Configuring Q-in-VNI over VXLAN

Q-in-VNIs

In traditional network segmentation, the number of segments that can be created is limited to 4096 VLANs. This can be insufficient in situations, such as cloud computing environments where multiple tenants or customers may require their own virtual networks. VxLAN solves the problem of limited network segmentation and isolation in traditional VLAN-based networks.

Q-in-VNI over VxLAN addresses the requirement of limited network segmentation and isolation by stacking multiple VLANs above each other, allowing for even more virtual network segmentation and isolation. This provides a large number of virtual networks to be created, with the necessary flexibility and scalability for cloud computing environments and other situations where many virtual networks are required.

In summary, Q-in-VNI deployment using VxLAN EVPN as the transport network provides a highly efficient and scalable solution for delivering Layer 2 VPN services. It allows users to maintain their existing VLAN structure while connecting to the service provider's network and provides high availability and redundancy with VxLAN EVPN's robust control plane.

For more information on Q-in-VNI over VXLAN fabric deployment, see **Q-in-VNI over VxLAN Fabric Deployment Guide**.

Guidelines and Limitations for Q-in-VNI

Configuration Guidelines and Limitations

Follow these configuration guidelines and limitations when using Q-in-VNI.

- The **system dot1q-tunnel transit** [**vlan** *vlan-range*] command is required when running this feature on vPC VTEPs.
- Port VLAN mapping and Q-in-VNI cannot coexist on the same port.
- Port VLAN mapping and Q-in-VNI cannot coexist on a switch if the **system dot1q-tunnel transit** command is enabled. Beginning with Cisco NX-OS Release 9.3(5), port VLAN mapping and Q-in-VNI can coexist on the same switch but on different ports and different provider VLANs, which are configured using the **system dot1q-tunnel transit vlan** *vlan-range* command.
- For proper operation during L3 uplink failure scenarios on vPC VTEPs, configure a backup SVI and enter the **system nve infra-vlans** *backup-svi-vlan* command. On Cisco Nexus 9000-EX platform switches, the backup SVI VLAN needs to be the native VLAN on the peer-link.
- When configuring access ports and trunk ports for Cisco Nexus 9000 Series switches with a Leaf Spine Engine (LSE), you can have access ports, trunk ports, and dot1q ports on different interfaces on the same switch.
- You cannot have the same VLAN configured for both dot1q and trunk ports/access ports.
- Disable ARP suppression on the provider VNI for ARP traffic originated from a customer VLAN in order to flow.

  ```
  switch(config)# interface nve 1
  switch(config-if-nve)# member VNI 10000011
  switch(config-if-nve-vni)# no suppress-arp
  ```

- Q-in-VNI cannot coexist with a VTEP that has Layer 3 subinterfaces configured. Beginning with Cisco NX-OS Release 9.3(5), this limitation no longer applies to Cisco Nexus 9332C, 9364C, 9300-FX/FX2, and 9300-GX platform switches.
- When VLAN1 is configured as the native VLAN with selective Q-in-VNI with the multiple provider tag, traffic on the native VLAN gets dropped. Do not configure VLAN1 as the native VLAN when the port is configured with selective Q-in-VNI. When VLAN1 is configured as a customer VLAN, the traffic on VLAN1 gets dropped.
- The base port mode must be a dot1q tunnel port with an access VLAN configured.
- VNI mapping is required for the access VLAN on the port.
- If you have Q-in-VNI on one Cisco Nexus 9300-EX Series switch VTEP and trunk on another Cisco Nexus 9300-EX Series switch VTEP, the bidirectional traffic will not be sent between the two ports.
- Cisco Nexus 9300-EX Series of switches performing VXLAN and Q-in-Q, a mix of provider interface and VXLAN uplinks is not considered. The VXLAN uplinks have to be separated from the Q-in-Q provider or customer interface.

  For vPC use cases, the following considerations must be made when VXLAN and Q-in-Q are used on the same switch.

  - The vPC peer-link has to be specifically configured as a provider interface to ensure orphan-to-orphan port communication. In these cases, the traffic is sent with two IEEE 802.1q tags (double dot1q tagging). The inner dot1q is the customer VLAN ID while the outer dot1q is the provider VLAN ID (access VLAN).
  - The vPC peer-link is used as backup path for the VXLAN encapsulated traffic in the case of an uplink failure. In Q-in-Q, the vPC peer-link also acts as the provider interface (orphan-to-orphan port communication). In this combination, use the native VLAN as the backup VLAN for traffic to

handle uplink failure scenarios. Also make sure the backup VLAN is configured as a system infra VLAN (system nve infra-vlans).

Supported Platforms and Features

This section lists the supported platforms and features for Q-in-VNI.

- Cisco Nexus 9300 platform switches support single tag. You can enable it by entering the **no overlay-encapsulation vxlan-with-tag** command for the NVE interface:

  ```
  switch(config)# interface nve 1
  switch(config-if-nve)# no overlay-encapsulation vxlan-with-tag
  switch# show run int nve 1

  !Command: show running-config interface nve1
  !Time: Wed Jul 20 23:26:25 2016

  version 7.0(3u)I4(2u)

  interface nve1
    no shutdown
    source-interface loopback0
    host-reachability protocol bgp
    member vni 900001 associate-vrf
    member vni 2000980
      mcast-group 225.4.0.1
  ```

- Beginning with Cisco NX-OS Release 10.1(1), Selective Q-in-VNI and VXLAN VLAN on Same Port feature is supported on Cisco Nexus 9300-FX3 platform switches.
- Q-in-VNI only supports VXLAN bridging. It does not support VXLAN routing.
- Q-in-VNI and selective Q-in-VNI are supported with VXLAN Flood and Learn with Ingress Replication and VXLAN EVPN with Ingress Replication.
- Beginning with Cisco NX-OS Release 10.2(3)F, the Cisco Nexus 9300-FX3/GX2 platform switches support Q-in-VNI to coexist with a VTEP that has Layer 3 subinterfaces configured.
- Beginning with Cisco NX-OS Release 9.3(5), Q-in-VNI is supported on Cisco Nexus 9300-GX platform switches.
- Beginning with Cisco NX-OS Release 10.2(3)F, Q-in-VNI is supported on the Cisco Nexus 9300-GX2 platform switches.
- Beginning with Cisco NX-OS Release 9.3(5), Q-in-VNI supports vPC Fabric Peering.
- Beginning with Cisco NX-OS Release 10.3(3)F, IPv6 underlay is supported on Q-in-VNI, Selective Q-in-VNI and Q-in-Q-Q-in-VNI for VXLAN EVPN on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2 switches.

Unsupported Platforms and Features

This section lists the unsupported platforms and features for Q-in-VNI.

- Cisco Nexus 9300-EX platform switches do not support double tag. They support only single tag.

- Cisco Nexus 9300-EX platform switches do not support traffic between ports configured for Q-in-VNI and ports configured for trunk.
- The dot1q tunnel mode does not support ALE ports on Cisco Nexus 9300 Series and Cisco Nexus 9500 platform switches.
- Q-in-VNI does not support FEX.
- Q-in-VNI, selective Q-in-VNI, and QinQ-QinVNI are not supported with the multicast underlay on Cisco Nexus 9000-EX platform switches.
- Q-in-VNI is not supported as part of multi-site solution.
- Q-in-VNI and Selective Q-in-VNI are not supported on Cisco Nexus 9500 Series switches with 9700-EX/FX/GX line cards.

Configure the Q-in-VNI

Using Q-in-VNI provides a way for you to segregate traffic by mapping to a specific port. In a multi-tenant environment, you can specify a port to a tenant and send/receive packets over the VXLAN overlay.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# config terminal
```

Step 2 Use the **interface** *type port* command to enter interface configuration mode.

Example:

```
switch(config)# interface ethernet 1/4
```

Step 3 Use the **switchport mode dot1q-tunnel** command to create a 802.1Q tunnel on the port.

Example:

```
switch(config-if)# switchport mode dot1q-tunnel
```

Step 4 Use the **switchport access vlan***vlan-id* command to specify the port assigned to a VLAN.

Example:

```
switch(config-if)# switchport access vlan 10
```

Step 5 Use the **spanning-tree bpdufilter enable** command to enable BPDU Filtering for the specified spanning tree edge interface.

Example:

```
switch(config-if)# spanning-tree bpdufilter enable
```

The following is an example of configuring Q-in-VNI:

```
switch# config terminal
switch(config)# interface ethernet 1/4
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport access vlan 10
switch(config-if)# spanning-tree bpdufilter enable
switch(config-if)#
```

Selective Q-in-VNIs

Selective Q-in-VNI is a VXLAN tunneling feature that allows a user specific range of customer VLANs on a port to be associated with one specific provider VLAN. Packets that come in with a VLAN tag that matches any of the configured customer VLANs on the port are tunneled across the VXLAN fabric using the properties of the service provider VNI. The VXLAN encapsulated packet carries the customer VLAN tag as part of the L2 header of the inner packet.

The packets that come in with a VLAN tag that is not present in the range of the configured customer VLANs on a selective Q-in-VNI configured port are dropped. This includes the packets that come in with a VLAN tag that matches the native VLAN on the port. Packets coming untagged or with a native VLAN tag are L3 routed using the native VLAN's SVI that is configured on the selective Q-in-VNI port (no VXLAN).

- Selective Q-in-VNI is supported on both vPC and non-vPC ports on Cisco Nexus 9300-EX and 9300-FX/FXP/FX2/FX3 and 9300-GX platform switches. This feature is not supported on Cisco Nexus 9200 and 9300 platform switches.
- Beginning with Cisco NX-OS Release 9.3(5), selective Q-in-VNI supports vPC Fabric Peering.
- Configuring selective Q-in-VNI on one VTEP and configuring plain Q-in-VNI on the VXLAN peer is supported. Configuring one port with selective Q-in-VNI and the other port with plain Q-in-VNI on the same switch is supported.
- Selective Q-in-VNI is an ingress VLAN tag-policing feature. Only ingress VLAN tag policing is performed with respect to the selective Q-in-VNI configured range.

  For example, selective Q-in-VNI customer VLAN range of 100-200 is configured on VTEP1 and customer VLAN range of 200-300 is configured on VTEP2. When traffic with VLAN tag of 175 is sent from VTEP1 to VTEP2, the traffic is accepted on VTEP1, since the VLAN is in the configured range and it is forwarded to the VTEP2. On VTEP2, even though VLAN tag 175 is not part of the configured range, the packet egresses out of the selective Q-in-VNI port. If a packet is sent with VLAN tag 300 from VTEP1, it is dropped because 300 is not in VTEP1's selective Q-in-VNI configured range.
- Beginning with Cisco NX-OS Release 10.1(1), Selective Q-in-VNI and Advertise PIP on a VTEP feature is supported on Cisco Nexus 9300-FX3 platform switches.
- Beginning with Cisco NX-OS Release 9.3(5), the **advertise-pip** command is supported with selective Q-in-VNI on a VTEP.
- Port VLAN mapping and selective Q-in-VNI cannot coexist on the same port.
- Port VLAN mapping and selective Q-in-VNI cannot coexist on a switch if the **system dot1q-tunnel transit** command is enabled. Beginning with Cisco NX-OS Release 9.3(5), port VLAN mapping and

Q-in-VNI can coexist on the same switch but on different ports and different provider VLANs, which are configured using the **system dot1q-tunnel transit vlan** *vlan-range* command.

- Configure the **system dot1q-tunnel transit** [**vlan** *vlan-id*] command on vPC switches with selective Q-in-VNI configurations. This command is required to retain the inner Q-tag as the packet goes over the vPC peer link when one of the vPC peers has an orphan port. With this CLI configuration, the **vlan dot1Q tag native** functionality does not work. Prior to Cisco NX-OS Release 9.3(5), every VLAN created on the switch is a provider VLAN and cannot be used for any other purpose.

  Beginning with Cisco NX-OS Release 9.3(5), selective Q-in-VNI and VXLAN VLANs can be supported on the same port. With the [**vlan** *vlan-range*] option, you can specify the provider VLANs and allow other VLANs to be used for regular VXLAN traffic. In the following example, the VXLAN VLAN is 50, the provider VLAN is 501, the customer VLANs are 31-40, and the native VLAN is 2400.

  ```
  system dot1q-tunnel transit vlan 501
  interface Ethernet1/1/2
    switchport
    switchport mode trunk
    switchport trunk native vlan 2400
    switchport vlan mapping 31-40 dot1q-tunnel 501
    switchport trunk allowed vlan 50,501,2400
    spanning-tree port type edge trunk
    mtu 9216
    no shutdown
  ```

- The native VLAN configured on the selective Q-in-VNI port cannot be a part of the customer VLAN range. If the native VLAN is part of the customer VLAN range, the configuration is rejected.

  The provider VLAN can overlap with the customer VLAN range. For example, **switchport vlan mapping 100-1000 dot1q-tunnel 200**.

- By default, the native VLAN on any port is VLAN 1. If VLAN 1 is configured as part of the customer VLAN range using the **switchport vlan mapping** <range>**dot1q-tunnel** <sp-vlan> CLI command, the traffic with customer VLAN 1 is not carried over as VLAN 1 is the native VLAN on the port. If customer wants VLAN 1 traffic to be carried over the VXLAN cloud, they should configure a dummy native VLAN on the port whose value is outside the customer VLAN range.

- To remove some VLANs or a range of VLANs from the configured switchport VLAN mapping range on the selective Q-in-VNI port, use the **no** form of the **switchport vlan mapping** <*range*>**dot1q-tunnel** <*sp-vlan*> command.

  For example, VLAN 100-1000 is configured on the port. To remove VLAN 200-300 from the configured range, use the **no switchport vlan mapping** <*200-300*> **dot1q-tunnel** <*sp-vlan*> command.

  ```
  interface Ethernet1/32
    switchport
    switchport mode trunk
    switchport trunk native vlan 4049
    switchport vlan mapping 100-1000 dot1q-tunnel 21
    switchport trunk allowed vlan 21,4049
    spanning-tree bpdufilter enable
    no shutdown

  switch(config-if)# no sw vlan mapp 200-300 dot1q-tunnel 21
  switch(config-if)# sh run int e 1/32

  version 7.0(3)I5(2)

  interface Ethernet1/32
    switchport
    switchport mode trunk
  ```

```
  switchport trunk native vlan 4049
  switchport vlan mapping 100-199,301-1000 dot1q-tunnel 21
  switchport trunk allowed vlan 21,4049
  spanning-tree bpdufilter enable
  no shutdown
```

- See the following example for the provider VLAN configuration:

```
vlan 50
 vn-segment 10050
```

- See the following example for configuring VXLAN Flood and Learn with Ingress Replication:

```
member vni 10050
    ingress-replication protocol static
      peer-ip 100.1.1.3
      peer-ip 100.1.1.5
      peer-ip 100.1.1.10
```

- See the following example for the interface nve configuration:

```
interface nve1
 no shutdown
 source-interface loopback0  member vni 10050
mcast-group 230.1.1.1
```

- See the following example for configuring an SVI in the native VLAN to routed traffic.

```
vlan 150
interface vlan150
 no shutdown
 ip address 150.1.150.6/24
 ip pim sparse-mode
```

- See the following example for configuring selective Q-in-VNI on a port. In this example, native VLAN 150 is used for routing the untagged packets. Customer VLANs 200-700 are carried across the dot1q tunnel. The native VLAN 150 and the provider VLAN 50 are the only VLANs allowed.

```
switch# config terminal
switch(config)#interface Ethernet 1/31
switch(config-if)#switchport
switch(config-if)#switchport mode trunk
switch(config-if)#switchport trunk native vlan 150
switch(config-if)#switchport vlan mapping 200-700 dot1q-tunnel 50
switch(config-if)#switchport trunk allowed vlan 50,150
switch(config-if)#no shutdown
```

- Disable ARP suppression on the provider VNI for ARP traffic originated from a customer VLAN in order to flow.

```
switch(config)# interface nve 1
switch(config-if-nve)# member VNI 10000011
 switch(config-if-nve-vni)# no suppress-arp
```

Q-in-VNIs

Q-in-VNIs

Q-in-VNI with Layer 2 Protocol Tunneling (L2PT) is used to transport control and data packets across a VXLAN EVPN fabric for multi-tagged traffic.

To enable Q-in-VNI with L2PT at the VLAN level, use the **l2protocol tunnel vxlan vlan** <*vlan-range*> command which marks the VLANs for tunneling all packets including L2 protocol packets. The **switchport trunk allow-multi-tag** command is also required for the VXLAN fabric to tunnel packets with multiple tags.

For more information on Q-in-VNI with L2PT configuration, refer to Configure the Q-in-VNI with L2PT, on page 8.

Guidelines and Limitations for Q-in-VNI with L2PT

Follow these guidelines and limitations when using Q-in-VNI with L2PT.

- Beginning with Cisco NX-OS Release 10.3(2)F, Q-in-VNI with L2PT is supported on Cisco Nexus 9300-FX/FX2/FX3/GX/GX2 ToR switches.
- L2PT interface must be trunk on the Cisco Nexus 9300-FX/FX2/FX3/GX/GX2 ToR switches.
- Once the **l2protocol tunnel vxlan** command is run on an interface, all VLANs in the command become tunneling VLANs and cannot be used on any other port for any other purpose.
- Only two interfaces in the network can be member of the tunnel VLAN. For vPC cases, both vPC ports on the vPC switches and MCT will also be part of the tunnel VLAN.
- Same VLAN must not be tunneled on multiple interfaces.
- The **l2protocol tunnel vxlan** command is allowed only on trunk ports. It also requires "multi-tag" configuration to preserve the multiple tags across the vxlan fabric.
- Cross Connect feature and **l2protocol tunnel vxlan** command can not be used together on a switch.
- Existing L2PT command options like "STP" can not be used along with the **l2protocol tunnel vxlan** command.
- In vPC scenarios, disable spanning tree for L2PT-mapped VLANs by using the **no spanning-tree vlan** *vlan id* command. In non-vPC scenarios, either enable BPDU filtering on the L2PT ports or disable spanning tree for the L2PT-mapped VLANs.

Q-in-VNI with L2PT support and Ethertype support are introduced on specific Cisco Nexus switches in various releases.

- Beginning with Cisco NX-OS Release 10.3(3)F, Ethertype support for Q-in-VNI with L2PT is provided on Cisco Nexus 9300-FX2/FX3/GX/GX2 ToR switches.

Configure the Q-in-VNI with L2PT

Follow this procedure to configure the Q-in-VNI with L2PT on VXLAN VLAN:

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
```

Step 2 Use the **interface ethernet** *slot/port* command to specify the interface that you are configuring.

Example:

```
switch(config)# interface ethernet1/1
```

Step 3 Use the **switchport** command to configure it as a Layer 2 port.

Example:

```
switch(config-inf)# switchport
```

Step 4 Use the **switchport mode trunk** command to set the interface as a Layer 2 trunk port.

Example:

```
switch(config-inf)# switchport mode trunk
```

Step 5 Use the **switchport dot1q ethertype** *ethertype-value* command to set the Ethertype for the port.

Example:

```
switch(config-inf)# switchport dot1q ethertype 0x88a8
```

Step 6 Use the **switchport trunk allow-multi-tag** command to set the allowed VLANs as the provider VLANs excluding the native VLAN.

Example:

```
switch(config-inf)# switchport trunk allow-multi-tag
```

Step 7 Use the **switchport trunk allowed vlan** *vlan-list* command to set the allowed VLANs for the trunk interface.

Example:

```
switch(config-inf)# switchport trunk allowed vlan 1201-1202
```

Step 8 Use the **l2protocol tunnel vxlan vlan** <*vlan-range*> command to set all VLANs in the command as tunneling VLANs.

Example:

```
switch(config-inf)# l2protocol tunnel vxlan vlan 1201-1202
```

Verify the Q-in-VNI with L2PT Configuration

Procedure

Step 1 Use the **show run interface ethernet***slot/port* command to display L2PT VXLAN VLAN interface information.

Example:

```
switch(config-if)# sh run int e1/1
 interface Ethernet1/1
  switchport
  switchport mode trunk
  switchport trunk allow-multi-tag
  switchport trunk allowed vlan 1201-1202
  l2protocol tunnel vxlan vlan 1201-1202
  no shutdown
```

Step 2 Use the **show run l2pt** command to display L2PT VXLAN VLAN configuration information.

Example:

```
switch# sh run l2pt
interface Ethernet1/1
  switchport mode trunk
  l2protocol tunnel vxlan vlan 1201-1202
  no shutdown
```

Step 3 Use the **show l2protocol tunnel interface ethernet***slot/port* command to display L2PT interface information.

Example:

```
switch# show l2protocol tunnel interface e1/1
COS for Encapsulated Packets: 5
Interface: Eth1/1 Vxlan Vlan 1201-1202
```

Step 4 Use the **show vpc consistency-parameters interface***slot/port* command to display the status of the parameters that must be consistent across all vPC interfaces including L2PT VXLAN VLAN.

Example:

```
switch# sh run int po101

interface port-channel101
  switchport
  switchport mode trunk
  switchport trunk native vlan 80
  switchport trunk allow-multi-tag
  switchport trunk allowed vlan 80,1201-1203,1301
  spanning-tree port type edge trunk
  vpc 101
  l2protocol tunnel vxlan vlan 1201-1203,1301

switch# sh vpc consistency-parameters interface po101

    Legend:
        Type 1 : vPC will be suspended in case of mismatch

Name                        Type  Local Value            Peer Value
-------------               ----  ---------------------- -----------------------
delayed-lacp                1     disabled               disabled
lacp suspend disable        1     enabled                enabled
mode                        1     active                 active
Switchport Isolated         1     0                      0
Interface type              1     port-channel           port-channel
LACP Mode                   1     on                     on
Virtual-ethernet-bridge     1     Disabled               Disabled
Speed                       1     25 Gb/s                25 Gb/s
Duplex                      1     full                   full
MTU                         1     1500                   1500
Port Mode                   1     trunk                  trunk
Native Vlan                 1     80                     80
Admin port mode             1     trunk                  trunk
```

```
Port-type External          1     Disabled              Disabled
STP Port Guard              1     Default               Default
STP Port Type               1     Edge Trunk Port       Edge Trunk Port
STP MST Simulate PVST       1     Default               Default
lag-id                      1     [(7f9b,               [(7f9b,
                                  0-23-4-ee-be-4, 8065, 0-23-4-ee-be-4, 8065,
                                  0, 0), (8000,         0, 0), (8000,
                                  a8-9d-21-f8-4b-31, 64, a8-9d-21-f8-4b-31, 64,
                                   0, 0)]                0, 0)]
Allow-Multi-Tag             1     Enabled               Enabled
Vlan xlt mapping            1     Disabled              Disabled
L2PT Vxlan Vlans            2     1201-1203,1301        1201-1203,1301
vPC card type               1     N9K TOR               N9K TOR
Allowed VLANs               -     80,1201-1203,1301     80,1201-1203,1301
Local suspended VLANs       -     -                     -
```

Configure Q-in-VNI with LACP Tunneling

Q-in-VNI can be configured to tunnel LACP packets.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Use the **interface** *type port* command to enter interface configuration mode.

Example:

```
switch(config)# interface ethernet 1/4
switch(config-if)#
```

Step 3 Use the **switchport mode dot1q-tunnel** command to enable dot1q-tunnel mode.

Example:

```
switch(config-if)# switchport mode dot1q-tunnel
```

Step 4 Use the **switchport access vlan***vlan-id* command to specify the port assigned to a VLAN.

Example:

```
switch(config-if)# switchport access vlan 10
```

Step 5 Use the **interface nve***x* command to create a VXLAN overlay interface that terminates VXLAN tunnels.

Example:

```
switch(config-if)# interface nve1
switch(config-if)#
```

Step 6 Use the **overlay-encapsulation vxlan-with-tag tunnel-control-frames** command to enable Q-in-VNI for LACP tunneling.

Example:

```
switch(config-if)# overlay-encapsulation vxlan-with-tag tunnel-control-frames
```

Step 7 Use the **overlay-encapsulation vxlan-with-tag tunnel-control-frames lacp** command to enable Q-in-VNI for LACP tunneling.

Example:

```
switch(config-if)# overlay-encapsulation vxlan-with-tag tunnel-control-frames lacp
```

- The following is an example of configuring a Q-in-VNI for LACP tunneling:

```
switch# config terminal
switch(config)# interface ethernet 1/4
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport access vlan 10
switch(config-if)# spanning-tree bpdufilter enable
switch(config-if)# interface nve1
switch(config-if)# overlay-encapsulation vxlan-with-tag tunnel-control-frames
```

Note

- STP is disabled on VNI mapped VLANs.
- No spanning-tree VLAN <> on the VTEP.
- No MAC address-table notification for mac-move.

- The following is an example of configuring a Q-in-VNI for LACP tunneling (NX-OS 7.0(3)I2(2) and earlier releases):

```
switch# config terminal
switch(config)# interface ethernet 1/4
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport access vlan 10
switch(config-if)# spanning-tree bpdufilter enable
switch(config-if)# interface nve1
switch(config-if)# overlay-encapsulation vxlan-with-tag tunnel-control-frames
```

Note

- STP is disabled on VNI mapped VLANs.
- No spanning-tree VLAN <> on the VTEP.
- No MAC address-table notification for mac-move.
- As a best practice, configure a fast LACP rate on the interface where the LACP port is configured. Otherwise the convergence time is approximately 90 seconds.

- The following is an example of configuring a Q-in-VNI for LACP tunneling (NX-OS 7.0(3)I3(1) and later releases):

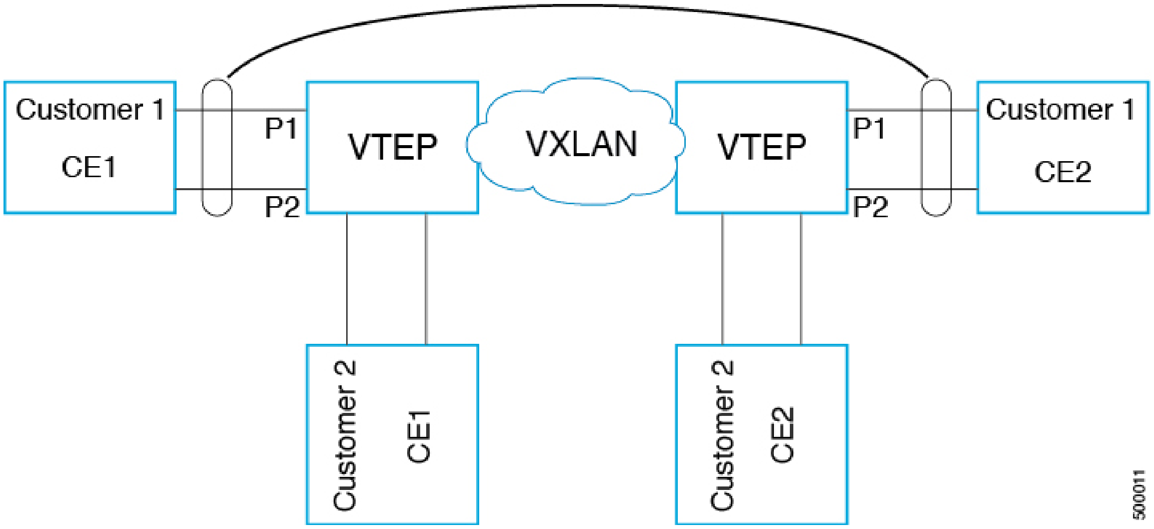
```
switch# config terminal
switch(config)# interface ethernet 1/4
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport access vlan 10
switch(config-if)# spanning-tree bpdufilter enable
switch(config-if)# interface nve1
switch(config-if)# overlay-encapsulation vxlan-with-tag tunnel-control-frames lacp
```

Note

- STP is disabled on VNI mapped VLANs.
- No spanning-tree VLAN <> on the VTEP.
- No MAC address-table notification for mac-move.
- As a best practice, configure a fast LACP rate on the interface where the LACP port is configured. Otherwise the convergence time is approximately 90 seconds.

- The following is an example topology that pins each port of a port-channel pair to a unique VM. The port-channel is stretched from the CE perspective. There is no port-channel on VTEP. The traffic on P1 of CE1 transits to P1 of CE2 using Q-in-VNI.

Figure 1: LACP Tunneling Over VXLAN P2P Tunnels



Note

- Q-in-VNI can be configured to tunnel LACP packets. (Able to provide port-channel connectivity across data-centers.)
    - Gives impression of L1 connectivity and co-location across data-centers.
    - Exactly two sites. Traffic coming from P1 of CE1 goes out of P1 of CE2. If P1 of CE1 goes down, LACP provides coverage (over time) to redirect traffic to P2.
- Uses static ingress replication with VXLAN with flood and learn. Each port of the port channel is configured with Q-in-VNI. There are multiple VNIs for each member of a port-channel and each port is pinned to specific VNI.
    - To avoid saturating the MAC, you should turn off/disable learning of VLANS.
- Configuring Q-in-VNI to tunnel LACP packets is not supported for VXLAN EVPN.
- The number of port-channel members supported is the number of ports supported by the VTEP.

Selective Q-in-VNI with Multiple Provider VLANs

Selective Q-in-VNIs

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

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

Follow these guidelines and limitations when using Selective Q-in-VNI with multiple provider VLANs.

- All the existing guidelines and limitations for Selective Q-in-VNI apply.
- This feature is supported with VXLAN BGP EVPN IR mode only.
- When enabling multiple provider VLANs on a vPC port channel, make sure that the configuration is consistent across the vPC peers.
- Port VLAN mapping and selective Q-in-VNI cannot coexist on the same port.
- Port VLAN mapping and selective Q-in-VNI cannot coexist on a switch if the **system dot1q-tunnel transit** command is enabled. Beginning with Cisco NX-OS Release 9.3(5), port VLAN mapping and selective Q-in-VNI can coexist on the same switch but on different ports and different provider VLANs, which are configured using the **system dot1q-tunnel transit vlan** *vlan-range* command.

- The **system dot1q-tunnel transit** [**vlan** *vlan-range*] command is required when using this feature on vPC VTEPs.
- For proper operation during Layer 3 uplink failure scenarios on vPC VTEPs, configure the backup SVI and enter the **system nve infra-vlans** *backup-svi-vlan* command. On Cisco Nexus 9000-EX platform switches, the backup SVI VLAN must be the native VLAN on the peer-link.
- As a best practice, do not allow provider VLANs on a regular trunk.

Additional guidelines and limitations for Selective Q-in-VNI with multiple provider VLANs.

- We recommend not creating or allowing customer VLANs on the switch where customer-VLAN to provider-VLAN mapping is configured.
- We do not support specific native VLAN configuration when the **switchport vlan mapping all dot1q-tunnel** command is entered.
- Beginning with Cisco NX-OS Release 9.3(5), selective Q-in-VNI with a multiple provider tag supports vPC Fabric Peering.
- Disable ARP suppression on the provider VNI for ARP traffic originated from a customer VLAN in order to flow.

  ```
  switch(config)# interface nve 1
  switch(config-if-nve)# member VNI 10000011
  switch(config-if-nve-vni)# no suppress-arp
  ```

- All incoming traffic should be tagged when the interface is configured with the **switchport vlan mapping all dot1q-tunnel** command.

Configure Selective Q-in-VNI with Multiple Provider VLANs

Before you begin

You must configure provider VLANs and associate the VLAN to a vn-segment.

You can configure selective Q-in-VNI with multiple provider VLANs.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
```

Step 2 Use the **vlan** and **vn-segment** commands to configure Layer 2 VLANs and associate them to a vn-segment.

Example:

```
switch(config)# vlan 10
vn-segment 10000010
switch(config)# vlan 20
 vn-segment 10000020
```

Step 3 Use the **interface port-channel** and **switchport vlan mapping** commands to configure interface settings for dot1Q VLAN tag traffic.

Example:

```
switch(config)# interf port-channel 10
switch(config-if)# switchport
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk native vlan 3962
switch(config-if)# switchport vlan mapping 2-400 dot1q-tunnel 10
switch(config-if)# switchport vlan mapping 401-800 dot1q-tunnel 20
switch(config-if)# switchport vlan mapping 801-1200 dot1q-tunnel 30
switch(config-if)# switchport vlan mapping 1201-1600 dot1q-tunnel 40
switch(config-if)# switchport vlan mapping 1601-2000 dot1q-tunnel 50
switch(config-if)# switchport vlan mapping 2001-2400 dot1q-tunnel 60
switch(config-if)# switchport vlan mapping 2401-2800 dot1q-tunnel 70
switch(config-if)# switchport vlan mapping 2801-3200 dot1q-tunnel 80
switch(config-if)# switchport vlan mapping 3201-3600 dot1q-tunnel 90
switch(config-if)# switchport vlan mapping 3601-3960 dot1q-tunnel 100
switch(config-if)# switchport trunk allowed vlan 10,20,30,40,50,60,70,80,90,100,3961-3967
```

This example shows how to configure Selective Qinvni with multiple provider VLANs:

```
switch# show run vlan 121
vlan 121
vlan 121
  vn-segment 10000021

switch#
switch# sh run interf port-channel 5

interface port-channel5
  description VPC PO
  switchport
  switchport mode trunk
  switchport trunk native vlan 504
  switchport vlan mapping 11 dot1q-tunnel 111
  switchport vlan mapping 12 dot1q-tunnel 112
  switchport vlan mapping 13 dot1q-tunnel 113
  switchport vlan mapping 14 dot1q-tunnel 114
  switchport vlan mapping 15 dot1q-tunnel 115
  switchport vlan mapping 16 dot1q-tunnel 116
  switchport vlan mapping 17 dot1q-tunnel 117
  switchport vlan mapping 18 dot1q-tunnel 118
  switchport vlan mapping 19 dot1q-tunnel 119
  switchport vlan mapping 20 dot1q-tunnel 120
  switchport trunk allowed vlan 111-120,500-505
  vpc 5

switch#

switch# sh spanning-tree vlan 111

VLAN0111
  Spanning tree enabled protocol rstp
  Root ID    Priority    32879
             Address     7079.b3cf.956d
             This bridge is the root
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32879  (priority 32768 sys-id-ext 111)
             Address     7079.b3cf.956d
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role Sts Cost      Prio.Nbr Type
```

```
---------------- ---- --- --------- -------- --------------------------------
Po1              Desg FWD 1         128.4096 (vPC peer-link) Network P2p
Po5              Desg FWD 1         128.4100 (vPC) P2p
Eth1/7/2         Desg FWD 10        128.26   P2p

switch#


switch# sh vlan internal info mapping | b Po5
  ifindex Po5(0x16000004)
  vlan mapping enabled: TRUE
  vlan translation mapping information (count=10):
    Original Vlan       Translated Vlan
    -------------       ---------------
    11                   111
    12                   112
    13                   113
    14                   114
    15                   115
    16                   116
    17                   117
    18                   118
    19                   119
    20                   120
switch#


switch# sh consistency-checker vxlan selective-qinvni interface port-channel 5
Performing port specific checks for intf port-channel5
Port specific selective QinVNI checks for interface port-channel5 : PASS
Performing port specific checks for intf port-channel5
Port specific selective QinVNI checks for interface port-channel5 : PASS

switch#
```

Configuring QinQ-QinVNI

QinQ-QinVNIs

QinQ-QinVNI is a VXLAN tunneling feature that allows you to configure a trunk port as a multi-tag port to preserve the customer VLANs that are carried across the network.

- On a port that is configured as multi-tag, packets are expected with multiple-tags or at least one tag. When multi-tag packets ingress on this port, the outer-most or first tag is treated as provider-tag or provider-vlan. The remaining tags are treated as customer-tag or customer-vlan.
- This feature is supported on both vPC and non-vPC ports.
- Ensure that the **switchport trunk allow-multi-tag** command is configured on both of the vPC-peers. It is a type 1 consistency check.
- This feature is supported with VXLAN Flood and Learn and VXLAN EVPN.

Guidelines and Limitations for QinQ-QinVNI

Follow these guidelines and limitations when configuring QinQ-QinVNI.

Supported platform and releases

- This feature is supported on the Cisco Nexus 9300-FX/FX2/FX3, and 9300-GX platform switches.
- Beginning with Cisco NX-OS Release 10.2(3)F, QinQ-QinVNI is supported on the Cisco Nexus 9300-GX2 platform switches.

Supported and unsupported features

- This feature supports vPC Fabric Peering.
- This feature supports VXLAN bridging but does not support VXLAN routing.

Configuration limitations

- On a multi-tag port, provider VLANs must be a part of the port. They are used to derive the VNI for that packet.
- Untagged packets are associated with the native VLAN. If the native VLAN is not configured, the packet is associated with the default VLAN (VLAN 1).
- Packets coming in with an outermost VLAN tag (provider-vlan), not present in the range of allowed VLANs on a multi-tag port, are dropped.
- Packets coming in with an outermost VLAN tag (provider-vlan) tag matching the native VLAN are routed or bridged in the native VLAN's domain.
- Multicast data traffic with more than two Q-Tags is not supported when snooping is enabled on the VXLAN VLAN.
- You need at least one multi-tag trunk port allowing the provider VLANs in Up state on both vPC peers. Otherwise, traffic traversing via the peer-link for these provider VLANs will not carry all inner C-Tags.
- The **system dot1q-tunnel transit** [**vlan** *vlan-range*] command is required when running this feature on vPC VTEPs.
- QinQ-QinVNI does not interoperate with Selective Q-in-VNI. Only one can be configured at any given time.

Configure the QinQ-QinVNI

Note You can also carry native VLAN (untagged traffic) on the same multi-tag trunk port.

The native VLAN on a multi-tag port cannot be configured as a provider VLAN on another multi-tag port or a dot1q enabled port on the same switch.

The **allow-multi-tag** command is allowed only on a trunk port. It is not available on access or dot1q ports.

The **allow-multi-tag** command is not allowed on Peer Link ports. Port channel with multi-tag enabled must not be configured as a vPC peer-link.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
```

Step 2 Use the **interface ethernet***slot/port* command to specify the interface to configure.

Example:

```
switch(config)# interface ethernet1/7
```

Step 3 Use the **switchport** command to configure the interface as a Layer 2 port.

Example:

```
switch(config-inf)# switchport
```

Step 4 Use the **switchport mode trunk** command to set the interface as a Layer 2 trunk port.

Example:

```
switch(config-inf)# switchport mode trunk
```

Step 5 Use the **switchport trunk native vlan***vlan-id* command to set the native VLAN for the 802.1Q trunk.

Example:

```
switch(config-inf)# switchport trunk native vlan 30
```

Step 6 Use the **switchport trunk allowed vlan***vlan-list* command to set the allowed VLANs for the trunk interface.

Example:

```
switch(config-inf)# switchport trunk allowed vlan 10,20,30
```

Step 7 Use the **switchport trunk allow-multi-tag** command to set the allowed VLANs as provider VLANs excluding the native VLAN.

Example:

```
switch(config-inf)# switchport trunk allow-multi-tag
```

```
interface Ethernet1/7
switchport
switchport mode trunk
switchport trunk native vlan 30
switchport trunk allow-multi-tag
switchport trunk allowed vlan 10,20,30
no shutdown
```