Contents

1 Configuring QinQ

1.1 Introduction

1.1.1 Overview

When a client packet enters a provider edge (PE), the QinQ technology first encapsulates the packet with a public virtual local area network (VLAN) tag before transmitting the packet in the SP network. The private VLAN tag, if any, in the client packet is regarded as data. Then, the packet carries two VLAN tags to pass through the SP network.

1.1.2 Background and Function

1. Background

On a metropolitan area network (MAN), SPs need a large number of VLANs to isolate and identify clients so as to achieve refined management. As a variety of services develop, SPs also need to differentiate services to provide different transmission pipelines and QoS policies. However, IEEE 802.1Q supports 4094 VLANs only, far from meeting the needs of SPs.

2. Function

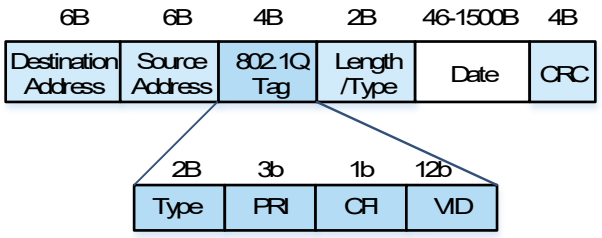
- QinQ is a kind of dual-tag encapsulation technology. It adds outer VLAN tags to packets in the SP network so that client VLANs in different private networks can be reused to increase the number of VLAN tags available to clients. The number of VLANs actually available to SPs is increased to 4094x4094.
- An outer tag is also equivalent to a tunnel, which enables transparent transmission of inner tags. This turns QinQ into a layer-2 VPN solution that is low in cost, easy to deploy, and suitable for small-sized networks.
- Inner and outer tags can also represent different information according to service requirements, helpful for SPs to segment and manage their services.

1.1.3 QinQ Packet Format

- Ethernet frame with a single VLAN tag

The IEEE 802.1Q standard modifies the Ethernet frame format by adding a 4-byte 802.1Q tag between the source MAC address field and protocol type field. Figure 1-1 shows the frame format. The tag contains a VID (VLAN ID) field, which is used to indicate the VLAN where the frame is. Not all devices can identify 802.1Q frames with tags, and therefore Ethernet frames on VLANs can be tagged or untagged.

Figure 1-1 Frame Format of 802.1Q Encapsulation

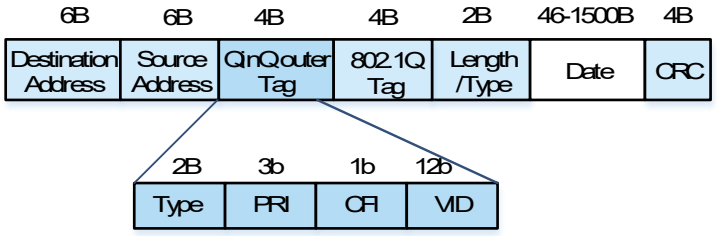


- **Type**: Consisting of 2 bytes; 0x8100 indicates 802.1Q frame. Devices that do not support IEEE 802.1Q will drop such frames received.
- **PRI**: Consisting of 3 bits, indicating layer-2 priority. The value range is 0 to 7, corresponding to CoS priorities of QoS. A larger value indicates a higher priority.
- **CFI**: Consisting of 1 bit, used to distinguish Ethernet frame, FDDI frame, and token ring network frame; 0 indicates Ethernet frame.
- **VID**: Consisting of 12 bits, indicating the client VLAN ID; values 0 and 4095 are reserved and unavailable. The value range is from 1 to 4094.

- Ethernet frame with QinQ dual VLAN tags

The QinQ function inserts a 4-byte QinQ outer tag (including four fields: **Type**, **PRI**, **CFI**, and **VID**) before the 802.1Q tag.

Figure 1-2 Frame Format of QinQ Encapsulation



- **Type**: Consisting of 2 bytes, also known as TPID; the default value is 0x8100, indicating 802.1Q frame. On the devices of some vendors, the TPID is set to 0x9100 or other values. TPID can be configured (Configuring TPID) to ensure that the TPIDs of packets to be forwarded are compatible with the TPIDs supported by third-party devices.
- **PRI**: Consisting of 3 bits, indicating priority; the default value 0 indicates a normal flow. The value of this field corresponds to the CoS value in the QoS policy. A larger value indicates a higher priority. The value of this field can be modified by configuring priority replication or priority mapping (Configuring Priority Replication or Priority Mapping) to define different transmission priority policies for packets.
- **CFI**: Consisting of 1 bit, used to distinguish Ethernet frame, FDDI frame, and token ring network frame; value 0 indicates Ethernet frame.
- **VID**: Consisting of 12 bits, indicating the SP VLAN ID. The value is from 1 to 4094.

1.1.4 QinQ Packet Forwarding

1. QinQ Encapsulation

The ingress of an SP's QinQ edge device is called dot1q-tunnel port, or tunnel port for short. All the frames entering the edge device, no matter whether they carry the 802.1Q tag or not, will be packaged with the SP tag and then be forwarded. As shown in Figure 1-1, when a packet of CE-A1 is forwarded to PE 1, the tunnel port encapsulates the packet with a VLAN 10 tag, and then forwards the packet from the uplink port. This process is called QinQ encapsulation. QinQ encapsulation is classified into Basic QinQ Encapsulation and Selective QinQ Encapsulation. Selective QinQ encapsulation is further classified into client tag-based encapsulation and ACL-based encapsulation.

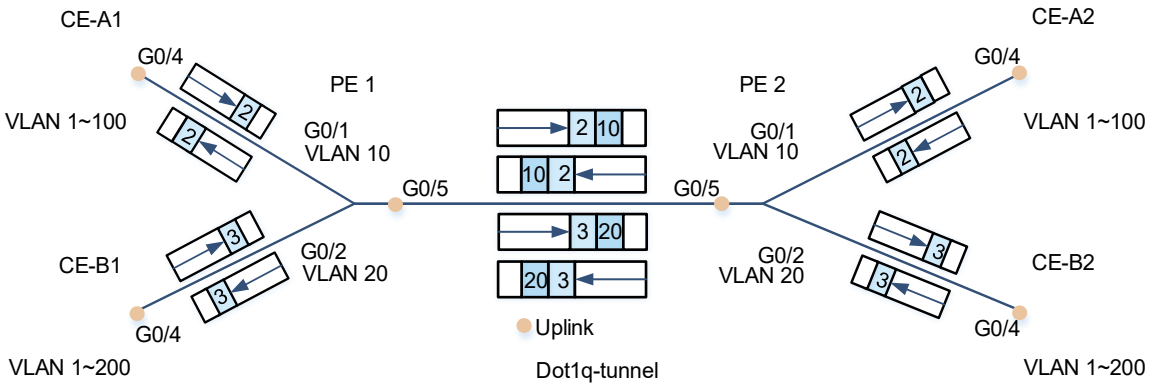
2. QinQ Forwarding

A QinQ-encapsulated packet will be transmitted in the SP network. Generally, devices directly forward the QinQ packet with dual tags. During the transmission, you may need to modify the VIDs in the inner and outer tags. In this case, you can configure a modification policy on the access, trunk, hybrid, or uplink port to modify QinQ tags (Modifying QinQ Tag).

3. QinQ Decapsulation

When a packet with an SP tag is forwarded to PE 2, PE 2 will forward the packet to the tunnel port with native VLAN set to VLAN 10, and the tunnel port will strip the outer tag of the packet and send the packet to CE-A2. This process is called QinQ decapsulation, and the decapsulation mode must be consistent with the encapsulation mode.

Figure 1-1 QinQ Packet Forwarding Process



Note

- The QinQ encapsulation and decapsulation functions are usually used on client-oriented interfaces to implement classification, package management, and transmission of client data.
- QinQ termination means that, after receiving packets, the device identifies and strips the dual VLAN tags, and then sends the packets to other service modules for processing. The QinQ termination function is usually used on a server or network core device to terminate client VLAN tags and realize the functions such as client location or service identification. QinQ termination is not QinQ decapsulation. QinQ decapsulation strips only the outer tag, while QinQ termination strips the dual tags.

1.1.5 Basic QinQ Encapsulation

The basic QinQ encapsulation mode is based on the process that the interface encapsulates the packets with tags. The VLAN ID in the tag is the native VLAN of dot1q-tunnel port.

After receiving packets, the dot1q-tunnel port regards all the packets as untagged regardless of whether they carry tags, determines whether the packets belong to the native VLAN of the local interface, learns the corresponding relationship between the client-end MAC address and the native VLAN, and stores it in the MAC address table. After that, the interface adds a tag with VID set to the native VLAN to the packets and then forwards the packets.

If the original packet does not carry a VLAN tag, the packet will have only one tag with native VLAN ID after undergoing QinQ encapsulation. If the original packet carries a VLAN tag, the packet will carry dual tags after undergoing QinQ encapsulation and be transmitted in the SP network.

By default, the native VLAN of dot1q-tunnel port is VLAN 1 and the only one allowed VLAN is untagged VLAN 1. "Untagged" in the QinQ function means that the outer tag of QinQ is not carried, because the dot1q-tunnel port must strip the outer tag before sending the packet back to the client network.

Weaknesses: The basic QinQ encapsulation cannot distinguish the types of client packets received by the interface or select the VLAN used for packet encapsulation, and therefore it is not flexible enough.

1.1.6 Selective QinQ Encapsulation

Selective QinQ encapsulation flexibly adds different outer VLAN tags to data flows according to rules.

Selective QinQ implements two encapsulation modes:

- Add an outer VLAN tag based on the inner client VLAN tag.

  The selective QinQ based on C-Tag can distinguish different types of data flows according to the client VLAN tag, and encapsulate different outer tags to implement transparent transmission.

- Add an outer VLAN tag based on the ACL.

  The selective QinQ based on ACL can distinguish the MAC address, IP protocol, source address, destination address, and priority of service flows, or the interface number of applications by using ACL policies, and encapsulate different service flows with different VLAN tags to implement transparent transmission. In this case, the VLAN ID in the VLAN tag is not necessarily the native VLAN of the interface.

When using this function, in addition to configuring the selective QinQ encapsulation policy, you also need to add the outer VLAN to the list of allowed VLANs of the dot1q-tunnel port.

Weaknesses: After the dot1q-tunnel port receives a packet, the learned MAC address entry is the corresponding relationship between the native VLAN and the client MAC address; the packet returned by the peer end belongs to the outer VLAN, and the corresponding entries of the outer VLAN and the client-end MAC address cannot be found in the MAC address table, which will cause the device to perform flooding so as to find the egress of packets.

1.1.7 Modifying QinQ Tag

In the process of QinQ packet transmission, if the outer or inner tags of data packets in the SP network need to be modified due to the actual network topology requirement, the QinQ tag modifying function needs to be enabled. The device can distinguish the packets of different clients, different services, and different priorities based on policies, and modify the VLAN IDs for the inner or outer tags of packets.

1.1.8 Layer-2 Protocol Tunneling

- Source

Spanning Tree Protocol (STP) is an application that eliminates layer-2 loops by blocking redundant links on the network. It also provides the link backup function. It will transmit a special bridge protocol data unit (BPDU) packet on the network. After recognizing such a packet, the device will calculate the spanning tree according to the packet information, which may lead to modification of the network topology.

The generic attribute registration protocol VLAN registration protocol (GVRP) is an application used to dynamically configure and diffuse VLAN memberships. It will transmit special VLAN registration and deregistration packets on the network. After recognizing a GVRP packet, the device will modify the VLAN relationship according to the packet information, which will also affect the network topology.

To unify the internal topology of the client network when the client networks are interconnected across the SP network, we need to run STP and GVRP on the client network. However, such a special layer-2 protocol packet will be identified as a special packet by the SP device and participate in the SP topology calculation, affecting the SP network topology.

Therefore, we need to transmit the STP and GVRP packets of the client network through a tunnel in the SP network, so that the SP can forward the client's special packets as ordinary packets. The layer-2 packet tunnel function is needed in this case.

- Principles

When a layer-2 protocol packet of the client network enters a PE, the destination MAC address is changed to a private address for forwarding in the SP network. When the packet arrives at an edge device of the other end, the destination MAC address is changed to a public address before the packet is returned to the client network of the other end. This implements layer-2 protocol packet transmission through a tunnel in the SP network.

1.1.9 Protocols and Standards

- IEEE 802.1ad: IEEE Standard for Local and Metropolitan Area Networks---Virtual Bridged Local Area Networks---Amendment 4:Provider Bridges

1.2 Restrictions and Guidelines

- A routed port cannot be configured as a tunnel port. The **switchport** command can be used to configure an interface as a switch port.
- The IEEE 802.1x function cannot be enabled on an interface set to **Tunnel**.
- The port security function cannot be enabled on an interface set to **Tunnel**.
- When a tunnel port is configured as the source port of the remote switched port analyzer (RSPAN), the packets whose outer tags contain VIDs consistent with the RSPAN VLAN IDs are monitored as a data flow.
- If you want to match the ACL applied to the tunnel port with the VID in the client tag, use the **inner** keyword.
- After a switch port is enabled with QinQ, you must enable SVGL sharing before IGMP snooping. Otherwise, IGMP snooping will not work on the QinQ-enabled port. Please see *Configuring IGMP Snooping*.
- Configure the egress on the client network that is connected to the SP network as an uplink port. The SP TPID value for the uplink port of the client network must be the same as that for the downlink interface of the

SP network.

- The default MTU value of an interface is 1,500 bytes. After added with an outer VLAN tag, a packet is four bytes longer. You are advised to increase the MTU value of each interface in the SP networks to at least 1,504 bytes.

1.3 Configuration Task Summary

1.3.1 Configuring Tag Adding

QinQ tag adding includes the following tasks:

(1) Configure the QinQ encapsulation mode of the tunnel port. Select at least one of them to configure.

- Configuring Basic QinQ Encapsulation
- Configuring C-Tag-Based Selective QinQ Encapsulation

(2) (Optional) Configuring Priority Replication or Priority Mapping. The following configuration tasks are mutually exclusive.

- Configuring Priority Replication
- Configuring Priority Mapping

(3) (Optional) Configuring TPID

1.3.2 Configuring Layer-2 Protocol Tunneling

The following tasks are optional.

- Configuring Layer-2 Protocol Tunneling. The following configuration tasks are optional.
  - Configuring STP Packet Tunneling
  - Configuring GVRP Packet Tunneling

1.4 Configuring Basic QinQ Encapsulation

1.4.1 Overview

Implement layer-2 VPN based on a port-based QinQ policy.

1.4.2 Restrictions and Guidelines

- Configure a VLAN of the SP network as the native VLAN of the dot1q-tunnel port. Because the trunk port will strip the tag of the packet whose VLAN ID is its native VLAN, do not configure the native VLAN of the trunk port in the SP network as the native VLAN of the dot1q-tunnel port.
- After the native VLAN of the dot1q-tunnel port is configured, the data of the VLAN cannot pass through the interface. The native VLAN must be added to the list of untagged VLANs allowed by the dot1q-tunnel port. After the configuration, the dot1q-tunnel port can output the packets of native VLAN without tags.
- If the native VLAN is added to the list of allowed tagged VLANs, packets of the native VLAN output from the dot1q-tunnel port will carry the tag of the SP VLAN, while the client device cannot identify the VLAN ID of SP network. Therefore, in order to ensure normal communication of the uplink and downlink packets in the application of QinQ, the SP VLAN (native VLAN) must be added to the allowed VLAN list of the interface in the untagged form.

- If port-based QinQ is enabled, you do not need to add the VLAN of the client network to the allowed VLAN list of the dot1q-tunnel port. If selective QinQ is enabled, add the VLAN of the client network to the allowed VLAN list of the interface in the tagged or untagged form based on the actual conditions.

1.4.3 Procedure

(1) Create an SP VLAN.

a Enter the privileged EXEC mode.

enable

b Enter the global configuration mode.

configure terminal

c Create an SP VLAN.

vlan *svid*

Only VLAN 1 exists by default.

d Return to the global configuration mode.

exit

(2) Configure the dot1q-tunnel port and allow SP VLAN packets to pass through.

a Enter the interface configuration mode of the port connected to the client network.

interface *interface-type interface-number*

b Configure the port as a dot1q-tunnel port.

switchport mode dot1q-tunnel

An interface works in access mode by default.

c Configure to add the SP VLAN to the untagged VLAN list of the dot1q-tunnel port.

switchport dot1q-tunnel allowed vlan { [**add**] **untagged** *svid* | **remove** *other-vlan-id* }

By default, the allowed VLAN of dot1q-tunnel port is untagged VLAN 1.

(3) Configure the SP VLAN as the native VLAN of dot1q-tunnel port so that the client network data is transmitted through a tunnel on the SP VLAN.

switchport dot1q-tunnel native vlan *svid*

The native VLAN of the dot1q-tunnel port is VLAN 1 by default.

1.5 Configuring C-Tag-Based Selective QinQ Encapsulation

1.5.1 Overview

Encapsulate outer VLAN tags (S-Tags) in packets based on inner tags to ensure preferential transmission and management of layer-2 VPN and service flows.

1.5.2 Restrictions and Guidelines

- This function must be configured based on the basic QinQ, and prevails over the basic QinQ policy.
- In the network environment, for packets received on the dot1q-tunnel port, the VID of the outer tag needs to be modified according to the VID of the inner tag. With this command, you can add an outer VID the same

as the inner VID to the packet so as to output the original VID tag packet from the egress.

- To continue to adopt the VLAN tag priority specified by the client network, you can configure priority replication to configure an outer tag the same as the inner tag.
- If the SP network requires the transmission of packets based on the priority of the outer tag, you also need to configure priority replication to set the CoS of the outer tag to the specified value.

Caution

- The priority of the VID change policies in a descending order: ACL-based > C-Tag-based > port-based.
- When a member is added to/deleted from an aggregation port, the policy of adding or modifying the VID configured on the aggregation port is deleted and needs to be reconfigured. You are advised to configure the VIP policy on an aggregation port after configuring the members.
- You must configure the tunnel port and the interface connected to the public network to permit packets with specified VLAN IDs (including the native VLAN ID) in the outer tag to pass through.

1.5.3 Procedure

(1) Create an SP VLAN.

a Enter the privileged EXEC mode.

enable

b Enter the global configuration mode.

configure terminal

c Create a group of SP VLANs.

vlan range *svidA,svidB*

Only VLAN 1 exists by default.

d Return to the global configuration mode.

exit

(2) Configure the dot1q-tunnel port and allow SP VLAN packets to pass through.

a Enter the interface configuration mode of the port connected to the client network.

interface *interface-type interface-number*

b Configure the port as a dot1q-tunnel port.

switchport mode dot1q-tunnel

An interface works in access mode by default.

c Configure to add the SP VLAN to the untagged VLAN list of the dot1q-tunnel port.

switchport dot1q-tunnel allowed vlan { [**add**] **untagged** *svidA,svidB* | **remove** *other-vlan-id* }

By default, the allowed VLAN of dot1q-tunnel port is untagged VLAN 1.

(3) Configure the C-Tag-based QinQ encapsulation rule. The packets of different client VLANs are allocated to different SP VLANs for transmission through a tunnel.

a The packets of client VLAN *cvid-lista* are allocated to the SP VLAN *svidA*.

dot1q outer-vid *svidA* **register inner-vid** *cvid-lista*

No policy of adding an outer VID based on the inner VID of packets is configured by default.

b The packets of client VLAN *cvid-listb* are allocated to the SP VLAN *svidB*.

dot1q outer-vid *svidB* **register inner-vid** *cvid-listb*

No policy of adding the outer VID based on the inner VID of packets is configured by default.

1.6 Configuring Priority Replication or Priority Mapping

1.6.1 Overview

This function configures the CoS value of the outer tag based on the user priority value (CoS) of the client VLAN tag by replication or mapping.

- Priority replication

By default, the user priority (PRI) field of the Ethernet frame tag is 0, namely, the normal flow. To make the specified packet preferentially processed and transmitted, configure this field to change the priority of the packet. The value of this field corresponds to the CoS value in the QoS policy. The CoS-based QoS policy can be configured to ensure the processing priority of a service.

If the VLAN tag of a client packet is configured with the CoS value of the user priority, and the QoS priority policy is configured in the SP network for the CoS value, the CoS value for the VLAN tag of client packet can be replicated as the CoS value of the outer tag, so that the client packet can be transmitted transparently and the QoS priority policy provided by the SP network can be used.

- Priority mapping

When an outer VLAN tag is added, the user priority of the outer VLAN tag can be set according to the user priority of the inner VLAN tag through priority mapping so that the network QoS priority policy can be used after the packet is encapsulated with an outer VLAN tag.

The SP network sets different QoS policies for multiple CoS values, corresponding to different service flow services. To make some client services be preferentially transmitted in the SP network, the priority mapping function can be configured. The CoS value of the outer VLAN tag can be specified according to the CoS value of the client inner VLAN tag, so that the packet can enjoy the QoS policy of the SP network after being encapsulated with the outer VLAN tag.

1.6.2 Restrictions and Guidelines

- Priority replication and priority mapping cannot be enabled on the same interface at the same time.
- Priority replication or priority mapping of client tags can be configured only on the dot1q-tunnel port.

1.6.3 Configuration Tasks

The following configuration tasks are mutually exclusive. Select one task.

- Configuring Priority Replication
- Configuring Priority Mapping

1.6.4 Configuring Priority Replication

1. Overview

It is assumed that the SP network is configured with the QoS policy based on the user priority value (CoS) of client VLAN tag. If the CoS value of an inner tag is 3, the CoS value of the outer tag can be set to 3 by

configuring priority replication, so that the packet can enjoy the QoS policy with a priority of 3 as the inner tag after being encapsulated with the outer VLAN tag.

Configure priority replication to apply the QoS policy provided by the SP network for the client VLAN tag.

2. Restrictions and Guidelines

Priority replication has a higher priority than trusted QoS but lower than ACL data flow-based QoS.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the interface configuration mode.

interface *interface-type interface-number*

(4) Configure the CoS priority for trusted packets of the interface.

mls qos trust cos

The default trust mode of an interface is untrust.

(5) Configure priority replication.

inner-priority-trust enable

The priority replication function is disabled by default.

1.6.5 Configuring Priority Mapping

1. Overview

It is assumed that the SP network is set with the QoS policy based on the user priority value (CoS) of client VLAN tag. Suppose the CoS value of an inner tag is 3. The CoS value of the outer tag can be set to 5 by configuring priority mapping (CoS-to-CoS), so that the packet can enjoy the QoS policy with a priority of 5 after being encapsulated with the outer VLAN tag.

If the user priority of the outer VLAN tag needs to be set according to the VLAN tag of the client to flexibly apply the QoS priority, configure the priority mapping policy.

2. Restrictions and Guidelines

- Priority mapping takes precedence over QoS.
- If no trust mode is configured (namely, **Trust None**) or the configured trust mode is not matched with priority mapping, the configuration of priority mapping does not take effect.
- If no priority mapping is configured, the following mapping is used by default, and the value ranges of *inner-cos-value* and *outer-cos-value* are from 0 to 7.

Table 1-1 Default Priority Mapping

Inner CoS	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Outer CoS	0	1	2	3	4	5	6	7

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the interface configuration mode.

interface *interface-type interface-number*

(4) Configure the CoS priority for trusted packets of the interface.

mls qos trust cos

The default trust mode of an interface is untrust.

(5) (Optional) Disable priority replication.

no inner-priority-trust enable

The priority replication function is disabled by default.

(6) Configure priority mapping.

dot1q-tunnel cos *inner-cos-value* **remark-cos** *outer-cos-value*

No priority mapping rule is set by default.

1.7 Configuring TPID

1.7.1 Overview

An Ethernet frame tag consists of four fields: Tag Protocol Identifier (**TPID**, written as **Type** in the frame format map), User Priority (**PRI**), **CFI**, and **VLAN ID**.

The TPID value is 0x8100 by default, while the devices of some vendors set the TPID value for the outer tag of packets to 0x9100 or other values. To be compatible with these devices, this device supports port-based TPID modification for packets.

You can configure the TPID value of ports independently. When forwarding packets, these ports replace the TPID in the outer VLAN tag of the packets with the value set by the user, ensuring compatibility with the TPIDs of different vendors.

Purpose: The interface on the PE that connects to the public network needs to be configured as an uplink port. Configure the TPIDs in the tags on SP network devices to realize TPID compatibility with third-party devices. Perform the configuration in interface configuration mode of the uplink port.

1.7.2 Restrictions and Guidelines

- If a PE is connected to a third-party device on which the TPID is not the default value 0x8100 of IEEE 802.1Q, you need to configure the TPID on the interface connected to the third-party device.

Caution

Do not set the TPID to the following well-known types: 0x0806 (ARP), 0x0200 (PUP), 0x8035 (RARP), 0x0800 (IP), 0x86DD (IPv6), 0x8863/0x8864 (PPPoE), 0x8847/0x8848 (MPLS), 0x8137 (IPX/SPX), 0x8000 (IS-IS), 0x8809 (LACP), 0x888E (802.1x), 0x88A7 (cluster), and 0x0789 (reserved).

- TPIDs can be configured in interface configuration mode and global configuration mode.

1.7.3 Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enter the interface configuration mode of the port of the PE that is connected to the public network.

interface *interface-type interface-number*

(4) Configure this port as an uplink port.

switchport mode uplink

An interface works in access mode by default.

(5) Configure the TPID value.

frame-tag tpid *tpid*

The TPID value is 0x8100 by default.

1.8 Configuring Layer-2 Protocol Tunneling

1.8.1 Overview

BPDU tunneling is a layer-2 tunneling function based on the Layer 2 Tunnel Protocol (L2TP).

When the tunnel port of a PE receives STP BPDU packets from the client network, due to their special destination MAC address, the PE will not process them as ordinary client packets, but will recognize them as BPDU packets and not forward them. It will also calculate the spanning tree according to the content of BPDU. This may affect the topology of the SP network. Similarly, when the tunnel interlace of a PE receives GVRP BPDU packets from the client network, the PE will not forward them, but will update the VLAN information according to the packet content. This may affect the topology of the SP network.

When the STP or GVRP function is also enabled for the SP device, the BPDU tunnel function needs to be used to transmit the STP and GVRP packets of the client network in the SP network without affecting the SP network.

When the BPDU packets on the client network enter the tunnel port of the PE, the destination MAC address is changed to the private address and then the packets are forwarded in the SP network. When the packets reach the PE at the other end, the destination MAC address is changed to the public address and the packets are transmitted on the client network at the other end. This achieves transmission of the BPDU packets

through the tunnel in the SP network, preventing the client network and the SP network from interfering with each other.

1.8.2 Restrictions and Guidelines

- If STP or GVRP is not enabled for the SP device, you can configure **bridge-frame forwarding protocol bpdu** to enable transparent transmission of BPDU packets.
- To make protocol tunneling effective on an interface, you need to first enable protocol tunneling in global configuration mode. When the protocol tunneling function on an interface takes effect, the interface does not participate in protocol calculation. If the tunnel port receives a packet whose destination MAC address is a special multicast address, there is an error in the networking and the packet will be dropped directly.

1.8.3 Configuration Tasks

The configuration includes the following tasks:

- Configuring STP Packet Tunneling
- Configuring GVRP Packet Tunneling

1.8.4 Configuring STP Packet Tunneling

1. Overview

If the STP function is enabled for the SP device and STP BPDU packets of the client network need to be transmitted, this function must be configured.

2. Restrictions and Guidelines

- The STP tunneling function takes effect only when it is enabled in global configuration mode and interface configuration mode respectively.
- The optional tunnel addresses of STP packets are 01d0.f800.0005 (default value), 011a.a900.0005, 010f.e200.0003, 0100.0ccd.cdd0, 0100.0ccd.cdd1, and 0100.0ccd.cdd2.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable the STP tunneling function globally.

l2protocol-tunnel stp

The STP packet tunneling function is disabled by default.

(4) (Optional) Configure a tunnel address in the corresponding protocol.

l2protocol-tunnel stp tunnel-dmac *mac-address*

The STP packet tunnel address is 01d0.f800.0005 by default.

(5) Enter the interface configuration mode.

interface *interface-type interface-number*

(6) Enable the STP tunneling function on the interface.

l2protocol-tunnel stp enable

The STP packet tunneling function is disabled for the dot1q-tunnel port by default.

1.8.5 Configuring GVRP Packet Tunneling

1. Overview

If the GVRP function is enabled for the SP device and GVRP BPDU packets of the client network need to be transmitted, this function must be configured.

2. Restrictions and Guidelines

- The GVRP tunneling function takes effect only when it is enabled in global configuration mode and interface configuration mode respectively.
- The optional tunnel addresses of GVRP packets are 01d0.f800.0006 (default value) and 011a.a900.0006.

3. Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable the GVRP tunneling function globally.

l2protocol-tunnel gvrp

The GVRP packet tunneling function is disabled by default.

(4) (Optional) Configure a tunnel address in the corresponding protocol.

l2protocol-tunnel gvrp tunnel-dmac *mac-address*

The GVRP packet tunnel address is 01d0.f800.0006 by default.

(5) Enter the interface configuration mode.

interface *interface-type interface-number*

(6) Enable the GVRP tunneling function on the interface.

l2protocol-tunnel gvrp enable

The GVRP packet tunneling function is disabled for the dot1q-tunnel port by default.

1.9 Monitoring

Run the **show** commands to check the running status of a configured function to verify the configuration effect.

This section also describes the **debug** command used for outputting debugging information.

⚠ Caution

System resources are occupied when debugging information is output. Therefore, disable the debugging function immediately after use.

You can run the **clear** commands to clear information.

Caution

Running the **clear** commands may lose vital information and thus interrupt services.

Table 1-1 Monitoring

Command	Purpose
show dot1q-tunnel [**interfaces** *interface-type interface-number*]	Displays whether the dot1q-tunnel function is enabled on the interface.
show interfaces dot1q-tunnel	Displays configuration of the dot1q-tunnel port.
show registration-table [**interfaces** *interface-type interface-number*]	Displays the policy of adding the VID based on the protocol on the dot1q-tunnel port.
show translation-table [**interfaces** *interface-type interface-number*]	Displays the policy of modifying the VID based on the access, trunk, and hybrid ports.
show frame-tag tpid interfaces [*interface-type interface-number*]	Displays the TPID configuration of the interface.
show inner-priority-trust [**interfaces** *interface-type interface-number*]	Displays the configuration of priority replication.
show interfaces [*interface-type interface-number*] **remark**	Displays the configuration of priority mapping.
show l2protocol-tunnel { **gvrp** | **stp** }	Displays the layer-2 tunnel configuration.
debug bridge qinq	Enables the QinQ debugging function.

1.10 Configuration Examples

1.10.1 Configuring Basic QinQ to Implement Layer-2 VPN

1. Requirements

An SP provides a VPN for Company A and Company B, as shown in Figure 1-1. Basic QinQ is enabled on PEs to meet the layer-2 VPN requirement. On the public network, Company A and Company B belong to different VLANs, and client data packets are transmitted on different native VLANs to implement the simple layer-2 VPN.

- Company A uses public VLAN 10. In Area 1, clients are connected through CE-A1 and access the public network through PE 1; in Area 2, clients are connected through CE-A2, and access the public network through PE 2. The tunnel port first encapsulates the client data packets of Company B with one VLAN 10 tag, and then forwards them on the public network.
- Company B uses public VLAN 20. In Area 1, clients are connected through CE-B1 and access the public network through PE 1; in Area 2, clients are connected through CE-B2, and access the public network through PE 2. The tunnel port first encapsulates the client data packets of Company B with one VLAN 20 tag, and then forwards them on the public network.

The client VLANs in Company A and Company B are transparent to the public network. The client VLANs can be reused without conflicts. Therefore, client VLANs do not need to be configured on the SP device, but need to be configured only on the CE device.

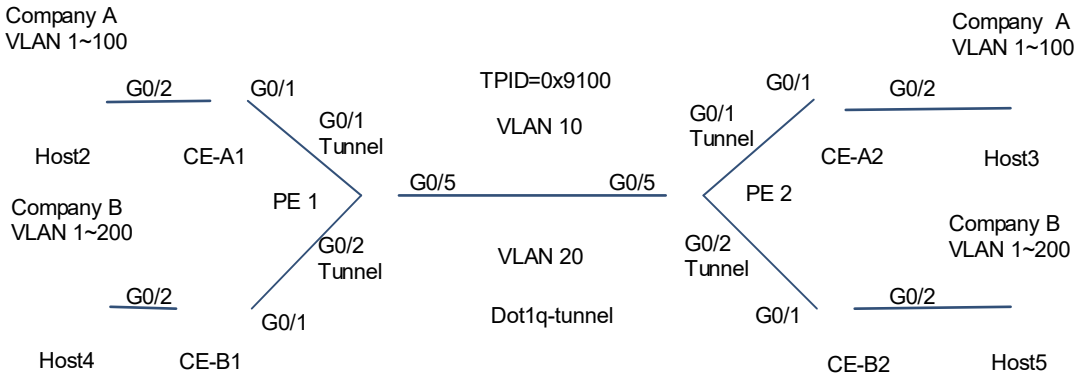
- The office network VLAN of Company A ranges from 1 to 100.
- The office network VLAN of Company B ranges from 1 to 200.

The priority replication and priority mapping functions of CoS can be configured on the tunnel port of the PE, and the QoS policy of CoS is configured to enable the client data packets to enjoy different QoS policies.

Generally, the TPID of the device is 0x8100 specified in IEEE 802.1Q. Because there are some devices in the SP network whose TPID is 0x9100, the TPID value needs to be modified on the uplink port of PE to ensure compatibility with these devices.

2. Topology

Figure 1-1 Basic QinQ Implements Layer-2 VPN



3. Notes

- Configure the interfaces on PE 1 and PE 2 connected to CE-A1, CE-A2, CE-B1 and CE-B2 as tunnel ports. The public network transmission VLAN is the native VLAN of the tunnel port, and belongs to the allowed untagged VLAN list of the port.
- Configure the interface on the PE that connects to the public network as a trunk, hybrid, or uplink port. When the connection interface between SP devices is a trunk or hybrid port, do not configure the native VLAN of the trunk or hybrid port as the native VLAN of dot1q-tunnel port. The reason is that, when the packets of the native VLAN is output from the trunk or hybrid port, the interface will strip the tag from the packet.
- The client VLAN is configured on the CE according to the user's needs, and the uplink port on the CE is configured as trunk, hybrid, or uplink port.

4. Procedure

(1) Configure the basic QinQ transmission channel VLAN 10 of Company A. VLAN 10 is the native VLAN of the tunnel port, and belongs to the allowed untagged VLAN list of the tunnel port.

On PE 1, configure VLAN 10 to implement tunnel transmission for the data of Company A's network.

```
PE1> enable
PE1# configure terminal
```

```
PE1(config)# vlan 10
PE1(config-vlan)# exit
PE1(config)# interface gigabitethernet 0/1
PE1(config-if-GigabitEthernet 0/1)# switchport
PE1(config-if-GigabitEthernet 0/1)# switchport mode dot1q-tunnel
PE1(config-if-GigabitEthernet 0/1)# switchport dot1q-tunnel native vlan 10
PE1(config-if-GigabitEthernet 0/1)# switchport dot1q-tunnel allowed vlan add
untagged 10
PE1(config-if-GigabitEthernet 0/1)# exit
```

On PE 2, configure VLAN 10 to implement tunnel transmission for the data of Company A's network.

```
PE2> enable
PE2# configure terminal
PE2(config)# vlan 10
PE2(config-vlan)# exit
PE2(config)# interface gigabitethernet 0/1
PE2(config-if-GigabitEthernet 0/1)# switchport
PE2(config-if-GigabitEthernet 0/1)# switchport mode dot1q-tunnel
PE2(config-if-GigabitEthernet 0/1)# switchport dot1q-tunnel native vlan 10
PE2(config-if-GigabitEthernet 0/1)# switchport dot1q-tunnel allowed vlan add
untagged 10
PE2(config-if-GigabitEthernet 0/1)# exit
```

(2) Configure the basic QinQ transmission channel VLAN 20 of Company B. VLAN 20 is the native VLAN of the tunnel port, and belongs to the allowed untagged VLAN list of the tunnel port.

On PE 1, configure VLAN 20 to implement tunnel transmission for the data of Company B's network.

```
PE1(config)# vlan 20
PE1(config-vlan)# exit
PE1(config)# interface gigabitethernet 0/2
PE1(config-if-GigabitEthernet 0/2)# switchport
PE1(config-if-GigabitEthernet 0/2)# switchport mode dot1q-tunnel
PE1(config-if-GigabitEthernet 0/2)# switchport dot1q-tunnel native vlan 20
PE1(config-if-GigabitEthernet 0/2)# switchport dot1q-tunnel allowed vlan add
untagged 20
PE1(config-if-GigabitEthernet 0/2)# exit
```

On PE 2, configure VLAN 20 to implement tunnel transmission for the data of Company B's network.

```
PE2(config)# vlan 20
PE2(config-vlan)# exit
PE2(config)# interface gigabitethernet 0/2
PE2(config-if-GigabitEthernet 0/2)# switchport
PE2(config-if-GigabitEthernet 0/2)# switchport mode dot1q-tunnel
PE2(config-if-GigabitEthernet 0/2)# switchport dot1q-tunnel native vlan 20
PE2(config-if-GigabitEthernet 0/2)# switchport dot1q-tunnel allowed vlan add
untagged 20
PE2(config-if-GigabitEthernet 0/2)# exit
```

(3) Configure the interface on the PE for connecting to the public network as a trunk port. VLAN 10 and VLAN 20 need to be added to the allowed VLAN list of the trunk port, but cannot be the native VLAN of the trunk port.

Configure the interface on PE 1 for connecting to the public network as a trunk port, and modify the TPID value of output packets to 0x9100.

```
PE1(config)# interface gigabitethernet 0/5
PE1(config-if-GigabitEthernet 0/5)# switchport mode trunk
PE1(config-if-GigabitEthernet 0/5)# frame-tag tpid 9100
PE1(config-if-GigabitEthernet 0/5)# end
PE1# write
```

Configure the interface on PE 2 for connecting to the public network as a trunk port, and modify the TPID value of output packets to 0x9100.

```
PE2(config)# interface gigabitethernet 0/5
PE2(config-if-GigabitEthernet 0/5)# switchport mode trunk
PE2(config-if-GigabitEthernet 0/5)# frame-tag tpid 9100
PE2(config-if-GigabitEthernet 0/5)# end
PE2# write
```

(4) On the CE, configure the interface for connecting to the PE as a trunk port, and configure the interface for connecting to the user terminal as an access port and add it to the client VLAN.

Create client VLANs 1–100 on CE-A1; configure the interface GigabitEthernet 0/1 for connecting to the PE as a trunk port; configure one client interface GigabitEthernet 0/2 to join VLAN 60 (the configuration of other client interfaces is omitted here).

```
CE-A1> enable
CE-A1# configure terminal
CE-A1(config)# vlan range 1-100
CE-A1(config-vlan)# exit
CE-A1(config)# interface gigabitethernet 0/1
CE-A1(config-if-GigabitEthernet 0/1)# switchport
CE-A1(config-if-GigabitEthernet 0/1)# switchport mode trunk
CE-A1(config-if-GigabitEthernet 0/1)# exit
CE-A1(config)# interface gigabitethernet 0/2
CE-A1(config-if-GigabitEthernet 0/2)# switchport
CE-A1(config-if-GigabitEthernet 0/2)# switchport access vlan 60
```

Create client VLANs 1–100 on CE-A2; configure the interface GigabitEthernet 0/1 for connecting to the PE as a trunk port; configure one client interface GigabitEthernet 0/2 to join VLAN 60 (the configuration of other client interfaces is omitted here).

```
CE-A2> enable
CE-A2# configure terminal
CE-A2(config)# vlan range 1-100
CE-A2(config-vlan)# exit
CE-A2(config)# interface gigabitethernet 0/1
CE-A2(config-if-GigabitEthernet 0/1)# switchport
CE-A2(config-if-GigabitEthernet 0/1)# switchport mode trunk
CE-A2(config-if-GigabitEthernet 0/1)# exit
```

```
CE-A2(config)# interface gigabitethernet 0/2
CE-A2(config-if-GigabitEthernet 0/2)# switchport
CE-A2(config-if-GigabitEthernet 0/2)# switchport access vlan 60
```

Create client VLANs 1–200 on CE-B1; configure the interface GigabitEthernet 0/1 for connecting to the PE as a trunk port; configure one client interface GigabitEthernet 0/2 to join VLAN 60 (the configuration of other client interfaces is omitted here).

```
CE-B1> enable
CE-B1# configure terminal
CE-B1(config)# vlan range 1-200
CE-B1(config-vlan)# exit
CE-B1(config)# interface gigabitethernet 0/1
CE-B1(config-if-GigabitEthernet 0/1)# switchport
CE-B1(config-if-GigabitEthernet 0/1)# switchport mode trunk
CE-B1(config-if-GigabitEthernet 0/1)# exit
CE-B1(config)# interface gigabitethernet 0/2
CE-B1(config-if-GigabitEthernet 0/2)# switchport
CE-B1(config-if-GigabitEthernet 0/2)# switchport mode access
CE-B1(config-if-GigabitEthernet 0/2)# switchport access vlan 60
```

Create client VLANs 1–200 on CE-B2; configure the interface GigabitEthernet 0/1 for connecting to the PE as a trunk port; configure the client interface GigabitEthernet 0/2 to join VLAN 60 (the configuration of other client interfaces is omitted here).

```
CE-B2> enable
CE-B2# configure terminal
CE-B2(config)# vlan range 1-200
CE-B2(config-vlan)# exit
CE-B2(config)# interface gigabitethernet 0/1
CE-B2(config-if-GigabitEthernet 0/1)# switchport
CE-B2(config-if-GigabitEthernet 0/1)# switchport mode trunk
CE-B2(config-if-GigabitEthernet 0/1)# exit
CE-B2(config)# interface gigabitethernet 0/2
CE-B2(config-if-GigabitEthernet 0/2)# switchport
CE-B2(config-if-GigabitEthernet 0/2)# switchport mode access
CE-B2(config-if-GigabitEthernet 0/2)# switchport access vlan 60
```

5. Verification

(1) Display the interface configuration of PEs.

Display the interface configuration of PE 1.

```
PE1# show interfaces switchport
Interface                       Switchport Mode      Access Native Protected
VLAN lists
------------------------------- ---------- --------- ------ ------ ---------
----------------
GigabitEthernet 0/1             enabled    TUNNEL    1      10     Disabled
1,10
```

```
GigabitEthernet 0/2              enabled    TUNNEL    1      20     Disabled
1,20
GigabitEthernet 0/5              enabled    TRUNK     1      1      Disabled
ALL
```

Display the interface configuration of PE 2.

```
PE2# show interfaces switchport
Interface                        Switchport Mode      Access Native Protected
VLAN lists
-------------------------------- ---------- --------- ------ ------ ---------
----------------
GigabitEthernet 0/1              enabled    TUNNEL    1      10     Disabled
1,10
GigabitEthernet 0/2              enabled    TUNNEL    1      20     Disabled
1,20
GigabitEthernet 0/5              enabled    TRUNK     1      1      Disabled
ALL
```

(2) Display the tunnel port configuration of PEs.

Display the tunnel port configuration of PE 1.

```
PE1# show interfaces dot1q-tunnel
========Interface Gi0/1========
Native vlan: 10
Allowed vlan list:1,10,
Tagged vlan list:
========Interface Gi0/2========
Native vlan: 20
Allowed vlan list:1,20,
Tagged vlan list:
```

Display the tunnel port configuration of PE 2.

```
PE2# show interfaces dot1q-tunnel
========Interface Gi0/1========
Native vlan: 10
Allowed vlan list:1,10,
Tagged vlan list:
========Interface Gi0/2========
Native vlan: 20
Allowed vlan list:1,20,
Tagged vlan list:
```

(3) Display the uplink port configuration of PEs.

Display the uplink port configuration of PE 1.

```
PE1# show frame-tag tpid
Ports         Tpid
------------  ----------
Gi0/5         0x9100
```

Display the uplink port configuration of PE 2.

```
PE2# show frame-tag tpid
Ports         Tpid
------------  ----------
Gi0/5         0x9100
```

(4) Display the CE configuration of Company A.

Display the configuration of CE-A1.

```
CE-A1(config)# show interface switchport
Interface                       Switchport Mode      Access Native Protected
VLAN lists
------------------------------- ---------- --------- ------ ------ ---------
----------------
GigabitEthernet 0/1             enabled    TRUNK     1      1      Disabled
ALL
GigabitEthernet 0/2             enabled    ACCESS    60     1      Disabled
ALL
```

Display the configuration of CE-A2.

```
CE-A2(config)#show interface switchport
Interface                       Switchport Mode      Access Native Protected
VLAN lists
------------------------------- ---------- --------- ------ ------ ---------
----------------
GigabitEthernet 0/1             enabled    TRUNK     1      1      Disabled
ALL
GigabitEthernet 0/2             enabled    ACCESS    60     1      Disabled
ALL
```

(5) Take Hosts 2 and 3 as examples to verify that clients in Company A can implement layer-2 interworking.

Configure an IP address for Host 2.

```
Host2> enable
Host2# configure terminal
Host2(config)#interface gigabitethernet 0/1
Host2(config-if-GigabitEthernet 0/1)#no switchport
Host2(config-if-GigabitEthernet 0/1)#ip address 192.168.60.2/24
```

Configure an IP address for Host 3.

```
Host3> enable
Host3# configure terminal
Host3(config)#interface gigabitethernet 0/1
Host3(config-if-GigabitEthernet 0/1)#no switchport
Host3(config-if-GigabitEthernet 0/1)#ip address 192.168.60.3/24
```

Verify that Host 2 can ping through Host 3.

```
Host2# ping 192.168.60.3
Sending 5, 100-byte ICMP Echoes to 192.168.60.3, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 3/4/6 ms.
```

(6) Take Hosts 4 and 5 as examples to verify that clients in Company B can implement layer-2 interworking.

Configure an IP address for Host 4.

```
Host4> enable
Host4# configure terminal
Host4(config)#interface gigabitethernet 0/1
Host4(config-if-GigabitEthernet 0/1)#no switchport
Host4(config-if-GigabitEthernet 0/1)#ip address 192.168.60.4/24
```

Configure an IP address for Host 5.

```
Host5> enable
Host5# configure terminal
Host5(config)#interface gigabitethernet 0/1
Host5(config-if-GigabitEthernet 0/1)#no switchport
Host5(config-if-GigabitEthernet 0/1)#ip address 192.168.60.5/24
```

Verify that Host 4 can ping through Host 5.

```
Host4# ping 192.168.60.5
Sending 5, 100-byte ICMP Echoes to 192.168.60.5, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 3/4/6 ms.
```

(7) Host 2 in Company A cannot ping through Host 5 in Company B, which indicates that clients in different companies cannot implement layer-2 interworking.

```
Host2# ping 192.168.60.5
Sending 5, 100-byte ICMP Echoes to 192.168.60.5, timeout is 2 seconds:
  < press Ctrl+C to break >
.....
Success rate is 0 percent (0/5).
```

6. Configuration Files

- Configuration files of PE 1 and PE 2

```
vlan range 1,10,20
!
interface GigabitEthernet 0/1
 switchport mode dot1q-tunnel
 switchport dot1q-tunnel allowed vlan add untagged 10
 switchport dot1q-tunnel native vlan 10
!
interface GigabitEthernet 0/2
 switchport mode dot1q-tunnel
 switchport dot1q-tunnel allowed vlan add untagged 20
 switchport dot1q-tunnel native vlan 20
!
interface GigabitEthernet 0/5
 switchport mode trunk
```

```
 frame-tag tpid 0x9100
```

- Configuration files of CE-A1 and CE-A2 (Only GigabitEthernet 0/2 is taken as an example of interfaces connected to user terminals; the configuration files of the other interfaces are omitted.)

```
vlan range 1-100
!
interface gigabitethernet 0/1
switchport mode trunk
!
interface gigabitethernet 0/2
switchport access vlan 60
```

- Configuration files of CE-B1 and CE-B2 (Only GigabitEthernet 0/2 is taken as an example of interfaces connected to user terminals; the configuration files of the other interfaces are omitted.)

```
vlan range 1-200

interface gigabitethernet 0/1
switchport mode trunk

interface gigabitethernet 0/2
switchport access vlan 60
```

7. Common Errors

- The same company has different native VLANs on the dot1q-tunnel ports of PE 1 and PE 2, which leads to the failure of interworking.
- The native VLAN of the dot1q-tunnel is not added to the allowed VLAN list of the interface in the untagged form.
- The TPID value of a third-party device is not the default value 0x8100, but the TPID is not configured on the egress connected to the third-party device, and therefore the packet cannot be recognized by the third-party device.

1.10.2 Configuring C-Tag-Based Selective QinQ to Implement Layer-2 VPN and Service Flow Management

1. Requirements

Broadband Internet access and IPTV are important services carried by the metropolitan area network (MAN). MAN SPs allocate different service flows to different SP VLANs to achieve differentiated management, and provide QoS policy services for these VLANs or CoSs.

The basic QinQ function can encapsulate the client data packets received by the tunnel port only with the outer tag with the same VLAN ID, and the VLAN ID of outer tags is the native VLAN of the tunnel port; the basic QinQ cannot meet the requirement for differentiating SP VLANs in multi-service scenarios.

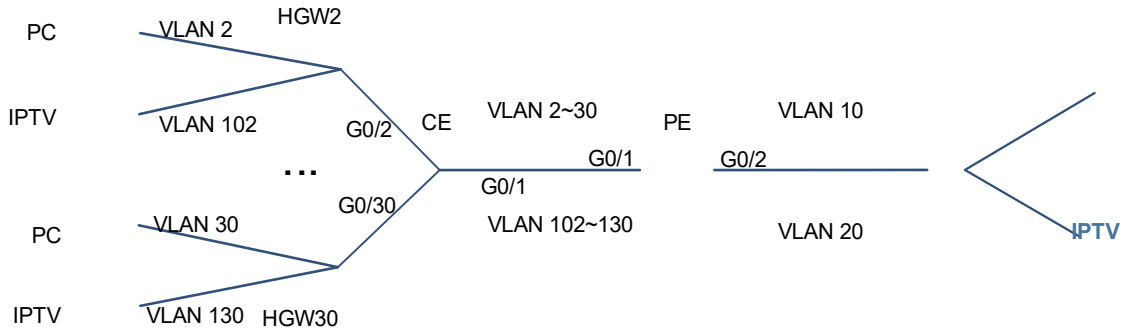
Selective QinQ can flexibly add the SP outer tag (S-Tag) according to the tag (C-Tag) of client packets, encapsulate the client service flows received by the same tunnel port into different SP VLANs, and use the QoS policy of SP network to guarantee reliability while transparently transmitting client data.

As shown in Figure 1-1, devices involved include provider edges (PEs), customer edges (CEs), and home gateways (HGWs) 1–30.

- The broadband Internet access data and IPTV service data are allocated to different VLANs. VLANs 2–30 are for client broadband Internet access service flows, and VLANs 102–130 are for client IPTV service flows.
- On the public network, the broadband Internet access data is transmitted through SP VLAN 10 to implement data communication with the Internet; the IPTV service flow is transmitted through VLAN 20 to implement data communication with the IPTV network. The tunnel port is configured on PE, and different VLANs are encapsulated to distinguish different service data. Client VLANs 2–30 are encapsulated as SP VLAN 10, while client VLANs 102–130 are encapsulated as SP VLAN 20.
- Client network packets carry tags and contain the CoS priority. The SP network enables the priority replication function on PEs, and uses the CoS value for the inner tag of packets in the outer tag so that service flows of the client network can use the QoS policy of the SP network to ensure transmission.

2. Topology

Figure 1-1 Selective QinQ Implements Layer-2 VPN



3. Notes

- Configure the interface GigabitEthernet 0/1 of PE for connecting to CE as tunnel port. Add SP VLANs 10 and 20 to the untagged VLAN list of the tunnel port connected to the client network so that the packets from the peer end return to the tunnel port and the SP tag is stripped when packet output.
- On the downlink interface of PE, configure the selective QinQ function of adding outer VLAN tags based on C-Tag. If the tunnel port receives the frames of client VLANs 2–30 (C-Tag), the frames will be encapsulated with the tag (S-Tag) of SP VLAN 10; if the tunnel port receives the frames of client VLANs 102–130 (C-Tag), the frames will be encapsulated with the tag (S-Tag) of SP VLAN 20.
- The SP network provides the CoS-based QoS priority policy. You can configure priority replication to replicate the CoS value of the client VLAN tag to the CoS of the outer VLAN tag so that the packet encapsulated with the outer tag is transmitted based on the priority policy for the client VLAN tag.
- The interface of PE for connecting to the public network is configured as an uplink port to transmit the data of VLAN 10 and VLAN 20.
- The interface of CE for connecting to the PE is configured as an uplink port to transmit the data of VLANs 2–30 and VLANs 102–130.
- The interface of CE for connecting to HGW is configured as a trunk port, and the VLAN transmitted by HGW device is added to the allowed VLAN list of the trunk port. The clients adopting HGW 2 use VLAN 2 to transmit broadband Internet access data and VLAN 102 to transmit IPTV data. The clients adopting HGW 30 use VLAN 30 to transmit broadband Internet access data and VLAN 130 to transmit IPTV data. The

configurations of other clients are similar.

4. Procedure

(1) Configure the C-Tag-based QinQ encapsulation rule.

Add SP VLAN 10 and VLAN 20 to the untagged VLAN list of the tunnel port GigabitEthernet 0/1.

```
PE> enable
PE# configure terminal
PE(config)# vlan range 10,20
PE(config-vlan-range)# exit
PE(config)# interface gigabitethernet 0/1
PE(config-if-GigabitEthernet 0/1)# switchport
PE(config-if-GigabitEthernet 0/1)# switchport mode dot1q-tunnel
PE(config-if-GigabitEthernet 0/1)# switchport dot1q-tunnel allowed vlan add
untagged 10,20
```

If the tunnel port receives the frames of client VLANs 2–30 (C-Tag), the frames will be encapsulated with the tag (S-Tag) of SP VLAN 10.

```
PE(config-if-GigabitEthernet 0/1)# dot1q outer-vid 10 register inner-vid 2-30
```

If the tunnel port receives the frames of client VLANs 102–130 (C-Tag), the frames will be encapsulated with the tag (S-tag) of SP VLAN 20.

```
PE(config-if-GigabitEthernet 0/1)# dot1q outer-vid 20 register inner-vid 102-
130
```

(2) On the tunnel port, replicate the priority of inner tag in the client packet to the outer tag.

```
PE(config-if-GigabitEthernet 0/1)# mls qos trust cos
PE(config-if-GigabitEthernet 0/1)# inner-priority-trust enable
PE(config-if-GigabitEthernet 0/1)# exit
```

(3) Configure the interface on the PE for connecting to the public network as an uplink port.

```
PE(config)# interface gigabitethernet 0/2
PE(config-if-GigabitEthernet 0/2)# switchport mode uplink
PE(config-if-GigabitEthernet 0/2)# end
PE# write
```

(4) Create a client VLAN on CE, and configure the interface for connecting PE as an uplink port.

```
CE> enable
CE# configure terminal
CE(config)# vlan range 2-30,102-130
CE(config-vlan)# exit
CE(config)# interface gigabitethernet 0/1
CE(config-if-GigabitEthernet 0/1)# switchport
CE(config-if-GigabitEthernet 0/1)# switchport mode uplink
CE(config-if-GigabitEthernet 0/1)# exit
```

(5) Configure the interface on the CE for connecting to the HGW as a trunk port. Gigabit Ethernet 0/2 and Gigabit Ethernet 0/30 are taken as examples below. The configurations of the other interface are similar.

```
CE(config)# interface gigabitethernet 0/2
```

```
CE(config-if-GigabitEthernet 0/2)# switchport
CE(config-if-GigabitEthernet 0/2)# switchport mode trunk
CE(config-if-GigabitEthernet 0/2)# switchport trunk allowed vlan only 2,102
CE(config-if-GigabitEthernet 0/2)# exit
CE(config)# interface gigabitethernet 0/30
CE(config-if-GigabitEthernet 0/30)# switchport
CE(config-if-GigabitEthernet 0/30)# switchport mode trunk
CE(config-if-GigabitEthernet 0/30)# switchport trunk allowed vlan only 30,130
CE(config-if-GigabitEthernet 0/30)# end
CE# write
```

5. Verification

(1) Display the PE configuration.

Display the interface configuration of the PE.

```
PE# show interfaces switchport
Interface                       Switchport Mode      Access Native Protected
VLAN lists
------------------------------- ---------- --------- ------ ------ ---------
----------------
GigabitEthernet 0/1             enabled    TUNNEL    1      1      Disabled
1,10,20
GigabitEthernet 0/2             enabled    UPLINK    1      1      Disabled
ALL
```

Display the tunnel port configuration of the PE. The outer tag VLAN has been added to the allowed VLAN list of the interface.

```
PE# show interfaces dot1q-tunnel
========Interface Gi0/1========
Native vlan: 1
Allowed vlan list:1,10,20
Tagged vlan list:
```

Verify that the encapsulation policy relationship for adding S-tags based on C-tags is correct.

```
PE# show registration-table
Ports     Type            Outer-VID    Inner-VID-list
------    ----------      ----------   --------------
Gi0/1     Add-outer       10            2-30
Gi0/1     Add-outer       20            102-130
```

Display the priority replication relationship of the interface.

```
PE# show inner-priority-trust interfaces gigabitethernet 0/1
Port   Inner-priority-trust
-----  --------------------
Gi0/1  Enable
```

(2) Display the CE configuration.

Display the interface configuration of the CE.

```
CE# show interfaces switchport
Interface                        Switchport Mode      Access Native Protected
VLAN lists
-------------------------------- ---------- --------- ------ ------ ---------
----------------
GigabitEthernet 0/1              enabled    Uplink    1      1      Disabled
ALL
GigabitEthernet 0/2              enabled    Trunk     1      1      Disabled
2,102
GigabitEthernet 0/3              enabled    Trunk     1      1      Disabled
3,103
……
GigabitEthernet 0/30             enabled    Trunk     1      1      Disabled
30,130
```

6. Configuration Files

- Configuration file of the PE

```
vlan range 1,10,20
!
interface GigabitEthernet 0/1
 switchport mode dot1q-tunnel
 switchport dot1q-tunnel allowed vlan add untagged 10,20
 dot1q outer-vid 100 register inner-vid 2-30
 dot1q outer-vid 200 register inner-vid 102-130
 mls qos trust cos
 inner-priority-trust enable
!
interface GigabitEthernet 0/2
 switchport mode uplink
```

- Configuration file of the CE (Only Gigabit Ethernet 0/2 and Gigabit Ethernet 0/30 are taken as examples of interfaces connected to the HGW. The configurations of the other interfaces are similar.)

```
vlan range 1-30,102-130
!
interface GigabitEthernet 0/1
 switchport mode uplink
!
interface GigabitEthernet 0/2
 switchport mode trunk
 switchport trunk allowed vlan only 2,102
!
interface GigabitEthernet 0/30
 switchport mode trunk
 switchport trunk allowed vlan only 30,130
```

1.10.3 Configuring ACL-Based Selective QinQ to Implementing Layer-2 VPN and Service Flow Management

1. Requirements

If the service flows of client network are not classified according to VLAN, but according to the MAC address, IP address, or protocol type, or there are a large number of old low-end network access devices on the client network and the service flows cannot be effectively distinguished using VLAN ID, the packets from the client network cannot be encapsulated with outer tags based on their C-tags to realize transparent transmission and implement QoS policies.

ACL can classify service flows according to the MAC address, IP address, or protocol type, and selective QinQ can distinguish different service flows using ACL, add and modify outer tags, and assign different service data to different VLANs and implement the VPN channel for transparent transmission of client services. If the SP network provides different QoS service policies for different services, it can guarantee first transmission of high-priority services.

As shown in Figure 1-1, internal networks of the company's branches in two cities are connected to CE 1 and CE 2, and then connected to the SP network through PE 1 and PE 2.

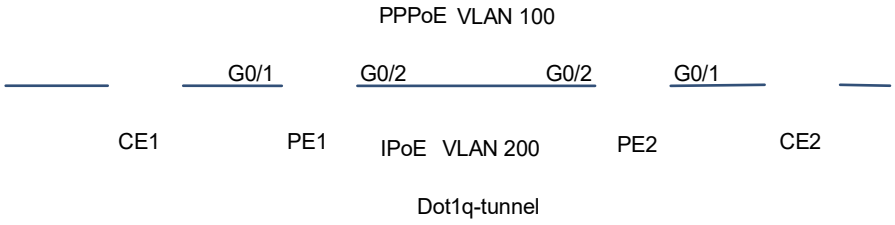
The selective QinQ function based on ACL is used on PE 1 and PE 2 to encapsulate different service data of the company with different outer VLAN tags so that the data can be transmitted through the SP network transparently and the branches can access each other's network.

- ACL 1 identifies service flows of PPPoE type and allocates the service flows to VLAN 100 for transmission.
- ACL 2 identifies service flows of IPTV (IPoE) type and allocates the service flows to VLAN 200 for transmission.

The SP network sets the QoS policy based on CoS. You can modify the CoS values of packets through priority mapping on PE 1 and PE 2 to ensure preferential transmission of service flows of the client network using the QoS policy of the SP network.

2. Topology

Figure 1-1 Selective QinQ Implements Layer-2 VPN



3. Notes

- Configure the ACL policy on PE 1 and PE 2 to distinguish different service flows of the client network: the protocol type of PPPoE packets is 0x8863 or 0x8864, and that of IPoE packets is 0x0800. Configure the interface for connecting to the CE as tunnel port, and configure the selective QinQ function based on ACL. The configuration of PE 2 is similar to that of PE 1. Only PE 1 is taken as an example in the configuration procedure.

- The SP network provides a CoS-based QoS priority policy, and the packets of client network are attached with tags. You can configure priority mapping to modify the CoS value of the outer tag. In this way, the packet is transmitted based on the corresponding QoS priority policy of the SP after being encapsulated with an outer tag, and flexible control is implemented on the packet priority on the tunnel port.

4. Procedure

(1) On the PE, create an ACL for distinguishing service flows.

```
PE1> enable
PE1# configure terminal
PE1(config)# expert access-list extended acl1
PE1(config-exp-nacl)# permit 0x8863 any any
PE1(config-exp-nacl)# permit 0x8864 any any
PE1(config-exp-nacl)# exit
PE1(config)# expert access-list extended acl2
PE1(config-exp-nacl)# permit 0x0800 any any
PE1(config-exp-nacl)# exit
```

(2) Configure the ACL-based QinQ encapsulation rule.

Add the SP VLAN 100 and VLAN 200 to the untagged VLAN list of the tunnel port GigabitEthernet 0/1.

```
PE1(config)# vlan rang 100,200
PE1(config-vlan-range)# exit
PE1(config)# interface gigabitethernet 0/1
PE1(config-if-GigabitEthernet 0/1)# switchport mode dot1q-tunnel
PE1(config-if-GigabitEthernet 0/1)# switchport dot1q-tunnel allowed vlan add
untagged 100,200
```

(3) Configure the outer tag priority of packets according to the inner tag priority of packets.

```
PE1(config-if-GigabitEthernet 0/1)# mls qos trust cos
PE1(config-if-GigabitEthernet 0/1)# no inner-priority-trust enable
PE1(config-if-GigabitEthernet 0/1)# dot1q-tunnel cos 3 remark-cos 5
PE1(config-if)# exit
```

(4) Configure the interface connected to the public network as an uplink port.

```
PE1(config)# interface gigabitethernet 0/2
PE1(config-if-GigabitEthernet 0/2)# switchport mode uplink
PE1(config-if-GigabitEthernet 0/2)# end
PE1# write
```

5. Verification

(1) Display the interface configuration of the PE.

```
PE1# show interfaces switchport
Interface                       Switchport Mode      Access Native Protected
VLAN lists
------------------------------- ---------- --------- ------ ------ ---------
----------------
GigabitEthernet 0/1             enabled    TUNNEL    1      1      Disabled
1,100,200
```

```
GigabitEthernet 0/2              enabled    UPLINK    1      1      Disabled
ALL
```

(2) Display the priority mapping relationship on the interface.

```
PE1# show interfaces gigabitethernet 0/1 remark
Ports        From COS  To COS
------------ --------- --------
Gi0/1        3         5
```

(3) The clients of the same service in the company's branch can realize layer-2 VPN interworking. Specific service data can enjoy a higher transmission priority.

6. Configuration Files

Configuration files of PE 1 and PE 2

```
expert access-list extended acl1
 10 permit 0x8863 any any
 20 permit 0x8864 any any
!
expert access-list extended acl2
 10 permit 0x800 any any
!
vlan range 1,100,200
!
interface GigabitEthernet 0/1
 switchport mode dot1q-tunnel
 switchport dot1q-tunnel allowed vlan add untagged 100,200
 dot1q-Tunnel cos 3 remark-cos 5
 mls qos trust cos
!
 interface GigabitEthernet 0/2
 switchport mode uplink
```

7. Common Errors

- The ACL policy is configured incorrectly. The ACL policy number is not in the parameter range. The ACL name is already taken.
- Priority replication is not disabled before priority mapping is configured.

1.10.4 Configuring Layer-2 Protocol Tunneling Based on QinQ

1. Requirements

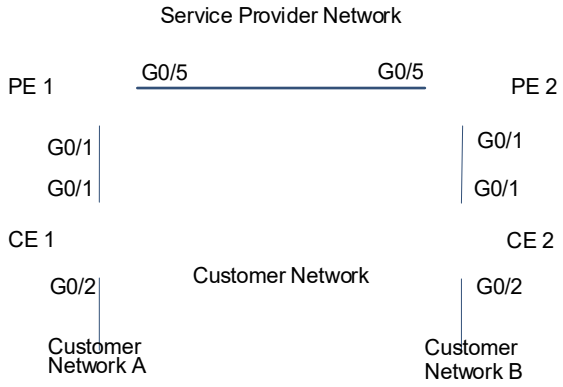
As shown in Figure 1-1, the upper part of the typical topology of QinQ network is the SP network, and the lower part is the client network. The SP network consists of PE 1 and PE 2. Customer Network A and Customer Network B are a user's two sites in different regions. CE 1 and CE 2 are access devices from the client network to the SP network, and access the SP network through PE 1 and PE 2 respectively.

The layer-2 protocol tunneling is configured to transparently transmit layer-2 protocol packets from one end to the other end of the client network so that the layer-2 protocols of the client network and the SP network are calculated separately without interference.

- Configure the STP protocol tunneling function based on user requirements to realize tunnel transmission of BPDU packets of the client network in the SP network and to perform unified spanning tree calculation for the cross-regional client network across the SP network.
- Configure the GVRP protocol tunneling function based on user requirements to realize tunnel transmission of GVRP packets of the client network and perform dynamic VLAN configuration of the client network across the SP network.

2. Topology

Figure 1-1 QinQ Implements Layer-2 Protocol Tunneling



3. Notes

- On PE 1 and PE 2, configure the STP and GVRP tunneling functions globally.
- Configure the QinQ encapsulation mode. Basic QinQ is taken as an example (for details, see "1.10.1 Configuring Basic QinQ to Implement Layer-2 VPN") to configure the native VLAN of tunnel port as VLAN 10, and VLAN 10 is used as the public network transmission channel. A client data packet is first encapsulated with one VLAN 10 tag, and then forwarded on the public network.
- Enable the STP and GVRP tunneling functions on the interface GigabitEthernet 0/1.
- Configure the uplink port.

4. Procedure

Both PE 1 and PE 2 need to be configured. The configuration method on the two devices is the same. PE 1 is taken as an example below.

(1) Enable the STP and GVRP tunneling functions globally.

```
PE1> enable
PE1# configure terminal
PE1(config)# l2protocol-tunnel stp
PE1(config)# l2protocol-tunnel gvrp
```

(2) Configure the basic QinQ transmission channel VLAN 10. VLAN 10 is the native VLAN of the tunnel port, and belongs to the allowed untagged VLAN list of the tunnel port.

```
PE1(config)# vlan 10
PE1(config-vlan)# exit
```

```
PE1(config)# interface gigabitethernet 0/1
PE1(config-if-GigabitEthernet 0/1)# switchport
PE1(config-if-GigabitEthernet 0/1)# switchport mode dot1q-tunnel
PE1(config-if-GigabitEthernet 0/1)# switchport dot1q-tunnel native vlan 10
PE1(config-if-GigabitEthernet 0/1)# switchport dot1q-tunnel allowed vlan add
untagged 10
```

(3) Enable the STP and GVRP tunneling functions on the tunnel port.

```
PE1(config-if-GigabitEthernet 0/1)# l2protocol-tunnel stp enable
PE1(config-if-GigabitEthernet 0/1)# l2protocol-tunnel gvrp enable
PE1(config-if-GigabitEthernet 0/1)# exit
```

(4) Configure the uplink port.

```
PE1(config)# interface gigabitethernet 0/5
PE1(config-if-GigabitEthernet 0/5)# switchport
PE1(config-if-GigabitEthernet 0/5)# switchport mode uplink
PE1(config-if-GigabitEthernet 0/5)# end
PE1# write
```

5. Verification

The verification method is the same on PE1 and PE2. PE 1 is taken as an example below.

Verify that the STP tunneling function is enabled globally and on the interface at the same time.

```
PE1# show l2protocol-tunnel stp
L2protocol-tunnel: Stp Enable
L2protocol-tunnel destination mac address: 01d0.f800.0005
GigabitEthernet 0/1 l2protocol-tunnel stp enable
```

Verify that the GVRP tunneling function is enabled globally and on the interface at the same time.

```
PE1# show l2protocol-tunnel gvrp
L2protocol-tunnel: Gvrp Enable
L2protocol-tunnel destination mac address: 01d0.f800.0006
GigabitEthernet 0/1 l2protocol-tunnel gvrp enable
```

6. Configuration Files

Configuration files of PE 1 and PE 2

```
l2protocol-tunnel stp
l2protocol-tunnel gvrp
!
vlan range 1,10
!
interface GigabitEthernet 0/1
 switchport mode dot1q-tunnel
 switchport dot1q-tunnel allowed vlan add untagged 10
 switchport dot1q-tunnel native vlan 10
 l2protocol-tunnel stp enable
 l2protocol-tunnel gvrp enable
!
```

```
interface GigabitEthernet 0/5
 switchport mode uplink
```

7. Common Errors

- The layer-2 tunneling function is not enabled globally and on the interface at the same time, and therefore it fails to take effect.
- In the SP network, the configured BPDU tunnel addresses are inconsistent, and therefore BPDU frames of the client network cannot be correctly transmitted.