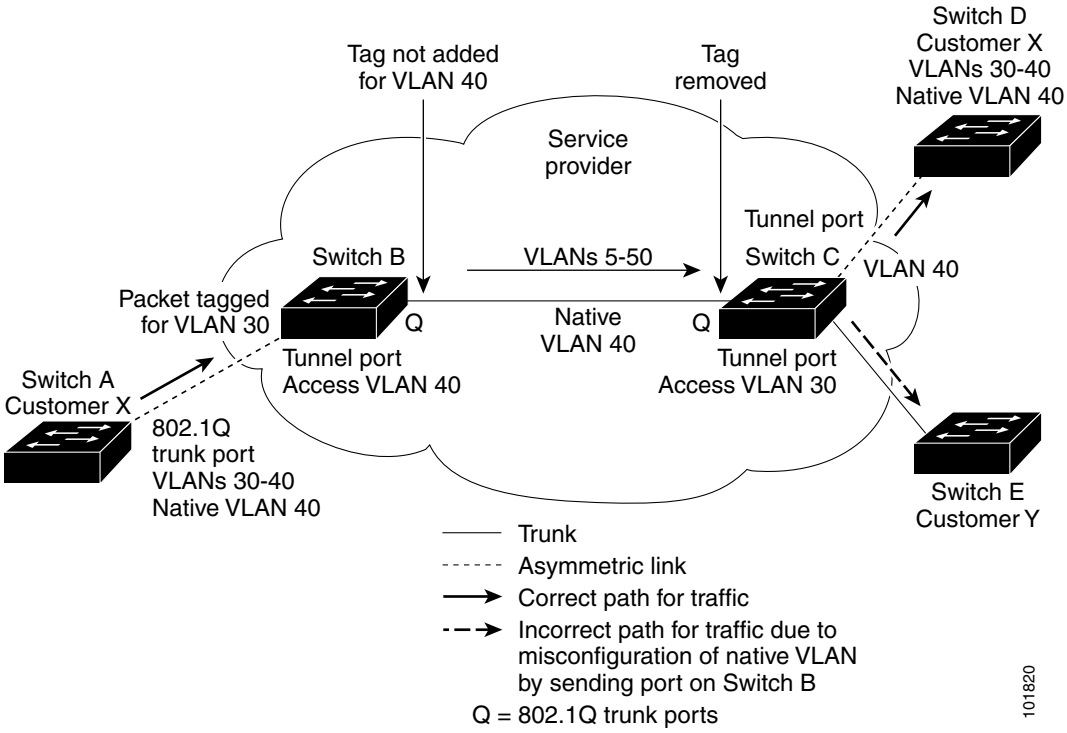
See Figure 8-3. VLAN 40 is configured as the native VLAN for the 802.1Q trunk port from Customer X at the ingress edge switch in the service-provider network (Switch B). Switch A of Customer X sends a tagged packet on VLAN 30 to the ingress tunnel port of Switch B in the service-provider network, which 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. The switch does not support ISL trunks.
- 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 8-3 Potential Problem with 802.1Q Tunneling and Native VLANs



System MTU

The default system MTU for traffic on the switch is 1500 bytes. You can configure Fast Ethernet ports to support frames larger than 1500 bytes by using the **system mtu** global configuration command. You can configure Gigabit Ethernet ports to support frames larger than 1500 bytes by using the **system mtu jumbo** global configuration command. Because the 802.1Q tunneling feature increases the frame size by 4 bytes when the metro tag is added, you must configure all switches in the service-provider network to be able to process maximum frames by increasing the switch system MTU size to at least 1504 bytes. The maximum allowable system MTU for Gigabit Ethernet interfaces is 9000 bytes; the maximum system MTU for Fast Ethernet interfaces is 1998 bytes.

802.1Q Tunneling and Other Features

Although 802.1Q tunneling works well for Layer 2 packet switching, there are incompatibilities between some Layer 2 features and Layer 3 switching.

Note Layer 3 switching is supported only when the IP services image is running on the switch.

- A tunnel port cannot be a routed port.
- IP routing is not supported on a VLAN that includes 802.1Q tunnel 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 its 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.
- UniDirectional Link Detection (UDLD) is supported on 802.1Q tunnel ports.
- Port Aggregation Protocol (PAgP) and Link Aggregation Control Protocol (LACP) are supported only on 802.1Q tunnel ports that are network node interfaces (NNIs) or enhanced network interfaces (ENIs). UNIs do not support PAgP and LACP.
- Loopback detection is supported on 802.1Q tunnel ports.
- When an NNI or ENI port is configured as an 802.1Q tunnel port, spanning-tree bridge protocol data unit (BPDU) filtering is automatically enabled on the interface, and the Cisco Discovery Protocol (CDP) and the Layer Link Discovery Protocol (LLDP) are automatically disabled on the interface. UNIs do not support BPDU filtering, CDP, or LLDP.
- In a UNI-ENI isolated VLAN, 802.1Q tunneled access ports are isolated from each other, but in a UNI-ENI community VLAN, local switching occurs between these ports. For more information about UNI-ENI VLANs, see Chapter 3, “Configuring VLANs.”

Default Settings

By default, 802.1Q tunneling is disabled because the default switchport mode is access. Tagging of 802.1Q native VLAN packets on all 802.1Q trunk ports is also disabled. By default, VLANs on the switch are UNI-ENI isolated VLANs.

Configuring an 802.1Q Tunneling Port

BEFORE YOU BEGIN

Review the “Guidelines and Limitations” section on page 8-4.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter interface configuration mode for the interface to be configured as a tunnel port. This should be the edge port in the service-provider network that connects to the customer switch. Valid interfaces include physical interfaces and port-channel logical interfaces (port channels 1 to 48).
Step 3	**no shutdown**	Enable the port, if necessary. By default, UNIs and ENIs are disabled, and NNIs are enabled.
Step 4	**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. **Note** If the VLAN is a UNI-ENI isolated VLAN, local switching does not occur between UNIs and ENIs on the switch. If the VLAN is a UNI-ENI community VLAN, local switching is allowed.
Step 5	**switchport mode dot1q-tunnel**	Set the interface as an 802.1Q tunnel port.
Step 6	**exit**	Return to global configuration mode.
Step 7	**vlan dot1q tag native**	(Optional) Set the switch to enable tagging of native VLAN packets on all 802.1Q trunk ports. When not set, and a customer VLAN ID is the same as the native VLAN, the trunk port does not apply a metro tag, and packets could be sent to the wrong destination.
Step 8	**end**	Return to privileged EXEC mode.
Step 9	**show running-config** **show dot1q-tunnel**	Display the ports configured for 802.1Q tunneling. Display the ports that are in tunnel mode.
Step 10	**show vlan dot1q tag native**	Display 802.1Q native VLAN tagging status.
Step 11	**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 access. Use the **no vlan dot1q tag native** global configuration command to disable tagging of native VLAN packets.

EXAMPLE

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 is VLAN 22. This VLAN is by default a UNI-ENI isolated VLAN.

```
Switch(config)# interface gigabitethernet0/2
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 gigabitethernet0/2
dot1q-tunnel mode LAN Port(s)
------------------------------
Gi0/1

Switch# show vlan dot1q tag native
dot1q native vlan tagging is enabled
```

Verifying Configuration

Command	Purpose
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 vlan dot1q tag native	Display the status of native VLAN tagging on the switch.

Related Documents

- Cisco IOS Master Command List, All Releases
- Cisco IOS LAN Switching Command Reference
- Cisco IOS Interface and Hardware Component Command Reference

Feature History

Platform	First Supported Release
IE 2000U	Cisco IOS Release 15.0(2)EH
CGS 2520 Switch	Cisco IOS Release 12.2(53)EX
Ethernet Switch Module (ESM) for CGR 2010	Cisco IOS Release 12.2(53)EX