

Configure and Troubleshoot Selective Q-in-Q on Nexus 9000

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Network Diagram](#)

[Configure](#)

[Troubleshoot](#)

[Consistency Checker](#)

[Useful commands to gather and check for any Q-in-Q issue](#)

[Related Information](#)

Introduction

This document describes how to configure and troubleshoot Selective Q-in-Q in Nexus 9300 (EX-FX-GX) model.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Q-in-Q encapsulation
- Dot1q switching
- Nexus 9000

Components Used

The information in this document is based on these software and hardware versions:

- Hardware: N9K-C9336C-FX2
- Version: 10.2(3)

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

A Q-in-Q VLAN tunnel enables a service provider to segregate the traffic of different users in their

infrastructure, while still giving the you a full range of VLANs for their internal use by adding a second 802.1Q tag to an already tagged frame.

Selective Q-in-Q with multiple provider VLANs is a tunneling feature that allows user-specific range of VLANs on a port to be associated with one specific provider VLAN.

Q-in-Q allows you to have multiple user VLAN-to-provider VLAN mappings on a port.

Packets that come in with a VLAN tag that matches any of the configured user VLANs on the port are tunneled across the fabric using the properties of the service provider VLAN. The encapsulated packet carries the user VLAN tag as part of the Layer 2 header of the inner packet.

Network Diagram



Configure

The goal of the configuration of Selective Q-in-Q is to map the VLANs from the traffic received in N9KFX2 and N9KFX2-1 on interfaces eth1/1.

The Nexus encapsulates the traffic and sends it with the DOT1Q header VLAN 20 to the provider, simulated by Nexus 93180FX.

Host-A

```
interface Ethernet1/1
  switchport
  switchport mode trunk
  no shutdown

interface Vlan10
  no shutdown
  ip address 192.168.1.1/24
```

Host-B

```
interface Ethernet1/1
  switchport
  switchport mode trunk
  no shutdown

interface Vlan10
  no shutdown
  ip address 192.168.1.2/24
```

N9KFX2

```
interface Ethernet1/1
  switchport
  switchport mode trunk
  switchport vlan mapping all dot1q-tunnel 20
  switchport trunk allowed vlan 20
  spanning-tree port type edge trunk
  spanning-tree bpdupfilter enable
  no shutdown

interface Ethernet1/35
  switchport
  switchport mode trunk
  no shutdown
```

N93180FX

```
system dot1q-tunnel transit

interface Ethernet1/17
  switchport
  switchport mode trunk
  no shutdown

interface Ethernet1/18
  switchport
  switchport mode trunk
  no shutdown
```

N9KFX2-1

```
interface Ethernet1/1
  switchport
  switchport mode trunk
  switchport vlan mapping all dot1q-tunnel 20
  switchport trunk allowed vlan 20
```

```
spanning-tree port type edge trunk
spanning-tree bpdufilter enable
no shutdown
```

```
interface Ethernet1/35
switchport
switchport mode trunk
no shutdown
```

Troubleshoot

The Q-in-Q programming moves from the control plane to the hardware through the components:

:



Consistency Checker

Recent versions of the Nexus switch have consistency checker to use in case misprogramming.

```
N9KFX2# show consistency-checker selective-qinq
```

```
Box level hardware programming checks      : PASS
Fetching ingressVlanXlate entries from slice: 0 HW
Fetching ingressVlanXlate entries from slice: 1 HW
Fetching ingressVlanXlate TCAM entries
Performing port specific checks for intf Eth1/1
Port specific selective QinQ checks for interface Eth1/1 : PASS
Selective QinQ Overall status              : PASS
```

Control plane

Confirm if the VLAN is active on the ports.

```
N9KFX2#sh vlan
```

VLAN Name	Status	Ports
1 default	active	Eth1/1
20 VLAN0020	active	Eth1/1, Eth1/35

Confirm that the VLAN mapping is clearly seen.

```
N9KFX2# sh vlan internal info mapping
```

Per Port Vlan Translation Details

```
=====
Entry No          : 0
-----
 ifindex Eth1/1(0x1a000000)
vlan mapping enabled: TRUE
vlan translation mapping information (count=1):
  Original Vlan    Translated Vlan
  -----
  all              20

Entry No          : 1
-----
 ifindex Eth1/35(0x1a004400)
vlan mapping enabled: FALSE
vlan translation mapping information (count=0):
  Original Vlan    Translated Vlan
  -----
```

ELTM

Confirm if there is any errors increasing or non zero.

```
N9KFX2# show system internal eltm error count
interface config error: 0
dce error: 0
port struct error: 0
port channel error: 0
tunnel error: 0
tlv count error: 0
vpn count error: 0
SI add to DB error: 0
```

Run the command **attach module X**. The X is the module where the physical interface is allocated.

```
N9KFX2# attach module 1
module-1# show system internal eltm info vlan 20
IFTMC Detailed info for VLAN = 20
cr_flags = VLAN , bd = 21(0x15) stg-id = 0 type = 1 stp:0
shut_state = 2 , oper_state = 1
vni = 0, ipmc_idx = 0x0 ipmc_remap_idx = 0x0 ipmc_remap_idx2 = 0x0 dvif: 0
ipmc_mrrouter_idx = 0x0 ipmc_mrrouter_remap_idx = 0x0 ipmc_mrrouter_remap_idx2 = 0x0
pbp_idx=0
eid[0] = 0, alt_eid[0] = 0eid[1] = 0, alt_eid[1] = 0eid[2] = 0, alt_eid[2] = 0eid[3] = 0, alt_eid[3] = 0
```

Active ports: <<<<< Confirm Active ports are listed for the

```
Ethernet1/1(0x1a000000) Ethernet1/35(0x1a004400)
```

Active Local ports:

```
Ethernet1/1(0x1a000000) Ethernet1/35(0x1a004400)
```

Ports using this VLAN as Provider: <<<<< Displays the interface that is doing
Ethernet1/1(0x1a000000)

SDB Active ports:
Ethernet1/1

pbp_idx:0

VLAN Features:

prim_vlan : 0, prim_intf_vlan : 0, vlan mode : 0(CE)
pt_cam_en : 1, ipv4_igmp_snoop : 1, ipv4_pim_snoop : 0
ipv6_mld_snoop : 0, ipv6_pim_snoop : 0, l2l3_lkup_cfg : 1
v4_table_id : 1 (0x00000001), v4_vpn_id : 1 (0x1)
v6_table_id : 2147483649 (0x80000001), v6_vpn_id : 1 (0x1)
age_tmr_sel : 1, sm_en : 0 fcf : 0 omf : 1, dhcp_snoop_en : 0
proxy_learn : 0, iic_en : 0, use_oxid : 0, age_grp_sel : 0
disable_learn : 0, otv_en : 0, global : FALSE
bc_uc_ftag : 4095 , mcast_ftag_num : 1, arp_sup: 0x0 , icmpv6 nd local proxy: 0x0
isdot1qvlan : 0 , num_dot1q_ports: 0 num_sel_dot1q QinVni_ports: 0
num_multi_tag_ports: 0, eid: 0, xconnect: 0, mcast_ftag[1] : {4095}
is_transit : 0 mdns_en: 0,

TAH:VLAN_CONTROL:

vrf: 0 forwarding_vlan: 0 forwarding mode: 0
Flags: 0x0
up: 0 vni: 0 is_sr_evpn: 0

You can check for interface programming. The output displays information about the port programming.

N9KFX2# attach module 1

module-1# show system internal eltm info interface e1/1
IFTMC Detailed info for Interface Ethernet1/1(0x1a000000)
CR_flags = INTF LIF (0x3), LTL = 6144 (0x1800), (S 0x0, P 0x0)
IF_INDEX = Ethernet1/1(0x1a000000), LIF = 4100 (0x1004), SDB LIF = 4100 (0x1004)

State = UP

Layer = L2, Mode = TRUNK, native_vlan = 1 tag_native_vlan = 0 <<< Confirm mode and Layer

AdminMode = 0x100000 is_monitor = 0 src_if_pruning 1

Operational VLAN's (2): <<< Confirm if the VLAN for tunnel and

1,20

voice vlan = 4096

MVR Configured VLAN's (0):

local_port = 1, ldb_sharing = 1, ilm_sharing = 0x1

ldb_port_flags = 0x0, dsm = 0, dn1 = 0

pc_ifindx = 0x0, fpc = 0x0

Interface Features:

ipv4_en : 0, ipv4_mcast_en : 0, df_mask : 0, mpls_en : 0

ipv6_en : 0, ipv6_mcast_en : 0 pim_dr_en : 0 ipsg_en : 0

FC port mode:0 vsan: 0 trunk mode: 0 speed: 0

FCF Enabled : 0 FCF MAC:0000:0000:0000

PTP fid = 0

mpls_strip_en : 0

external port : 0

Operational VSAN's (0):

v4_table_id : 1 (0x00000001), v4_vpn_id : 1 (0x1)

```

v6_table_id      : 2147483649 (0x80000001), v6_vpn_id      : 1 (0x1)
per_pkt_ls_en    : 0, bd                          : N/A (0x0)
icmp_redirect    : 1, ipv6_redirect              : 1
v4_same_if_chk   : 0, v6_same_if_check           : 0
mtu_index        : 1 (0x0001), new_mtu_index       : 1 (0x0001)
mtu              : 1500 (0x05dc), port_trust      : 0
v4_rpf_mode      : 0 , v6_rpf_mode              : 0
v4_rpf_mode (sdb) : 0 , v6_rpf_mode (sdb) : 0
intf_type        : 0x0, intf_flags              : 0x20
sub_flags        : 0x0, port_type              : 0x0, is_aida : 0x0, vpc_dvp_no : 0x0
fcoe_mac_ip_chk : 0 buffer_boost: 0 openflow_en: 0 span_buffer_cfg: 0 span_mode: 0 outer_bd = 512
interface mac: 00:00:00:00:00:00 old_mac: 00:00:00:00:00:00 mac_index:0
Src-Idx:1
mc_index : 4100, egress_vs1_drop: 0, pv_en : 0, is_selective_qinvni_port 1 ether_type 0x8100 vlan_bmp
is_multi_tag_port 0, nat_realm_id 0
global_id 0
enm_if_type : 0
pbp_idx : 0
pinned_ifindex : 0x0
normal_vlan_bmp (0):

```

```

arp_suppress inst[0] 0 arp_suppress inst[1] 0 arp_suppress inst[2] 0 arp_suppress inst[3] 0 arp_suppress
inst[4] 0 arp_suppress inst[5] 0 arp_suppress inst[6] 0 arp_suppress inst[7] 0 arp_suppress inst[8] 0
mdns_en inst[3] 0 mdns_en inst[4] 0 mdns_en inst[5] 0 mdns_en inst[6] 0 mdns_en inst[7] 0 arp_s
y en inst[6] 0 icmpv6 nd local proxy en inst[7] 0 mdns_en inst[0] 0 mdns_en inst[1] 0 mdns_en inst[2] 0
service provider Vlans 0

```

```

vlan_xlt_tlb_en_ingress : 1 num_vlan_xlt_ingr : 1

```

```

Vlan Translation Table (INGRESS) <<< This table displays xlt the vlan used to send to the pro

```

```

-----
in_vlan  xlt_vlan
  1       20

```

```

SDB INFO:

```

```

-----
MVR VLAN's (0):

```

```

STATE = UP, is_mcec = 0, is_mct = 0

```

```

TAH:L2 INFO: 4100

```

```

-----
Flags : 0x41

```

```

NIV Index: 1 Source Chip: 1 Source Port: 72

```

```

slice: 1 slice_port: 0

```

```

is_monitor: 0 lif: 4100 phy_intf_id: 1 vnic_if: 0 learn_en: 1 native_vlan 1

```

```

trunk_port: 1 src_if_check: 1

```

```

FC info: intf_type = 0x0 port_mode = 0x0 port_vsan = 0 port_speed = 0 fcf_mac = 0:0:0:0:0:0

```

IFTMC

For IFTMC you can run the 2 commands. Run the **attach module 1** command before running the next commands.

```

N9KFX2# attach module 1

```

```

module-1# show system internal iftmc info interface e1/1

```

```

module-1# show system internal iftmc info vlan 20

```

Both Commands

- show system internal iftmc info interface ethernet1/1
- show system internal iftmc info vlan 20

Displays similar information as ELTM outputs, make sure to verify you have same values between ELTM and IFTMC modules.

Hardware

To confirm the outputs are correctly programmed on the hardware you can run the command show hardware internal tah interface ethernet 1/1. In the output of the command, you can go to the BdStateTable part.

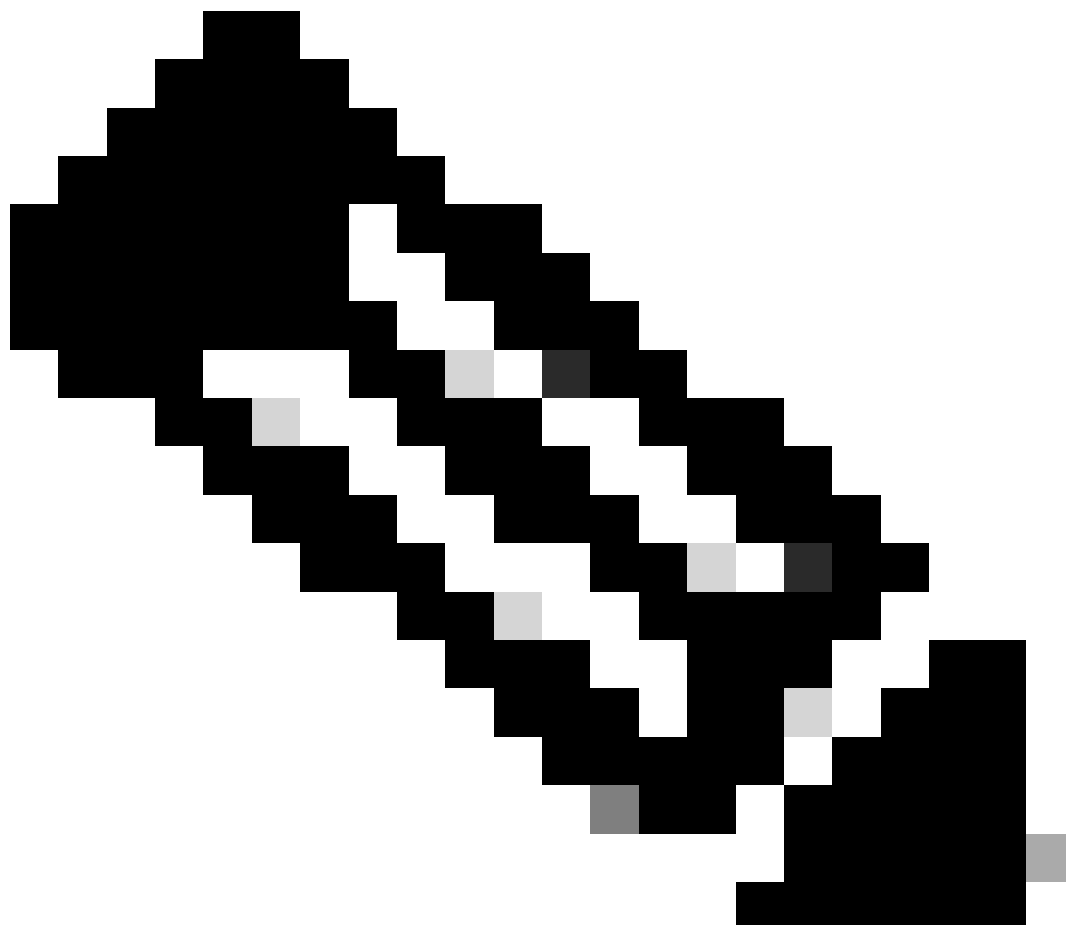
```
N9KFX2-1(config-if)# show hardware internal tah interface ethernet 1/1
BdStateTable:
ENTRY: 1
    info_leaf_flood_dst_ptr : 1                <<<Displays native vlan
    info_leaf_igmp_mld_dst_ptr : 4098
    info_leaf_fid : 1                          <<<Displays native vlan
    info_leaf_vrf : 1
    info_leaf_igmp_snp_en : 1
    info_leaf_l3_bind_check_en : 1
    info_leaf_v4_omf : 1
    info_leaf_unknown_uc_flood : 1
    info_leaf_unknown_mc_flood : 1
    info_leaf_ecn_mark_en : 1
    info_leaf_l2_bind_check_en : 1
    info_leaf_bd_profile_idx : 1
    info_leaf_analytics_en : 1
    info_leaf_flow_collect_en : 1
    info_spine_info0_gipo_idx : 1              <<<Displays native vlan
    info_spine_info0_scope : 64
    info_spine_info0_spare : 0x00000000:0x00100044
    info_spine_info1_spare : 0x00001000:0x00000000
    info_spine_info2_gipo_idx : 512
    info_spine_info2_scope : 12416
    info_spine_info2_spare : 0x00005000:0x10000000
    info_spine_info3_gipo_idx : 384
    info_spine_info3_spare : 0x00000000:0x00000000
=====
```

Useful commands to gather and check for any Q-in-Q issue

- show vlan internal info mapping
- show consistency-checker selective-qinq
- show consistency-checker selective-qinq interface <ethX/Y, port-channel X>
- show hardware internal tah interface ethernet x/y
- show hardware internal tah interface all
- show hardware internal tah sdk event-history error
- show system internal iftmc info interface <>

On Nexus platform you can collect show tech-support more specific than the general, which provides a series of commands:

- **show tech details**
 - **show tech vlan**
 - **show tech usd-all**
 - **show tech eltm**
 - **attach module 1 ; show tech iftmc**
-



Note: Look at the guidelines and limitations sections on the configuration guide listed on Related Information for the version your Nexus is running to confirm that your configuration is not hitting any hardware or software limitations.

Related Information

- [Release 9.3.X](#)
- [Release 10.1.X](#)
- [Release 10.2.X](#)
- [Release 10.3.X](#)
- [Release 10.4.X](#)
- [Cisco Technical Support & Downloads](#)