Configuring Q-in-Q VLAN Tunnels

Information About Q-in-Q Tunnels

This chapter describes how to configure IEEE 802.1Q-in-Q VLAN tunnels and Layer 2 protocol tunneling on Cisco NX-OS devices.

A Q-in-Q VLAN tunnel enables a service provider to segregate the traffic of different customers in their infrastructure, while still giving the customer a full range of VLANs for their internal use by adding a second 802.1Q tag to an already tagged frame.

Q-in-Q 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 the 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 of 4096 of the 802.1Q specification.

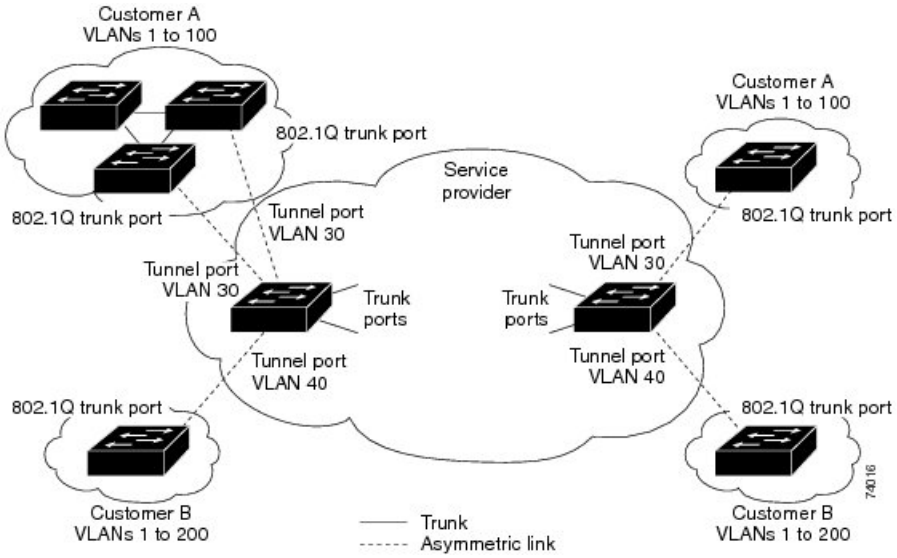
Note Q-in-Q is supported on port channels. To configure a port channel as an asymmetrical link, all ports in the port channel must have the same tunneling configuration.

Using the 802.1Q tunneling feature, service providers can use a single VLAN to support customers who have multiple VLANs. Customer VLAN IDs are preserved and the traffic from different customers is segregated within the service-provider infrastructure even when they appear to be on the same VLAN. The 802.1Q

tunneling expands the VLAN space by using a VLAN-in-VLAN hierarchy and tagging the tagged packets. A port configured to support 802.1Q tunneling is called a tunnel port. When you configure tunneling, you assign a tunnel port to a VLAN that is dedicated to tunneling. Each customer requires a separate VLAN, but that VLAN supports all of the customer's VLANs.

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

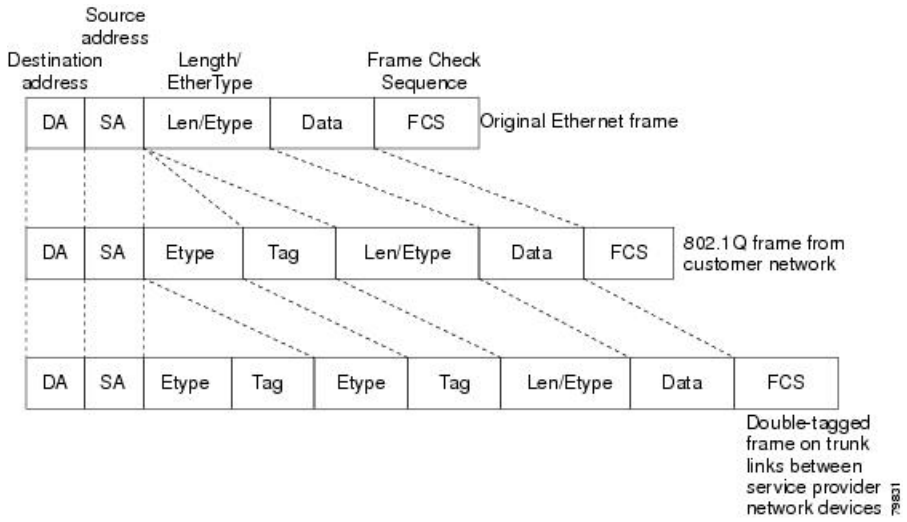
Figure 1: 802.1Q-in-Q Tunnel Ports



Packets that enter the tunnel port on the service-provider edge switch, which are already 802.1Q-tagged with the appropriate VLAN IDs, are encapsulated with another layer of an 802.1Q tag that contains a VLAN ID that is unique to the customer. The original 802.1Q tag from the customer is preserved in the encapsulated packet. Therefore, packets that enter the service-provider infrastructure are double-tagged.

The outer tag contains the customer's access VLAN ID (as assigned by the service provider), and the inner VLAN ID is the VLAN of the incoming traffic (as assigned by the customer). This double tagging is called tag stacking, Double-Q, or Q-in-Q as shown in the figure below.

Figure 2: Untagged, 802.1Q-Tagged, and Double-Tagged Ethernet Frames



By using this method, the VLAN ID space of the outer tag is independent of the VLAN ID space of the inner tag. A single outer VLAN ID can represent the entire VLAN ID space for an individual customer. This technique allows the customer's Layer 2 network to extend across the service provider network, potentially creating a virtual LAN infrastructure over multiple sites.

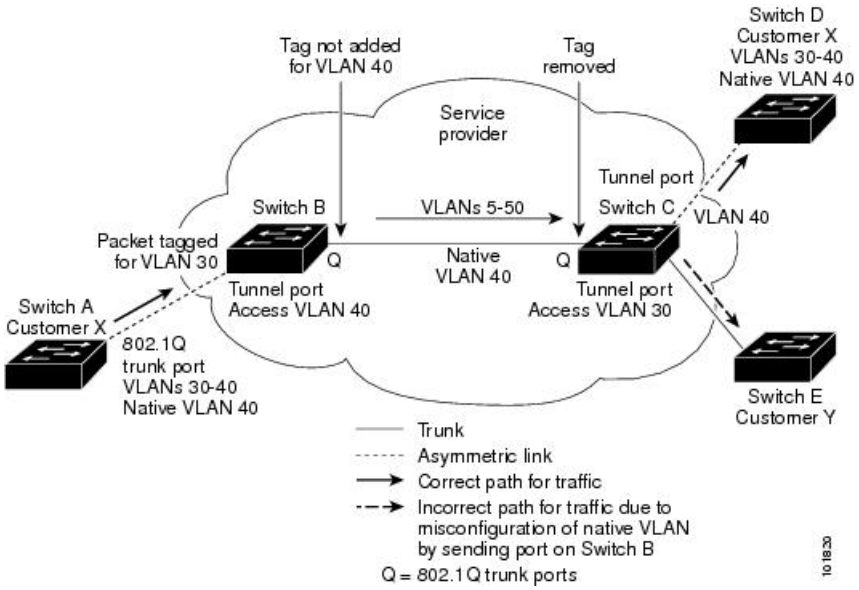
Note Hierarchical tagging, or multi-level dot1q tagging Q-in-Q, is not supported.

Native VLAN Hazard

When configuring 802.1Q tunneling on an edge switch, you must use 802.1Q trunk ports for sending out packets into the service-provider network. However, packets that go through the core of the service-provider network might be carried through 802.1Q trunks, ISL trunks, or nontrunking links. When 802.1Q trunks are used in these core switches, the native VLANs of the 802.1Q trunks must not match any native VLAN of the dot1q-tunnel port on the same switch because traffic on the native VLAN is not tagged on the 802.1Q transmitting trunk port.

In the figure below, 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 that 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 802.1Q tag is not added to tagged packets that are 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.

Figure 3: Native VLAN Hazard



These are a couple ways to solve the native VLAN problem:

- Configure the edge switch so that all packets going out an 802.1Q trunk, including the native VLAN, are tagged by using the vlan dot1q tag native command. 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.

Note The **vlan dot1q tag native** command is a global command that affects the tagging behavior on all trunk ports.

- 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.

Information About Layer 2 Protocol Tunneling

Customers at different sites connected across a service-provider network need to run various Layer 2 protocols to scale their topology to include all remote sites, as well as the local sites. The Spanning Tree Protocol (STP) must run properly, and every VLAN should build a proper spanning tree that includes the local site and all remote sites across the service-provider infrastructure. The Cisco Discovery Protocol (CDP) must be able to discover neighboring Cisco devices from local and remote sites, and the VLAN Trunking Protocol (VTP) must provide consistent VLAN configuration throughout all sites in the customer network.

You can configure the switch to allow multi-tagged BPDUs on a tunnel port. If you enable the **l2protocol tunnel allow-double-tag** command, when a multi-tagged customer BPDU enters the tunnel port, the original 802.1Q tags from the customer traffic is preserved and an outer VLAN tag (customer's access VLAN ID, as assigned by the service-provider) is added in the encapsulated packet. Therefore, BPDU packets that enter the service-provider infrastructure are multi tagged. When the BPDUs leave the service-provider network, the outer tag is removed and the original multi-tagged BPDU is sent to the customer network.

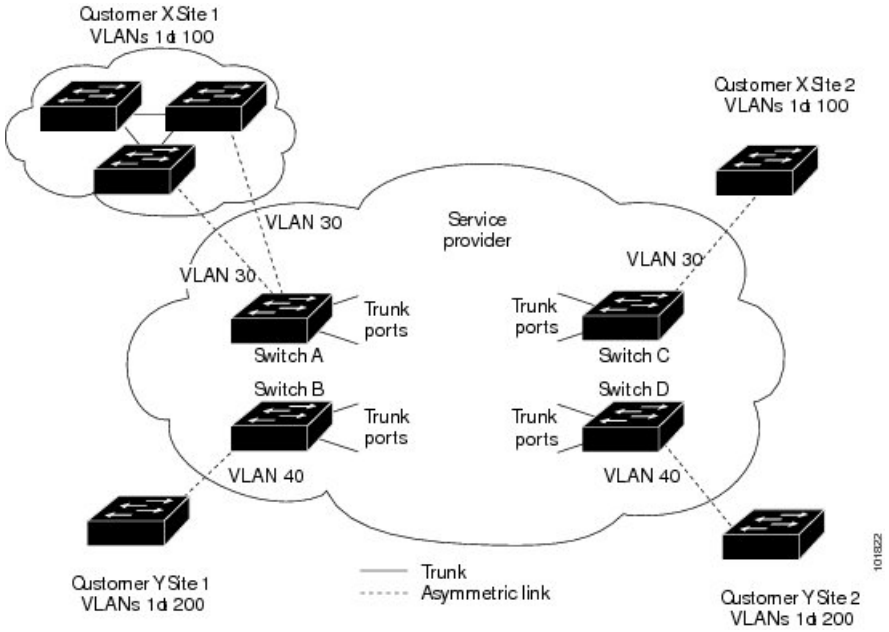
When protocol tunneling is enabled, edge switches on the inbound side of the service-provider infrastructure encapsulate Layer 2 protocol packets with a special MAC address and send them across the service-provider network. Core switches in the network do not process these packets, but forward them as normal packets. Bridge protocol data units (BPDUs) for CDP, STP, or VTP cross the service-provider infrastructure and are delivered to customer switches on the outbound side of the service-provider network. Identical packets are received by all customer ports on the same VLANs.

If protocol tunneling is not enabled on 802.1Q tunneling ports, remote switches at the receiving end of the service-provider network do not receive the BPDUs and cannot properly run STP, CDP, 802.1X, and VTP. When protocol tunneling is enabled, Layer 2 protocols within each customer's network are totally separate from those running within the service-provider network. Customer switches on different sites that send traffic through the service- provider network with 802.1Q tunneling achieve complete knowledge of the customer's VLAN.

Note Layer 2 protocol tunneling works by tunneling BPDUs in the software. A large number of BPDUs that come into the supervisor will cause the CPU load to go up. You might need to make use of software rate limiters to reduce the load on the supervisor CPU. See Configuring Thresholds for Layer 2 Protocol Tunnel Ports, on page 19.

For example, in the figure below, Customer X has four switches in the same VLAN that are connected through the service-provider network. If the network does not tunnel BPDUs, switches on the far ends of the network cannot properly run the STP, CDP, 802.1X, and VTP protocols.

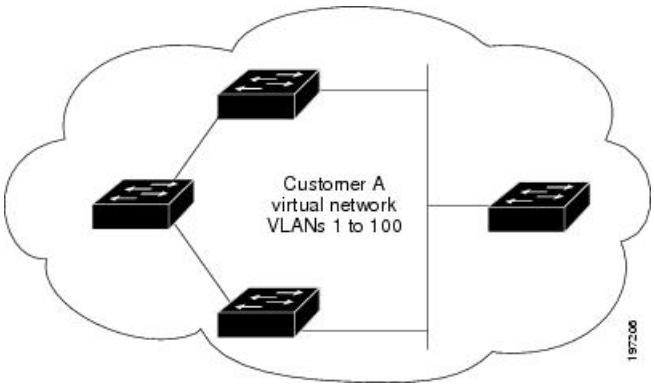
Figure 4: Layer 2 Protocol Tunneling



In the preceding example, STP for a VLAN on a switch in Customer X, Site 1 will build a spanning tree on the switches at that site without considering convergence parameters based on Customer X's switch in Site 2.

The figure below shows the resulting topology on the customer's network when BPDU tunneling is not enabled.

Figure 5: Virtual Network Topology Without BPDU Tunneling



Selective Q-in-Q with Multiple Provider VLANs

Selective Q-in-Q with multiple provider VLANs is a tunneling feature that allows user-specific range of customer VLANs on a port to be associated with one specific provider VLAN and enables you to have multiple customer VLAN to provider VLAN mappings on a port. Packets that come in with a VLAN tag that matches any of the configured customer VLANs on the port are tunneled across the fabric using the properties of the service provider VLAN. The encapsulated packet carries the customer VLAN tag as part of the Layer 2 header of the inner packet.

About Port VLAN Mapping on VLANs (Translating incoming VLANs)

When a service provider has multiple customers connecting to the same physical switch using the same VLAN encapsulation, but they should not be on the same Layer 2 segment, translating the incoming VLAN to a unique VLAN/VNI is the right way to extending the segment.

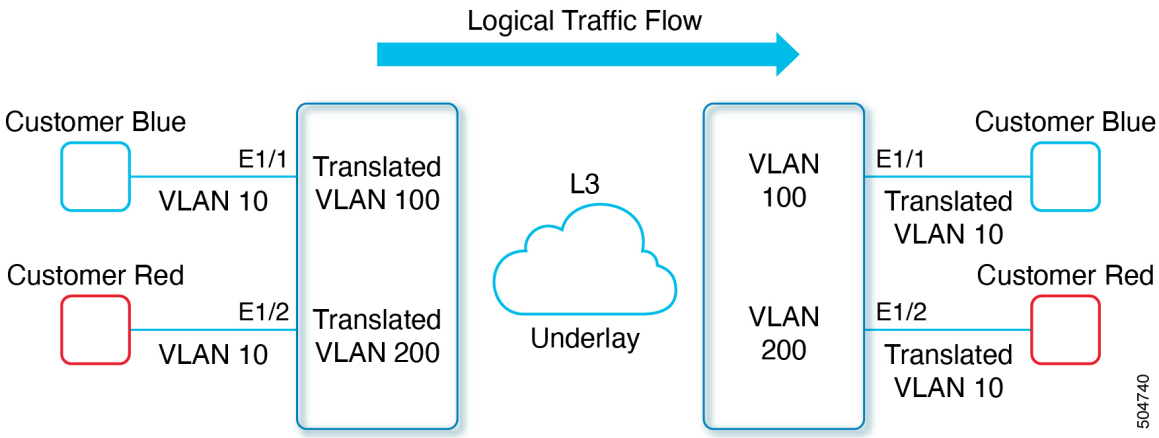
Beginning with Cisco NX-OS Release 10.3(3)F, Port VLAN mapping on non-VXLAN VLANs is supported on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2, C9408 platform switches and Cisco Nexus 9500 switches with 9700-EX/FX/GX line cards.

In the figure below two customers, Blue and Red are connecting to the leaf using VLAN 10 as their encapsulation.

In this example VLAN 10 for Customer Blue (on interface E1/1) is mapped/translated to VLAN 100, and VLAN 10 for customer Red (on interface E1/2) is mapped to VLAN 200.

On the other leaf, this mapping is applied in reverse. Incoming VLAN 100 is mapped to VLAN 10 on Interface E1/1 and VLAN 200 is mapped to VLAN 10 on Interface E1/2.

Figure 6: Logical Traffic Flow



You can configure VLAN translation between the ingress (incoming) VLAN and a local (translated) VLAN on a port. For the traffic arriving on the interface where VLAN translation is enabled, the incoming VLAN is mapped to a translated VLAN.

On the underlay, the inner dot1q is deleted, and switched over to the non-VXLAN network. On the outgoing interface, where VLAN translation is configured, the traffic is converted to the original VLAN and egressed out. Refer to the VLAN counters on the translated VLAN for the traffic counters and not on the ingress VLAN.

Guidelines and Limitations for Q-in-Q tunneling and Layer 2 Protocol Tunneling

Q-in-Q tunnels and Layer 2 tunneling have the following configuration guidelines and limitations:

- Q-in-Q should be configured on the customer-facing interface of the service provider's edge device. If an Ethernet frame ingresses a Cisco Nexus 9000 series switch, the switch cannot encapsulate the frame with two 802.1Q headers within a single forwarding decision. Similarly, if a Q-in-Q-encapsulated Ethernet frame needs to egress a Cisco Nexus 9000 series switch without any 802.1Q headers, the switch cannot decapsulate two 802.1Q headers from the Ethernet frame within a single forwarding decision.
- Mapping multiple VLANs is supported.
- Multi-tagged BPDUs are supported on the Cisco Nexus 93108TC-EX and 93180YC-EX switches. We support up to three tags.
- Selective Q-in-Q tunneling is not supported with multi-tagged BPDU.
- Only multi-tagged CDP and STP BPDUs are supported.
- The inner-most tag must always be 0x8100.
- Multiple selective Q-in-Q tags are not supported. That is, Q-in-Q does not support multiple SP tags on a single interface.
- Switches in the service-provider network must be configured to handle the increase in MTU size due to Q-in-Q tagging.

- MAC address learning for Q-in-Q tagged packets is based on the outer VLAN (Service Provider VLAN) tag. Packet forwarding issues might occur in deployments where a single MAC address is used across multiple inner (customer) VLANs.
- Layer 3 and higher parameters cannot be identified in tunnel traffic (for example, Layer 3 destination and source addresses). Tunneled traffic cannot be routed.
- The **system dot1q-tunnel transit** or **system dot1q-tunnel transit vlan** *provider_vlan_list* command have the following limitations:
  - These commands are required on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2 switches and 9500 switches with 9700-EX/FX/GX line cards if the device is configured with Q-in-Q, Selective Q-in-Q or Selective Q-in-Q with multiple provider VLAN features.
  - It is required that you configure the **system dot1q-tunnel transit** or **system dot1q-tunnel transit vlan** *provider_vlan_list* command on ToR or modular devices. Beginning with Cisco NX-OS Release 9.3(5), the **system dot1q-tunnel transit vlan** *provider_vlan_list* command is supported.
  - It is required that you configure the **system dot1q-tunnel transit** or the **system dot1q-tunnel transit vlan** *provider_vlan_list* command on vPC switches or non-vPC switches.
  - Layer 2 frames that exit trunk ports will always be tagged, even with the native VLAN of the port if these commands have been configured.
  - The MPLS, GRE, and IP-in-IP functionalities will not function effectively in conjunction with the Q-in-Q tunneling features if these commands have been configured on the switch.
- Cisco Nexus 9000 Series devices can provide only MAC-layer ACL/QoS for tunnel traffic (VLAN IDs and src/dest MAC addresses).
- You should use MAC address-based frame distribution.
- Asymmetrical links do not support the Dynamic Trunking Protocol (DTP) because only one port on the link is a trunk. You must configure the 802.1Q trunk port on an asymmetrical link to trunk unconditionally.
- You cannot configure the 802.1Q tunneling feature on ports that are configured to support private VLANs. Private VLAN are not required in these deployments.
- You must disable IGMP snooping on the tunnel VLANs.
- You should enter the vlan dot1Q tag native command to maintain the tagging on the native VLAN and drop untagged traffic. This command prevents native VLAN misconfigurations.
- You must manually configure the 802.1Q interfaces to be edge ports.
- IGMP snooping is not supported on the inner VLAN.
- Q-in-Q is not supported on the uplink ports of Cisco Nexus 9332PQ, 9372PX, 9372TX, and 93120TX switches and Cisco Nexus 9396PX, 9396TX, and 93128TX switches with the N9K-M6PQ or N9K-M12PQ generic expansion module (GEM).
- Q-in-Q tunnels might be affected by the limitations of the Application Leaf Engine (ALE) uplink ports on Cisco Nexus 9300 and 9500 Series devices: Limitations for ALE Uplink Ports
- Q-in-Q tunneling is not supported on the following Application Spine Engine 2 (ASE2) and Application Spine Engine 3 (ASE3) based Cisco Nexus switches.
  - ASE2 - N9236C, N9272Q, N92304QC, and N92300Y

  - ASE3 - N92160YC-X

- Q-in-Q tagging is not supported.
- Layer 2 protocol tunneling is not supported on Cisco Nexus 9500 Series switches with N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX line cards.
- Cisco Nexus 9500 Series switches with N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX line cards, Q-in-Q is supported only on port or port-channel Layer 2 Access VLAN Edge devices.
- FEX configuration is not supported on Q-in-Q ports.
- If the command **l2potocol tunnel stp** is configured on a tunnel interface, the VLAN that you configure on the service provider must be different from that of the customer network.

Guidelines and Limitations for Selective Q-in-Q with Multiple Provider VLANs

- For selective Q-in-Q with multiple provider VLANs, all the existing limitations and guidelines for selective Q-in-Q apply.
- Beginning with Cisco NX-OS Release 9.3(5), selective Q-in-Q with multiple provider VLANs feature is supported on Cisco Nexus N9K-C9316D-GX, N9K-C93600CD-GX, N9K-C9364C-GX switches.
- Selective Q-in-Q with multiple provider VLANs feature is supported on Nexus 9300-EX, 9300-FX, 9300-FX2, 9300-FX3 switches.
- When you enable multiple provider VLANs on a vPC port channel, you must make sure that the configuration is consistent across the vPC peers.
- We recommended not to allow provider VLANs on a regular trunk.
- Only allow native VLANs and provider VLANs on the trunk interface allowed VLAN list of a multiple provider VLAN interface.
- Port to VLAN mappings (for example: switchport vlan mapping 10 20) is not supported on a port that is configured for selective Q-in-Q with multiple provider VLANs.
- Private VLAN is not supported on a port that is configured for selective Q-in-Q with multiple provider VLANs.
- Only Layer 2 switching is supported.
- Routing on provider VLANs is not supported.
- FEX is not supported for selective Q-in-Q with multiple provider VLANs.
- Selective Q-in-Q with multiple provider VLANs commands not DME-ized
- When VLAN1 is configured as native VLAN with selective Q-in-Q and selective Q-in-Q with multiple provider tag, traffic on the native VLAN gets dropped. Do not configure VLAN1 as native VLAN when the port is configured with the selective Q-in-Q. When VLAN1 is configured as customer VLAN, then the traffic on VLAN1 gets dropped.

Guidelines and Limitations for Combined Access Port Feature set

- Beginning Cisco NX-OS Release 9.3(3), Combined Access Port Feature set is supported on Cisco Nexus C9348GC-FXP switches with IPv4 underlay.
- The Combined Access Port Feature set consists of the following features:
  - Private VLAN (with secondary isolated)
  - Selective Q-in-Q
  - Port-Security
- All the guidelines and limitations for PVLAN and selective Q-in-Q are applicable for Combined Access Port Feature set also.
- Port mode **private-vlan trunk secondary**is supported on Combined Access Port Feature set.
- When you enable Combined Access Port Feature set on a vPC port channel, you must ensure that the configuration is consistent across the vPC peers.
- We recommend that you enter **system dot1q-tunnel transit** when running the Combined Access Port Feature set.
- Port VLAN mapping (for example: **switchport vlan mapping 10 20**) is not supported.
- Only layer 2 switching is supported on Selective Q-in-Q
- Only routing is supported on native VLAN of the Combined Access Port Feature

Guidelines and Limitations for Port VLAN Mapping on VLANs

The following are the guidelines and Limitations for Port VLAN Mapping:

- Beginning with Cisco NX-OS Release 10.3(3)F, Port VLAN mapping on VLANs is supported on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2, C9408 platform switches and Cisco Nexus 9500 switches with 9700-EX/FX/GX line cards.
- The ingress (incoming) VLAN does not need to be configured on the switch as a VLAN. The translated VLAN must be configured.
- All Layer 2 source address learning and Layer 2 MAC destination lookup occurs on the translated VLAN. See the VLAN counters on the translated VLAN and not on the ingress (incoming) VLAN.
- Port VLAN mapping routing supports configuring an SVI on the translated VLAN.
- The following example shows incoming VLAN 10 being mapped to local VLAN 100:

```
interface ethernet1/1
switchport vlan mapping 10 100
```

- The following is an example of overlapping VLAN for PV translation. In the first statement, VLAN-102 is a translated VLAN. In the second statement, VLAN-102 the VLAN where it is translated to VLAN-103:

```
interface ethernet1/1
switchport vlan mapping 101 102
switchport vlan mapping 102 103
```

- When adding a member to an existing port channel using the force command, the "mapping enable" configuration must be consistent. For example:

```
Int po 101
switchport vlan mapping enable
switchport vlan mapping 101 10
switchport trunk allowed vlan 10

int eth 1/8
/***No configuration***/
```

Note The switchport VLAN mapping enable command is supported only when the port mode is trunk.

- VLAN mapping helps with VLAN localization to a port, scoping the VLANs per port. A typical use case is in the service provider environment where the service provider leaf switch has different customers with overlapping VLANs that come in on different ports. For example, customer A has VLAN 10 coming in on Eth 1/1 and customer B has VLAN 10 coming in on Eth 2/2.
- Port VLAN mapping does not coexist with PVLAN.
- If the **inherit port-profile** command is configured on a PV interface, use the **no inherit port-profile** <*profile name*> command to detach and then execute the **no switchport vlan mapping all** command.
- If the **system dot1q-tunnel transit vlan provider_vlan_list** command is globally configured on the switch, do not set the provider VLAN as the native or access port VLAN for any other trunk or access port on the system. It is expected to choose provider VLANs other than the native VLANs on the system.

Configuring Q-in-Q Tunnels and Layer 2 Protocol Tunneling

Creating a 802.1Q Tunnel Port

You create the dot1q-tunnel port using the **switchport mode** command.

Note You must set the 802.1Q tunnel port to an edge port with the **spanning-tree port type edge** command. The provider VLAN membership of the port is changed using the **switchport access vlan** *vlan-id* command.

You should disable IGMP snooping on the access VLAN allocated for the dot1q-tunnel port to allow multicast packets to traverse the Q-in-Q tunnel.

For seamless packet forwarding and preservation of all VLAN tags on pure transit boxes in the SP cloud that have no Q-in-Q encapsulation or decapsulation requirement, configure the system-wide **system dot1q-tunnel transit** or **system dot1q-tunnel transit vlan** *provider_vlan_list* command. To remove the configuration, use the **no system dot1q-tunnel transit** or **system dot1q-tunnel transit vlan** *provider_vlan_list* command.

For the supported platforms and limitations of the **system dot1q-tunnel transit** or **system dot1q-tunnel transit vlan** *provider_vlan_list* command, see Guidelines and Limitations for Q-in-Q tunneling and Layer 2 Protocol Tunneling , on page 7 section.

Before you begin

You must first configure the interface as a switchport.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface ethernet** *slot/port*
3. switch(config-if)# **switchport**
4. switch(config-if)# **switchport mode dot1q-tunnel**
5. switch(config-if)# **spanning-tree port type edge**
6. switch(config-if)# **switchport access vlan** *vlan-id*
7. (Optional) switch(config-if)# **no switchport mode dot1q-tunnel**
8. switch(config-if)# **exit**
9. (Optional) switch(config)# **show dot1q-tunnel** [**interface** *if-range*]
10. (Optional) switch(config)# **no shutdown**
11. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport**	Sets the interface as a Layer 2 switching port.
Step 4	switch(config-if)# **switchport mode dot1q-tunnel**	Creates a 802.1Q tunnel on the port. The port will go down and reinitialize (port flap) when the interface mode is changed. BPDU filtering is enabled and CDP is disabled on tunnel interfaces.
Step 5	switch(config-if)# **spanning-tree port type edge**	Designates the port as a spanning-tree edge port.
Step 6	switch(config-if)# **switchport access vlan** *vlan-id*	Configures the Provider access VLAN value.
Step 7	(Optional) switch(config-if)# **no switchport mode dot1q-tunnel**	Disables the 802.1Q tunnel on the port.
Step 8	switch(config-if)# **exit**	Exits configuration mode.
Step 9	(Optional) switch(config)# **show dot1q-tunnel** [**interface** *if-range*]	Displays all ports that are in dot1q-tunnel mode. Optionally, you can specify an interface or range of interfaces to display.
Step 10	(Optional) switch(config)# **no shutdown**	Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.

	Command or Action	Purpose
Step 11	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to create an 802.1Q tunnel port:

```
switch# configure terminal
switch(config)# interface ethernet 7/1
switch(config-if)# switchport
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# spanning-tree port type edge
switch(config-if)# switchport access vlan vlan 10
switch(config-if)# exit
switch(config)# exit
switch# show dot1q-tunnel
```

Configuring VLAN Mapping for Selective Q-in-Q on a 802.1Q Tunnel Port

To configure VLAN mapping for selective Q-in-Q on a 802.1Q tunnel port, complete the following steps.

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

You must set the 802.1Q tunnel port to an edge port with the **spanning-tree port type edge** command. The provider VLAN membership of the port is changed using the **switchport access vlan** *vlan-id* command.

You should disable IGMP snooping on the access VLAN allocated for the dot1q-tunnel port to allow multicast packets to traverse the Q-in-Q tunnel.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-id*
3. switch(config-if)# **switchport mode dot1q-tunnel**
4. switch(config-if)# **spanning-tree port type edge**
5. switch(config-if)# **switchport access vlan** *vlan-id*
6. switch(config-if)# **switchport vlan mapping** *vlan-id-range* **dot1q-tunnel** *outer vlan-id*
7. switch(config-if)# **exit**
8. switch# **show interfaces** *interface-id* **vlan mapping**
9. switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **interface** *interface-id*	Enters interface configuration mode for the interface connected to the service provider network. You can enter a physical interface or an EtherChannel port channel.
Step 3	switch(config-if)# **switchport mode dot1q-tunnel**	Configures the interface as a tunnel port.
Step 4	switch(config-if)# **spanning-tree port type edge**	Designates the port as a spanning-tree edge port.
Step 5	switch(config-if)# **switchport access vlan** *vlan-id*	Configures the Provider access VLAN value.
Step 6	switch(config-if)# **switchport vlan mapping** *vlan-id-range* **dot1q-tunnel** *outer vlan-id*	Enters the VLAN IDs to be mapped: • vlan-id-range—The customer VLAN ID range (C-VLAN) entering the switch from the customer network. The range is from 1 to 4094. You can enter a string of VLAN-IDs. • outer vlan-id—Enter the outer VLAN ID (S-VLAN) of the service provider network. The range is from 1 to 4094.
Step 7	switch(config-if)# **exit**	Exits the configuration mode.
Step 8	switch# **show interfaces** *interface-id* **vlan mapping**	Verifies the configuration.
Step 9	switch# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Use the **no switchport vlan mapping** *vlan-id-range* **dot1q-tunnel** *outer vlan-id* command to remove the VLAN mapping configuration.

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

Example

```
switch(config)# interface gigabitethernet0/1
switch(config-if)# switchport vlan mapping 1-5 dot1q-tunnel 100
switch(config-if)# spanning-tree port type edge
switch(config-if)# switchport access vlan 10
Switch(config-if)# exit
```

Configuring Selective Q-in-Q with Multiple provider VLANs

Before you begin

You must configure provider VLANs

You must disable spanning-tree on the trunk port using the **spanning-tree bpdufilter enable** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-id*
3. switch(config if)# **switchport**
4. switch(config-if)# **switchport mode trunk**
5. switch(config-if)# **spanning-tree bpdufilter enable**
6. switch(config-if)# **switchport trunk native vlan** *vlan-id*
7. switch(config-if)# **switchport vlan mapping** *vlan-id-range* **dot1q-tunnel** *outer vlan-id*
8. switch(config-if)# **switchport trunk allowed vlan** *vlan_list*
9. switch(config-if)# **exit**
10. switch(config-if)# **show interfaces** *interface-id* **vlan mapping**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-id*	Enters interface configuration mode for the interface connected to the service provider network. You can enter a physical interface or an EtherChannel port channel.
Step 3	switch(config if)# **switchport**	Sets the interface as a Layer 2 switching port.
Step 4	switch(config-if)# **switchport mode trunk**	Sets the interface as a Layer 2 trunk port.
Step 5	switch(config-if)# **spanning-tree bpdufilter enable**	Disables the sending and processing of spanning-tree BPDUs on this interface.
Step 6	switch(config-if)# **switchport trunk native vlan** *vlan-id*	Sets the native VLAN for the 802.1Q trunk. Valid values are from 1 to 4094. The default value is VLAN1.
Step 7	switch(config-if)# **switchport vlan mapping** *vlan-id-range* **dot1q-tunnel** *outer vlan-id*	Enter the VLAN IDs to be mapped: • vlan-id-range—The customer VLAN ID range(C-VLAN) entering the switch from the customer network. The range is from 1 to 4094. You can enter a string of VLAN-IDs. • outer vlan-id—Enter the outer VLAN ID (S-VLAN) of the service provider network. The range is from 1 to 4094.
Step 8	switch(config-if)# **switchport trunk allowed vlan** *vlan_list*	Sets the allowed VLANs for the trunk interface.
Step 9	switch(config-if)# **exit**	Exits the configuration mode.
Step 10	switch(config-if)# **show interfaces** *interface-id* **vlan mapping**	Verifies the mapping configuration.

The following example shows how to configure selective Q-in-Q with multiple provider VLANs:

Example

```
switch# sh run int e1/1

interface Ethernet1/1
  switchport
  switchport mode trunk
  switchport trunk native vlan 2
  switchport vlan mapping 3-400 dot1q-tunnel 400
  switchport vlan mapping 401-800 dot1q-tunnel 401
  switchport vlan mapping 801-1200 dot1q-tunnel 10
  switchport vlan mapping 1201-1600 dot1q-tunnel 1400
  switchport vlan mapping 1601-2000 dot1q-tunnel 9
  switchport vlan mapping 2001-2400 dot1q-tunnel 3000
  switchport vlan mapping 2401-2800 dot1q-tunnel 2099
  switchport vlan mapping 2801-3200 dot1q-tunnel 2800
  switchport vlan mapping 3201-3600 dot1q-tunnel 3967
  switchport vlan mapping 3601-4000 dot1q-tunnel 600
  spanning-tree bpdufilter enable
  switchport trunk allowed vlan 2,9-10,400-401,600,1400,2099,2800,3000,3967

switch# show interface e1/1 vlan mapping
Interface Eth1/1:
Original VLAN                     Translated VLAN
---------------                   -----------------
3                                        400
4                                        400
5                                        400
6                                        400
7                                        400
8                                        400
9                                        400
10                                       400
11                                       400
12                                       400
13                                       400
14                                       400
15                                       400
16                                       400
17                                       400
18                                       400
19                                       400
20                                       400

switch# show consistency-checker selective-qinq interface e1/1
Fetching ingressVlanXlate entries from slice:0 HW
Fetching ingressVlanXlate entries from slice:1 HW
Performing port specific checks for intf Eth1/1
Port specific selective QinQ checks for interface   Eth1/1 : PASS

Switch#
```

Changing the EtherType for Q-in-Q

The switch default EtherType is 0x8100 for 802.1Q and Q-in-Q encapsulations. EtherType cannot be configured to 0x9100, 0x9200 and 0x88a8 on the switchport interface.

Enabling the Layer 2 Protocol Tunnel

You can enable protocol tunneling on the 802.1Q tunnel port.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface ethernet** *slot/port*
3. switch(config-if)# **switchport**
4. switch(config-if)# **switchport mode dot1q-tunnel**
5. switch(config-if)# **l2protocol tunnel** [**cdp** | **stp** | **lacp** | **lldp** |**vtp**]
6. (Optional) switch(config-if)# **no l2protocol tunnel** [**cdp** | **stp** | **lacp** | **lldp** |**vtp**]
7. switch(config-if)# **exit**
8. (Optional) switch(config)# **no shutdown**
9. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport**	Sets the interface as a Layer 2 switching port.
Step 4	switch(config-if)# **switchport mode dot1q-tunnel**	Creates a 802.1Q tunnel on the port. The port will go down and reinitialize (port flap) when the interface mode is changed. BPDU filtering is enabled and CDP is disabled on tunnel interfaces.
Step 5	switch(config-if)# **l2protocol tunnel** [**cdp** | **stp** | **lacp** | **lldp** |**vtp**]	Enables Layer 2 protocol tunneling. Optionally, you can enable CDP, STP, LACP, LLDP, or VTP tunneling.
Step 6	(Optional) switch(config-if)# **no l2protocol tunnel** [**cdp** | **stp** | **lacp** | **lldp** |**vtp**]	Disables protocol tunneling.
Step 7	switch(config-if)# **exit**	Exits configuration mode.
Step 8	(Optional) switch(config)# **no shutdown**	Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.
Step 9	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to enable protocol tunneling on an 802.1Q tunnel port:

```
switch# configure terminal
switch(config)# interface ethernet 7/1
switch(config-if)# switchport
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# l2protocol tunnel stp
switch(config-if)# exit
switch(config)# exit
```

Configuring Global CoS for L2 Protocol Tunnel Ports

You can specify a Class of Service (CoS) value globally so that ingress BPDUs on the tunnel ports are encapsulated with the specified class.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **l2protocol tunnel cos** *value*
3. (Optional) switch(config)# **no l2protocol tunnel cos**
4. switch(config)# **exit**
5. (Optional) switch# **no shutdown**
6. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **l2protocol tunnel cos** *value*	Specifies a global CoS value on all Layer 2 protocol tunneling ports. The default cos-value is 5.
Step 3	(Optional) switch(config)# **no l2protocol tunnel cos**	Sets the global CoS value to default.
Step 4	switch(config)# **exit**	Exits configuration mode.
Step 5	(Optional) switch# **no shutdown**	Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to specify a global CoS value for the purpose of Layer 2 protocol tunneling:

```
switch# configure terminal
switch(config)# l2protocol tunnel cos 6
switch(config)# exit
```

Configuring Thresholds for Layer 2 Protocol Tunnel Ports

You can specify the port drop and shutdown value for a Layer 2 protocol tunneling port.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface ethernet** *slot/port*
3. switch(config-if)# **switchport**
4. switch(config-if)# **switchport mode dot1q-tunnel**
5. switch(config-if)# **l2protocol tunnel drop-threshold** [**cdp** | **stp** | **vtp**] *packets-per-sec*
6. (Optional) switch(config-if)# **no l2protocol tunnel drop-threshold** [**cdp** | **stp** | **vtp**]
7. switch(config-if)# **l2protocol tunnel shutdown-threshold** [**cdp** | **stp** | **vtp**] *packets-per-sec*
8. (Optional) switch(config-if)# **no l2protocol tunnel shutdown-threshold** [**cdp** | **stp** | **vtp**]
9. switch(config-if)# **exit**
10. (Optional) switch(config)# **no shutdown**
11. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport**	Sets the interface as a Layer 2 switching port.
Step 4	switch(config-if)# **switchport mode dot1q-tunnel**	Creates a 802.1Q tunnel on the port.
Step 5	switch(config-if)# **l2protocol tunnel drop-threshold** [**cdp** | **stp** | **vtp**] *packets-per-sec*	Specifies the maximum number of packets that can be processed on an interface before being dropped. Optionally, you can specify CDP, STP, or VTP. Valid values for the packets are from 1 to 4096.
Step 6	(Optional) switch(config-if)# **no l2protocol tunnel drop-threshold** [**cdp** | **stp** | **vtp**]	Resets the threshold values to 0 and disables the drop threshold.
Step 7	switch(config-if)# **l2protocol tunnel shutdown-threshold** [**cdp** | **stp** | **vtp**] *packets-per-sec*	Specifies the maximum number of packets that can be processed on an interface. When the number of packets is exceeded, the port is put in error-disabled state. Optionally,

	Command or Action	Purpose
		you can specify CDP, STP, or VTP. Valid values for the packets is from 1 to 4096.
Step 8	(Optional) switch(config-if)# **no l2protocol tunnel shutdown-threshold** [**cdp** | **stp** | **vtp**]	Resets the threshold values to 0 and disables the shutdown threshold.
Step 9	switch(config-if)# **exit**	Exits configuration mode.
Step 10	(Optional) switch(config)# **no shutdown**	Clears the errors on the interfaces and VLANs where policies correspond with hardware policies. This command allows policy programming to continue and the port to come up. If policies do not correspond, the errors are placed in an error-disabled policy state.
Step 11	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Configuring Combined Access Port Feature set

To configure combined access port feature set follow these steps.

SUMMARY STEPS

1. **interface** *interface* [**port** | **port-channel** | **vPC**]
2. **switchport mode private-vlan trunk** *secondary*
3. **switchport private-vlan trunk native vlan** *vlan_id*
4. **switchport private-vlan trunk allowed vlan** *vlan list*
5. **switchport private-vlan association trunk** *primary_vlan_ID secondary_vlan_ID*
6. **switchport vlan mapping** [*vlan-id-range* | *all*] *dot1q-tunnel outer vlan-id*
7. **storm-control broadcast level** [*high level*] [*lower level*]
8. **storm-control multicast level** [*high level*] [*lower level*]
9. **storm-control action** [**shutdown** | **trap**]
10. **load-interval counter** {*1* | *2* | *3* }
11. **switchport port-security maximum** [**max-addr**]
12. **switchport port-security** *action* [**restrict** | **shutdown** | **protect**]
13. **switchport port-security**
14. **service-policy** {**input** | **type** {**qos input** | **queuing** {**input** | **output**}}} *policy-map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface** *interface* [**port** | **port-channel** | **vPC**] **Example:** `switch# interface port-channel 202`	Places you into the interface configuration mode for the specified port channel. The range is from 1 to 4096.

	Command or Action	Purpose
Step 2	**switchport mode private-vlan trunk** *secondary* **Example:** `switch(config)# switchport mode private-vlan trunk secondary`	Configures the port as a secondary trunk port for a private VLAN.
Step 3	**switchport private-vlan trunk native vlan** *vlan_id* **Example:** `switch(config)# switchport private-vlan trunk native vlan 4002`	Configures native VLAN assigned on a PVLAN trunk port.
Step 4	**switchport private-vlan trunk allowed vlan** *vlan list* **Example:** `switch(config)# switchport private-vlan trunk allowed vlan 1002,4002`	Configures a list of allowed normal VLANs on a PVLAN trunk port.
Step 5	**switchport private-vlan association trunk** *primary_vlan_ID secondary_vlan_ID* **Example:** `switch(config)# switchport private-vlan association trunk 4050 4049`	Configures association between primary VLAN and secondary VLAN on the PVLAN trunk port.
Step 6	**switchport vlan mapping** [*vlan-id-range* | *all*] *dot1q-tunnel outer vlan-id* **Example:** `switch(config-if)# switchport vlan mapping all dot1q-tunnel 1002`	Enter the customer range VLANs or keyword all which includes all the 4K VLANs.
Step 7	**storm-control broadcast level** [*high level*] [*lower level*] **Example:** `switch(config-if)# storm-control broadcast level 1.00`	Configures broadcast storm control. Specifies the upper threshold levels for broadcast traffic.
Step 8	**storm-control multicast level** [*high level*] [*lower level*] **Example:** `switch(config-if)# storm-control multicast level 1.00`	Enables multicast traffic storm control on the interface, configures the traffic storm control level, and applies the traffic storm control level to all traffic storm control modes enabled on the interface.
Step 9	**storm-control action** [**shutdown** | **trap**] **Example:** `switch(config-if)# storm-control action shutdown`	Configures traffic storm-control to either generate trap or error-disable the port when a traffic storm occurs.
Step 10	**load-interval counter** {*1* | *2* | *3* } **Example:** `switch(config-if)# load-interval counter 1 5`	Specifies the interval between sampling statistics on the interface.

	Command or Action	Purpose
Step 11	**switchport port-security maximum** [**max-addr**] **Example:** `switch(config-if)# switchport port-security maximum 3`	Sets the maximum number of secure MAC addresses on a port.
Step 12	**switchport port-security** *action* [**restrict** | **shutdown** | **protect**] **Example:** `switch(config-if)# switchport port-security violation restrict`	Restrict security violation mode on the interface.
Step 13	**switchport port-security** **Example:** `switch(config-if)# switchport port-security`	Displays the port security configuration information.
Step 14	**service-policy** {**input** | **type** {**qos input** | **queuing** {**input** | **output**}}} *policy-map-name* **Example:** `switch(config-if)# service-policy type qos input ovh_qos`	Attaches a policy map to an interface.

Verifying the Q-in-Q Configuration

Command	Purpose
clear l2protocol tunnel counters [**interface** *if-range*]	Clears all the statistics counters. If no interfaces are specified, the Layer 2 protocol tunnel statistics are cleared for all interfaces.
show dot1q-tunnel [**interface** *if-range*]	Displays a range of interfaces or all interfaces that are in dot1q-tunnel mode.
show l2protocol tunnel [**interface** *if-range* | **vlan** *vlan-id*]	Displays Layer 2 protocol tunnel information for a range of interfaces, for all dot1q-tunnel interfaces that are part of a specified VLAN or all interfaces.
show l2protocol tunnel summary	Displays a summary of all ports that have Layer 2 protocol tunnel configurations.
show running-config l2pt	Displays the current Layer 2 protocol tunnel running configuration.

Configuration Examples for Q-in-Q and Layer 2 Protocol Tunneling

This example shows a service provider switch that is configured to process Q-in-Q for traffic coming in on Ethernet 7/1. A Layer 2 protocol tunnel is enabled for STP BPDUs. The customer is allocated VLAN 10 (outer VLAN tag).

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# vlan 10
switch(config-vlan)# no shutdown
switch(config-vlan)# no ip igmp snooping
switch(config-vlan)# exit
switch(config)# interface ethernet 7/1
switch(config-if)# switchport
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport access vlan 10
switch(config-if)# spanning-tree port type edge
switch(config-if)# l2protocol tunnel stp
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# exit
switch#
```

Configuring Port VLAN Mapping on VLANs

Before you begin

- Ensure that the physical or port channel on which you want to implement VLAN translation is configured as a Layer 2 trunk port.
- Ensure that the translated VLANs are created on the switch and are also added to the Layer 2 trunk ports trunk-allowed VLAN vlan-list.

Note As a best practice, do not add the ingress VLAN ID to the switchport allowed vlan-list under the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type/port*
3. **[no] switchport vlan mapping enable**
4. **[no] switchport vlan mapping** *vlan-id translated-vlan-id*
5. **[no] switchport vlan mapping all**
6. **copy running-config startup-config**
7. **show interface** [*if-identifier*] **vlan mapping**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**interface** *type/port* **Example:** `switch(config)# interface Ethernet1/1`	Specifies the interface that you are configuring.
Step 3	**[no] switchport vlan mapping enable** **Example:** `switch(config-if)# [no] switchport vlan mapping enable`	Enables VLAN translation on the switch port. VLAN translation is disabled by default. **Note** Use the **no** form of this command to disable VLAN translation.
Step 4	**[no] switchport vlan mapping** *vlan-id translated-vlan-id* **Example:** `switch(config-if)# switchport vlan mapping 10 100`	Translates a VLAN to another VLAN. • The range for both the *vlan-id* and *translated-vlan-id* arguments are from 1 to 4094. • You can configure VLAN translation between the ingress (incoming) VLAN and a local (translated) VLAN on a port. For the traffic arriving on the interface where VLAN translation is enabled, the incoming VLAN is mapped to a translated VLAN. Routing of traffic happens in context of SVI for translated VLAN. On the outgoing interface, where VLAN translation is configured, the traffic is converted to the original VLAN and egresses out. **Note** Use the **no** form of this command to clear the mappings between a pair of VLANs.
Step 5	**[no] switchport vlan mapping all** **Example:** `switch(config-if)# no switchport vlan mapping all`	Removes all VLAN mappings configured on the interface.
Step 6	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration. **Note** The VLAN translation configuration does not become effective until the switch port becomes an operational trunk port.
Step 7	**show interface** [*if-identifier*] **vlan mapping** **Example:** `switch# show interface ethernet1/1 vlan mapping`	Displays VLAN mapping information for a range of interfaces or for a specific interface.

Example

This example shows how to configure VLAN translation between (the ingress) VLAN 10 and (the local) VLAN 100. The show vlan counters command output shows the statistic counters as translated VLAN instead of customer VLAN.

```
switch#  configure terminal
switch(config)#  interface ethernet1/1
switch(config-if)#  switchport vlan mapping enable
switch(config-if)#  switchport vlan mapping 10 100
switch(config-if)#  switchport trunk allowed vlan 100
switch(config-if)#  show interface ethernet1/1 vlan mapping
Interface eth1/1:
Original VLAN           Translated VLAN
------------------      ---------------
10                           100

switch(config-if)#  show vlan counters
Vlan Id                              :100
Unicast Octets In                    :292442462
Unicast Packets In                   :1950525
Multicast Octets In                  :14619624
Multicast Packets In                 :91088
Broadcast Octets In                  :14619624
Broadcast Packets In                 :91088
Unicast Octets Out                   :304012656
Unicast Packets Out                  :2061976
L3 Unicast Octets In                 :0
L3 Unicast Packets In                :0
```