

FSOS

QinQ Operation

---

## Contents

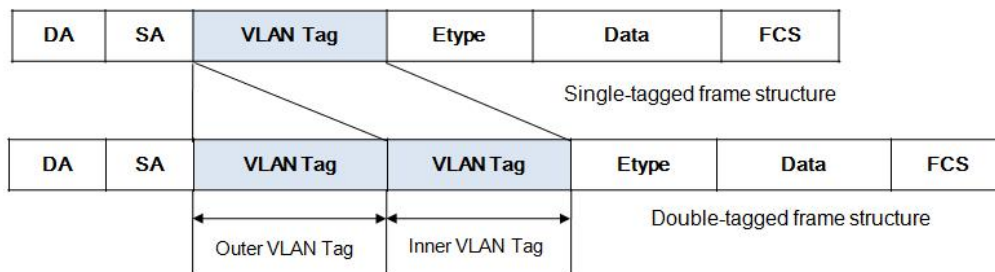
<b>1. QinQ Configuration.....</b>	<b>1</b>
1.1 QinQ Overview.....	1
1.2 Static QinQ Overview.....	4
1.3 Configuration Example for Static QinQ.....	5
1.4 Dynamic QinQ Overview.....	6
1.5 Configuration Example for Dynamic QinQ.....	8
1.6 Adjustable Function of VLAN Tag TPID Value.....	9
1.6.1 Configure TPID Value of VLAN Tag to be Adjustable.....	9
1.6.2 Configuration Example for TPID Value Adjustable.....	10

---

## 1. QinQ Configuration

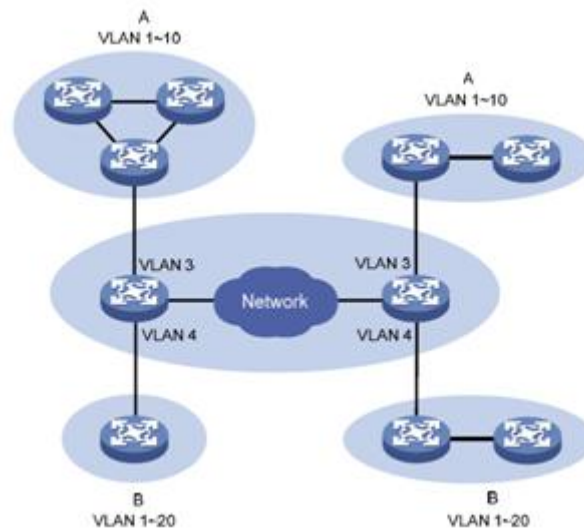
### 1.1 QINQ Overview

In the VLAN tag field defined in IEEE 802.1Q, only 12 bits are used for VLAN IDs, so a switch can support a maximum of 4,094 VLANs. In actual applications, however, a large number of VLANs are required to isolate users, especially in metropolitan area networks (MANs), and 4,094 VLANs are far from satisfying such requirements. The generation of QinQ technology solves the insufficient of VLAN users. The following shows the structure of 802.1Q-tagged and double-tagged Ethernet frames. The QinQ feature enables a device to support up to 4,094 x 4,094 VLANs to satisfy the requirement for the amount of VLANs.



QinQ Ethernet frame structure

The port QinQ feature is a flexible, easy-to-implement Layer 2 VPN technique, which enables the access point to encapsulate an outer VLAN tag in Ethernet frames from customer networks (private networks), so that the Ethernet frames will travel across the service provider's backbone network (public network) with double VLAN tags. The inner VLAN tag is the customer network VLAN tag while the outer one is the VLAN tag assigned by the service provider to the customer. In the public network, frames are forwarded based on the outer VLAN tag only, with the source MAC address learned as a MAC address table entry for the VLAN indicated by the outer tag, while the customer network VLAN tag is transmitted as part of the data in the frames.



### QinQ application

A VLAN tag uses the tag protocol identifier (TPID) field to identify the protocol type of the tag. The value of this field, as defined in IEEE 802.1Q, is 0x8100. The device can identify whether there is corresponded VLAN Tag according to TPID. If configured TPID is the same as the corresponded field, packet is regarded as with VLAN Tag.

The TPID in an Ethernet frame has the same position with the protocol type field in a frame without a VLAN tag. To avoid problems in packet forwarding and handling in the network, you cannot set the TPID value to any of the values in the table below.

common protocol type values

Protocol type	value
ARP	0x0806
PUP	0x0200
RARP	0x8035
IP	0x0800
IPv6	0x86DD
PPPoE	0x8863/0x8864
MPLS	0x8847/0x8848
IPX/SPX	0x8137
IS-IS	0x8000
LACP	0x8809
802.1x	0x888E

GnLink	0x0765
GSTP	0X5524

In QinQ, there are four interface mode: none, customer, customer-no-static and uplink. After enabling QinQ, interface mode will be **none** mode by default.

Four interface modes are suitable for different applications:

1. None, this mode is suitable for device to connect PC directly, and the packet of this interface will try to identify outer VLAN tag and inner VLAN tag; the inner VLAN tag will be stripped before forwarding while whether the outer VLAN tag will be stripped or not is determined by the outer VLAN configuration.
2. Customer, the interface of this mode and the interface of the uplink mode form the static QinQ application. The packet which enter this interface will be considered not include outer VLAN tag and inner VLAN tag, so the packet will take the PVID or the other specified vlan ID as the packet outer VLAN ID; the outer VLAN tag will be stripped before forwarding regardless the outer VLAN configuration.  
Customer interface needs to connect with the interface of user-device. The single VLAN tag enters from this interface will carry 2 VLAN tags when it is forwarded from the uplink interface.
3. customer-no-static, the interface of this mode and the interface of the uplink mode will form the dynamic QinQ application. The packet which enters this interface will try to identify outer VLAN tag, and then match the dynamic QinQ rule according to the VLAN ID of outer VLAN tag. If the dynamic QinQ rule is matched successfully, the packet will be considered not include outer VLAN tag and inner VLAN tag, what is more, it will take the specified VLAN ID as the outer VLAN ID; the outer VLAN tag will be stripped before forwarding regardless the outer VLAN configuration.
4. uplink, the interface of this mode needs to connect with the interface of operator-device. The packet which enters this interface will try to identify outer VLAN tag, and the packet will carry the inner VLAN tag before being forwarded, moreover, whether the outer VLAN tag will be stripped or not is determined by the outer VLAN configuration. The uplink interface judges whether the packet carries tag or not according to the vlan protocol number and the outer-tpid value. If the vlan protocol number is the same as the outer-tpid value, it will be taken as carrying with tag; if the vlan protocol number is the different from the outer-tpid value, it will be taken as not carrying with tag.

---

 Note:

---

---

The customer mode interface cannot communicate with none mode interface.

---

## 1.2 Static QINQ Overview

After enabling dtag function, device will not run static QinQ by default. Only when you configure customer under interface mode can the static QinQ take effect. Under the condition that the interface does not configure insert rule, the packet will be processed according to static QinQ. The following is static QinQ handling process:

### 1. Customer port process flow

Customer port will add vlan tag what is the same as interface pvid to the original packet and forward the packet in this vlan regardless whether the packet carries tag or not, whether the tpid is the same as configured outer-tpid/inner-tpid or not, whether the vid exists in the equipment or not.

### 2. Uplink port process flow

Whether the uplink port will insert a tag or not is decided by whether the packet carries a tag or not. Whether the uplink port carries tag or not is decided by whether the vlan protocol number is the same as global outer-tpid value or not:

- a) If the vlan protocol number is the same as global outer-tpid value, it will be taken as carrying with the tag and then it will forward the packet in the carried vlan without inserting a tag; if the carried vlan does not exists in the system, the packet will not be forwarded;
- b) If the vlan protocol number is different from global outer-tpid value, it will insert a vlan tag according to the pvid firstly, and then add an outer-tpid according to the inner-tpid before forwarding;

### 3. TPID value processing

When the packet forwarded from customer port or uplink port, TPID value is always consistent with the outer-tpid value.

Static QinQ Configuration

Operation	Command	Remarks
Enter global configuration mode	<b>configure terminal</b>	-
Enable(disable) global QinQ	<b>[no]dtag</b>	required
Enter port	<b>interface ethernet device/slot/port</b>	-

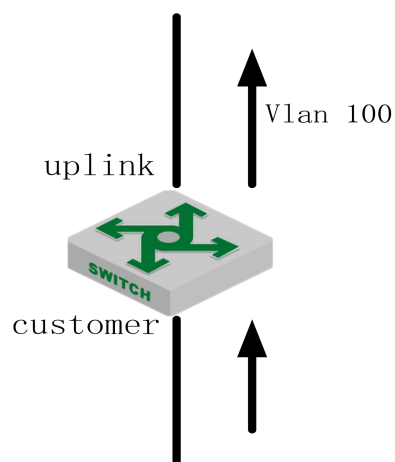
configuration mode		
Specify the port QinQ mode	<b>dtag mode {customer   uplink}</b>	required, customer and uplink interface form static qinq
Restore port default mode	<b>no dtag mode</b>	Optional none mode by default
Display QinQ configuration	<b>show dtag</b>	any mode

### 1.3 Configuration Example for Static QinQ

#### 1. Network requirements:

It requires packet add outer vlan 100 tag when forwarding from uplink interface regardless whether the packet carries VLAN tag or not and what the tag it will be.

Networking is as follows:



#### 2. Configuration steps:

# enable global dtag

```
Switch(config)#dtag
# enter port 1, configure the QinQ port mode to be customer
Switch(config)#interface ethernet 0/0/1
Switch(config-if-ethernet-0/0/1)#dtag mode customer
# configure the default vlan of port 1 to be 100
Switch(config-if-ethernet-0/0/1)#switchport default vlan 100
```

```
# enter port 2, configure the QinQ port mode to be uplink
Switch(config)#interface ethernet 0/0/2
Switch(config-if-ethernet-0/0/2)#dtag mode uplink
# configure port 2 to be trunk port
Switch(config-if-ethernet-0/0/2)#switchport mode trunk
```

### 3.Result validation:

No matter whether the receiving packet of customer interface carries the vlan tag or not as well as what the tag value it will be, the packet will add the tag head of vlan 100 when transmit from uplink port.

## 1.4 Dynamic QinQ Overview

Uplink port can be able to configure insert rule. If the packet tag conforms to insert rule, it will be dealt according to the corresponding rule regardless what the configured outer-tpid value it will be.

Customer-no-static port will not only check whether the packet carries tag or not but also judge whether it conforms to the insert rule. If the packet carries tag and it conforms to the insert rule, it will be processed according to the dynamic QinQ, or it will be processed according to 802.1Q rule. The following is the detailed process:

1. Compared the ingress packet with the configured outer-tpid firstly. If the packet tpid is different from the configured outer-tpid, the packet will be processed according to 802.1Q rule, adding a tag according to the interface PVID;
2. If packet's tpid is the same as outer-tpid but the packet VID is not in the insert range, the packet will be processed according to 802.1Q rule, performing unvarnished transmission or discard;
3. If packet's tpid is the same as outer-tpid and the packet VID is in the insert range, the packet will be processed according to the dynamic QinQ:
  - a) The packet will be processed according to insert rule regardless whether the packet VID exists or not, adding an outer tag according to the insert rule;
  - b) If the outer vlan which corresponding to the insert rule does not exist, or the vlan exists but not includes the ingress, the packet will not perform any processing but discard it;
  - c) If the outer vlan which corresponding to the insert rule do exist but the vlan not



includes any ingress port, the packet will be processed according to insert rule firstly, that is, this packet will learn the MAC address according to the outer vlan, but this packet will be discarded later.

Dynamic QINQ Configuration

Operation	Command	Operation
Enter global configuration mode	<b>configure terminal</b>	-
Enable(disable) global QinQ	<b>[no]dtag</b>	required
Enter port configuration mode	<b>interface ethernet</b> <i>device/slot/port</i>	-
QinQ mode of specified port	<b>dtag mode {customer   uplink}</b>	required
Restore the QinQ default mode of the interface	<b>no dtag mode</b>	Optional None mode by default
Configure insert rule	<b>dtag insert</b> <i>start-vlan end-vlan service-vlan</i>	required
Delete insert rule	<b>no dtag insert {all   start-vlan end-vlan }</b>	optional
Display QinQ configuration	<b>show dtag</b>	any mode

 Note:

Static QinQ and dynamic QinQ can be able to enable at the same time. However, when configuring the dynamic QinQ rule, the dynamic QinQ rule will be determined according to the outer VLAN ID of static QinQ. Therefore, dynamic QinQ interface mode is recommended to be configured as customer-no-static.

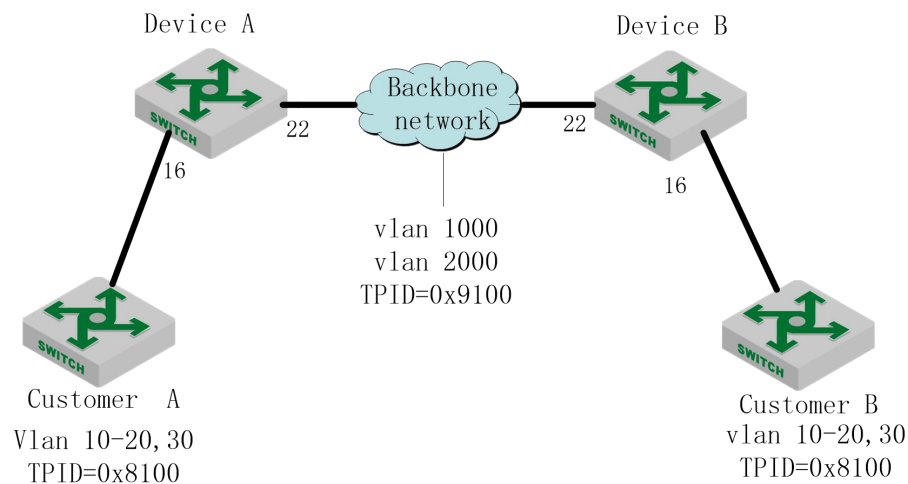
## 1.5 Configuration Example for Dynamic QINQ

### 1. Network requirements

Device A and device B act as the operator's network to access the device. Customer A and customer B act as user ends network to access the device. Device A connects with device B via trunk port, allow vlan 1000 and vlan 2000 to pass through; use the equipment of other equipment manufacturers among device A and device B, TPID=0x9100; It requires implementing the following requirements:

- a) After forwarded by the operator's network vlan1000, Customer A vlan10-20 packet can be able to communicate with customer B vlan 10-20 packet;
- b) After forwarded by the operator's network vlan2000, Customer A vlan30 packet can be able to communicate with customer B vlan 30 packet;

Network diagram is as follows:



### 2. Configuration steps

Device A configuration:

```
# create vlan 1000 and vlan 2000 , and then add them to user-port (port 16) and
service-port ( port 22)
```

```
Switch(config)#vlan 1000,2000
```

```
Switch(config-if-vlan)#switchport ethernet 0/0/16 ethernet 0/0/22
```

```
# configure the service-port to be trunk port
```

```
Switch(config)#interface ethernet 0/0/22
```

```
Switch(config-if-ethernet-0/0/22)#switchport mode trunk
```

```
# enable global qinq function
```

```
Switch(config)#dtag
```

```
Configure outer tag tpid to be 0x9100
```

```

Switch(config)#dtag outer-tpid 9100
# configure user-port qinq mode
Switch(config)#interface ethernet 0/0/16
Switch(config-if-ethernet-0/0/16)# dtag mode customer
# configure service-port qinq mode
Switch(config)#interface ethernet 0/0/22
Switch(config-if-ethernet-0/0/22)#dtag mode uplink

# configure the user-port insert rule
Switch(config)#interface ethernet 0/0/16
Switch(config-if-ethernet-0/0/16)#dtag insert 10 20 1000
Set SVLAN successfully.
Switch(config-if-ethernet-0/0/16)#dtag insert 30 30 2000
Set SVLAN successfully.
  
```

The configuration of Device B is the same as Device A, so no repeat here.

### 3.Result validation

After Customer A vlan10-20 packet passed through device A, the packet will carry the tag of vlan=1000, tpid=0x9100 to forward.

After Customer A vlan30 packet passed through device A, the packet outer carry the tag of vlan=2000, tpid=0x9100 to forward.

## 1.6 Adjustable Function of VLAN Tag TPID Value

### 1.6.1 Configure TPID Value of VLAN Tag to be Adjustable

TPID value of VLAN Tag		
Operation	Command	Operation
Enter global configuration mode	<b>configure terminal</b>	-
Configure inner tag TPID value	<b>dtag inner-tpid <i>tpid</i></b>	Optional 0x8100 by default
Restore default inner-tpid value	<b>no dtag inner-tpid</b>	optional
Enter port	<b>interface ethernet</b>	-

configuration mode	<i>device/slot/port</i>	
Configure the outer tag TPID value	<b>dtag outer-tpid <i>tpid</i></b>	Optional 0x8100 by default
Restore default outer-tpid value	<b>no dtag outer-tpid</b>	optional
Display tpid value	<b>show dtag</b>	any mode

 Note:

1. When double tag packet communicates with CPU, packet inner vlan tpid information should be in line with equipment configuration. Otherwise, it can't not perform any communication.
2. When forwarded from uplink interface, the packet TPID value will be in line with the egress outer-tpid value.

## 1.6.2 Configuration Example for TPID Value Adjustable

```

Modify the inner tag tpid to be 0x9100
Switch(config)#dtag inner-tpid 9100
Modify the inner tag tpid to be 0x9200
Switch(config-if-ethernet-0/0/1)#dtag outer-tpid 9200
Display TPID information
Switch(config)#show dtag
Current dtag status: enabled
inner-tpid      : 0x9100
outer-tpid     : 0x9200
cpu inner-tag  : vid 1 priority 0
interface      : dtag-mode
  
```