Troubleshooting VLANs

Troubleshooting VXLAN Issues

The VXLAN data path includes the following paths:

- Multicast encapsulation path—Native Layer 2 packets are encapsulated in VXLAN in the access to network (Layer 2 to Layer 3) direction
- Multicast decapsulation path—Native Layer 2 packets are decapsulated in VXLAN in the network to access (Layer 3 to Layer 2) direction
- Unicast encapsulation path—Native Layer 2 packets are encapsulated in VXLAN in the access to network (Layer 2 to Layer 3) direction
- Unicast decapsulation path—Native Layer 2 packets are decapsulated in VXLAN in the network to access (Layer 3 to Layer 2) direction

Understanding these data paths can help you troubleshoot VXLAN issues.

Caution To troubleshoot VXLAN issues, you need to run Broadcom shell commands. Use these Broadcom shell commands with caution and only under the direct supervision or request of Cisco Support personnel.

Note The Cisco Nexus 9300 Series switches support VXLAN. The Cisco Nexus 9500 Series switches do not.

Packets Dropped in the Multicast Encapsulation Path

Follow these steps if ARP requests or multicast packets are being dropped on the device in the access to network direction.

SUMMARY STEPS

1. Access the Broadcom shell.
2. Check the output of the **stg show** command to see if the ports are in the STP forward state for a given VLAN.
3. Verify if ports are part of the VLAN.
4. Check the output of the **mc show** command to see if the local VLAN ports and encapsulation port are part of the encapsulation flood list.
5. If the output of the **mc show** command is incorrect, exit the Broadcom shell mode, run the following commands, and view the output: **show tech-support pixm**, **show tech-support pixm-all**, and **show tech-support pixmc-all**.

DETAILED STEPS

Procedure

Step 1 Access the Broadcom shell.

Example:

```
switch# bcm-shell module 1
Warning: BCM shell access should be used with caution
Entering bcm shell on module 1
Available Unit Numbers: 0
```

Step 2 Check the output of the **stg show** command to see if the ports are in the STP forward state for a given VLAN.

Example:

```
bcm-shell.0> stg show
STG 6: contains 1 VLAN (3)
  Disable: xe56-xe95
    Block: xe0-xe22,xe24-xe55
  Forward: xe23,hg
```

In this example, VLAN 3 has eth1/24 and uplink tunnel port is eth2/2, so we would expect to see xe23 (1/24) and hg in the output.

Step 3 Verify if ports are part of the VLAN.

Example:

```
bcm-shell.0> vlan show
vlan 3  ports xe23
(0x00000000000000000000000000000000000000000000000000000100000000), untagged
none (0x00000000000000000000000000000000
00000000000000000000000000000000) MCAST_FLOOD_UNKNOWN
```

In this example, xe23 needs to be part of VLAN 3.

Step 4 Check the output of the **mc show** command to see if the local VLAN ports and encapsulation port are part of the encapsulation flood list.

a) Get the encapsulation flood list.

Example:

```
bcm-shell.0> d chg vfi 3
Private image version: R
VFI.ipipe0[3]:
<VP_1=0xc01,VP_0=0x1803,UUC_INDEX=0x1803,UMC_INDEX=0x1803,RSVD_VP_0=1,BC_INDEX=0x1803>
```

In this example, 0x1803 is the encapsulation flood list.

b) Feed the encapsulation flood list into the **mc show** command.

Example:

```
bcm-shell.0> mc show 0x1803
Group 0xc001803 (VXLAN)
        port hg7, encap id 400053
        port xe23, encap id 400057
```

In this example, hg7 is the uplink tunnel port, and xe23 is the local port in the VLAN.

If the uplink is a port channel, all members of the port channel should appear in the output. If the output includes duplicate entries, there will be a corresponding packet replication.

Step 5 If the output of the **mc show** command is incorrect, exit the Broadcom shell mode, run the following commands, and view the output: **show tech-support pixm**, **show tech-support pixm-all**, and **show tech-support pixmc-all**.

Example:

```
bcm-shell.0> exit
switch# show tech-support pixm
switch# show tech-support pixm-all
switch# show tech-support pixmc-all
```

Packets Dropped in the Multicast Decapsulation Path

Follow these steps if ARP requests or multicast packets are being dropped on the device in the network to access direction.

SUMMARY STEPS

1. Check if the packets were sent to the supervisor and if remote VXLAN tunnel endpoint (VTEP) discovery occurred.
2. If the mpls_entry is present in the hardware, check the vlan_xlate table.
3. If the vlan_xlate table has the correct entry for the multicast DIP, check if the VLAN flood list shows the correct members (that is, the members of the VLAN excluding the encapsulation tunnel port).

DETAILED STEPS

Procedure

Step 1 Check if the packets were sent to the supervisor and if remote VXLAN tunnel endpoint (VTEP) discovery occurred.

a) Check if the remote peer was learned in the software.

Example:

```
switch# show nve peers
Interface          Peer-IP           VNI        Up Time
-----------------  ---------------   --------   -----------
nve1               100.100.100.5     10000      00:02:23
```

b) Check if the remote peer was learned in the hardware by checking the mpls_entry table.

Example:

```
switch# bcm-shell module 1
bcm-shell.0> d chg mpls_entry | grep SVP
MPLS_ENTRY.ipipe0[12368]:
<VXLAN_SIP:SVP=0x1751,VXLAN_SIP:SIP=0x66666666,VXLAN_SIP:KEY=0x666666668,VXLAN_SIP:HASH_LSB=0x666
VXLAN_SIP:DATA=0
x1751,VALID=1,KEY_TYPE=8,>
```

c) If the mpls_entry is missing and the source virtual port (SVP) is not present, check if the packets are being sent to the supervisor and check for any IPFIB errors.

Example:

```
bcm-shell.0> show c cpu0
bcm-shell.0> exit
switch# attach module 1
module-1# show system internal ipfib errors
```

Step 2 If the mpls_entry is present in the hardware, check the vlan_xlate table.

Example:

```
module-1# exit
switch# bcm-shell module 1
bcm-shell.0> d chg vlan_xlate | grep 0xe1000003
VLAN_XLATE.ipipe0[8464]:
<VXLAN_DIP:KEY=0x7080000192,VXLAN_DIP:IGNORE_UDP_CHECKSUM=1,VXLAN_DIP:HASH_LSB=3,VXLAN_DIP:DIP=0xe1000003,
XLAN_DIP
:DATA=0x400000,VALID=1,KEY_TYPE=0x12,>
```

The vlan_xlate table should have one entry for the multicast destination IP address (DIP) of the packet. This example shows such an when multicast packets are sent to 225.0.0.3.

Step 3 If the vlan_xlate table has the correct entry for the multicast DIP, check if the VLAN flood list shows the correct members (that is, the members of the VLAN excluding the encapsulation tunnel port).

a) Check the VLAN flood list.

Example:

```
bcm-shell.0> d chg vfi 3
Private image version: R
VFI.ipipe0[3]:
<VP_1=0xc01,VP_0=0x1803,UUC_INDEX=0x1803,UMC_INDEX=0x1803,RSVD_VP_0=1,BC_INDEX=0x1803>
```

For the encapsulation flood list of 0x1803, the corresponding decapsulation flood list would be 0x1c03.

b) Check if the local port is part of the decapsulation flood list.

Example:

```
bcm-shell.0> mc show
Group 0xc001c03 (VXLAN)
        port xe23, encap id 400057
```

xe23 must be part of the decapsulation flood list.

c) Make sure the port is in the forwarding state and part of the VLAN.

Example:

```
bcm-shell.0> stg show
bcm-shell.0> vlan show
```

Packets Dropped in the Unicast Encapsulation Path

Unicast Packets Dropped When VTEP Is Reachable Through a Single Next Hop

Follow these steps if unicast packets are being dropped on the device in the access to network direction and VTEP is reachable through a single next hop.

SUMMARY STEPS

1. Check if the remote peer is discovered in the hardware.
2. Get the mapping of the source virtual port (SVP) to the next hop.
3. Get the port number from the next-hop index.
4. Get the mapping from the port number to the physical port on the chip.
5. Get the egress port to next-hop index mapping.
6. Check the tunnel parameters to make sure that the EGR IP tunnel shows the correct local VTEP IP address in the SIP field.
7. Make sure that the tunnel DIP is programmed.

DETAILED STEPS

Procedure

Step 1 Check if the remote peer is discovered in the hardware.

Example:

```
switch# bcm-shell module 1
bcm-shell.0> d chg mpls_entry | grep SVP
MPLS_ENTRY.ipipe0[12368]:
<VXLAN_SIP:SVP=0x1751,VXLAN_SIP:SIP=0x66666666,VXLAN_SIP:KEY=0x666666668,VXLAN_SIP:HASH_LSB=0x666
VXLAN_SIP:DATA=0
x1751,VALID=1,KEY_TYPE=8,>
```

Make sure a valid source IP address (SIP) exists.

In this example, 102.102.102.102 is the remote VTEP IP address.

Step 2 Get the mapping of the source virtual port (SVP) to the next hop.

Example:

```
bcm-shell.0> d chg ing_dvp_table 0x1751
Private image version: R
ING_DVP_TABLE.ipipe0[5969]:
<VP_TYPE=3,NEXT_HOP_INDEX=0x18,NETWORK_PORT=1,ECMP_PTR=0x18,DVP_GROUP_PTR=0x18,>
```

In this example, the next-hop index is 0x18.

Step 3 Get the port number from the next-hop index.

Example:

```
bcm-shell.0> d chg ing_l3_next_hop 0x18
Private image version: R
ING_L3_NEXT_HOP.ipipe0[24]:
<VLAN_ID=0xfff,TGID=0x88,PORT_NUM=8,MTU_SIZE=0x3fff,MODULE_ID=1,L3_OIF=0x1fff,ENTRY_TYPE=2
ENTRY_INFO_UPPER=3,DV
P_RES_INFO=0x7f,>
```

In this example, the port number is 8.

Step 4 Get the mapping from the port number to the physical port on the chip.

Example:

```
bcm-shell.0> phy info
Phy mapping dump:
      port   id0   id1  addr iaddr          name    timeout
  hg0(  1)  600d  8770   1b1   1b1   TSC-A2/31/4     250000
  hg1(  2)  600d  8770    81    81   TSC-A2/00/4     250000
  hg2(  3)  600d  8770   1ad   1ad   TSC-A2/30/4     250000
  hg3(  4)  600d  8770    85    85   TSC-A2/01/4     250000
  hg4(  5)  600d  8770   189   189   TSC-A2/23/4     250000
  hg5(  6)  600d  8770    ad    ad   TSC-A2/08/4     250000
  hg6(  7)  600d  8770   185   185   TSC-A2/22/4     250000
  hg7(  8)  600d  8770    b1    b1   TSC-A2/09/4     250000
  xe0(  9)  600d  84f9     0    89      BCM84848     250000
  xe1( 10)  600d  84f9     1    8a      BCM84848     250000
  xe2( 11)  600d  84f9     2    8b      BCM84848     250000
  xe3( 12)  600d  84f9     3    8c      BCM84848     250000
```

In this example, port number 8 is hg7.

Step 5 Get the egress port to next-hop index mapping.

Example:

```
bcm-shell.0> g chg egr_port_to_nhi_mapping
EGR_PORT_TO_NHI_MAPPING.hg7[2][0x4001808]=0x18: <NEXT_HOP_INDEX=0x18>
```

In this example, next-hop index 0x18 points to hg7.

Step 6 Check the tunnel parameters to make sure that the EGR IP tunnel shows the correct local VTEP IP address in the SIP field.

Example:

```
bcm-shell.0> d chg egr_ip_tunnel
Private image version: R
EGR_IP_TUNNEL.epipe0[1]:
```

```
<TUNNEL_TYPE=0xb,TTL=0xff,SIP=0x65656565,L4_DEST_PORT=0x2118,ENTRY_TYPE=1,DSCP_SEL=1,>
```

In this example, SIP is the local VTEP IP address (101.101.101.101), L4_DEST_PORT is 0x2118 (port 8472), and DSCP_SEL = 1 means that the inner DSCP packet will be copied to the outer DSCP packet.

Step 7 Make sure that the tunnel DIP is programmed.

Example:

```
bcm-shell.0> d chg egr_dvp_attribute 0x1751
Private image version: R
EGR_DVP_ATTRIBUTE.epipe0[5969]:
<VXLAN:TUNNEL_INDEX=1,VXLAN:DVP_IS_NETWORK_PORT=1,VXLAN:DIP=0x66666666,VP_TYPE=2,>
```

Unicast Packets Dropped When VTEP Is Reachable Through an ECMP Path

Follow these steps if unicast packets are being dropped on the device in the access to network direction and VTEP is reachable through an ECMP path.

SUMMARY STEPS

1. Get the ECMP next hop for a given remote peer virtual port (VP).
2. Convert the ECMP_PTR to decimal and add 200000 to get the port number.
3. Get the list of interfaces in the ECMP next-hop set.
4. Find the members of the port channel.
5. Find the physical next-hop interfaces for the given next-hop index.

DETAILED STEPS

Procedure

Step 1 Get the ECMP next hop for a given remote peer virtual port (VP).

Example:

```
bcm-shell.0> d chg ing_dvp_table 0x1751
Private image version: R
ING_DVP_TABLE.ipipe0[5969]:
<VP_TYPE=3,NEXT_HOP_INDEX=0x108,NETWORK_PORT=1,ECMP_PTR=0x108,ECMP=1,DVP_GROUP_PTR=0x108,>
```

In this example, 0x1751 is the VP number for the remote peer IP address derived from using the d chg mpls_entry output.

Note
If the remote VTEP is reachable through an ECMP path, ECMP=1 needs to be present in the output.

Step 2 Convert the ECMP_PTR to decimal and add 200000 to get the port number.

Example:

```
0x108 (264) + 200000 = 200264
```

In this example, the port number is 200264.

Step 3 Get the list of interfaces in the ECMP next-hop set.

Example:

```
bcm-shell.0> d chg l3 multipath show 200264
Multipath Egress Object 200264
Interfaces: 100606 100607 100608
Reference count: 2
bcm-shell.0> l3 egress show | grep 100606
100606  00:22:bd:f5:1a:60 4095 4101     1t   0        -1   no   no
bcm-shell.0> l3 egress show | grep 100607
100607  00:22:bd:f5:1a:60 4095 4102     2t   0        -1   no   no
bcm-shell.0> l3 egress show | grep 100608
100608  00:22:bd:f5:1a:60 4095 4103     3t   0        -1   no   no
```

In this example, the next-hop interfaces are 1t, 2t, and 3t, which are port channels.

Step 4 Find the members of the port channel.

Example:

```
bcm-shell.0> trunk show
Device supports 1072 trunk groups:
  1024 front panel trunks (0..1023), 256 ports/trunk
  48 fabric trunks (1024..1071), 64 ports/trunk
trunk 0: (front panel, 0 ports)
trunk 1: (front panel, 1 ports)=hg6 dlf=any mc=any ipmc=any psc=portflow (0x9)
trunk 2: (front panel, 1 ports)=hg4 dlf=any mc=any ipmc=any psc=portflow (0x9)
trunk 3: (front panel, 1 ports)=hg7 dlf=any mc=any ipmc=any psc=portflow (0x9)
```

Step 5 Find the physical next-hop interfaces for the given next-hop index.

Example:

```
bcm-shell.0> g chg egr_port_to_nhi_mapping
EGR_PORT_TO_NHI_MAPPING.hg4[2][0x4001805]=0x5f7: <NEXT_HOP_INDEX=0x5f7>
EGR_PORT_TO_NHI_MAPPING.hg6[2][0x4001807]=0x9b3: <NEXT_HOP_INDEX=0x9b3>
EGR_PORT_TO_NHI_MAPPING.hg7[2][0x4001808]=0x5f8: <NEXT_HOP_INDEX=0x5f8>
```

In this example, next-hop index 0x5f7 points to hg4, 0x9b3 points to hg6, and 0x5f8 points hg7.

Packets Dropped in the Unicast Decapsulation Path

Follow these steps if unicast packets are being dropped on the device in the network to access direction.

SUMMARY STEPS

1. Check if the packets were sent to the supervisor and if remote VXLAN tunnel endpoint (VTEP) discovery occurred.
2. If the mpls_entry is present in the hardware, check the vlan_xlate table.
3. Check if the unicast DIP entry is present in the vlan_xlate table.
4. Check if the unicast DIP entry is present in the vlan_xlate table.
5. Make sure that the destination MAC address appears in the Layer 2 MAC address table.

DETAILED STEPS

Procedure

Step 1 Check if the packets were sent to the supervisor and if remote VXLAN tunnel endpoint (VTEP) discovery occurred.

a) Check if the remote peer was learned in the software.

Example:

```
switch# show nve peers
Interface          Peer-IP           VNI        Up Time
-----------------  ----------------  --------   -----------
nve1               100.100.100.5     10000      00:06:54
```

b) Check if the remote peer was learned in the hardware by checking the mpls_entry table.

Example:

```
switch# bcm-shell module 1
bcm-shell.0> d chg mpls_entry | grep SVP
MPLS_ENTRY.ipipe0[12368]:
<VXLAN_SIP:SVP=0x1751,VXLAN_SIP:SIP=0x66666666,VXLAN_SIP:KEY=0x666666668,VXLAN_SIP:HASH_LSB=0x666
VXLAN_SIP:DATA=0
x1751,VALID=1,KEY_TYPE=8,>
```

c) If the mpls_entry is missing and the source virtual port (SVP) is not present, check if the packets are being sent to the supervisor and check for any IPFIB errors.

Example:

```
bcm-shell.0> show c cpu0
bcm-shell.0> exit
switch# attach module 1
module-1# show system internal ipfib errors
```

Step 2 If the mpls_entry is present in the hardware, check the vlan_xlate table.

Example:

```
module-1# exit
switch# bcm-shell module 1
bcm-shell.0> d chg vlan_xlate | grep 0xe1000003
VLAN_XLATE.ipipe0[8464]:
<VXLAN_DIP:KEY=0x7080000192,VXLAN_DIP:IGNORE_UDP_CHECKSUM=1,VXLAN_DIP:HASH_LSB=3,VXLAN_DIP:DIP=0xe1000003,
XLAN_DIP
:DATA=0x400000,VALID=1,KEY_TYPE=0x12,>
```

The vlan_xlate table should have one entry for the multicast destination IP address (DIP) of the packet. This example shows such an when multicast packets are sent to 225.0.0.3.

Step 3 Check if the unicast DIP entry is present in the vlan_xlate table.

Example:

```
bcm-shell.0> d chg vlan_xlate | grep 0x65656565
VLAN_XLATE.ipipe0[14152]:
<VXLAN_DIP:KEY=0x32b2b2b292,VXLAN_DIP:IGNORE_UDP_CHECKSUM=1,VXLAN_DIP:HASH_LSB=0x565
```

```
VXLAN_DIP:DIP=0x65656565,VXLAN_DIP:DATA=0x400000,VALID=1,KEY_TYPE=0x12,>
```

If the entry is present, decapsulation should occur.

Step 4 Check if the unicast DIP entry is present in the vlan_xlate table.

Example:

```
bcm-shell.0> d chg vlan_xlate | grep 0x65656565
VLAN_XLATE.ipipe0[14152]:
<VXLAN_DIP:KEY=0x32b2b2b292,VXLAN_DIP:IGNORE_UDP_CHECKSUM=1,VXLAN_DIP:HASH_LSB=0x565
VXLAN_DIP:DIP=0x65656565,VXLAN_DIP:DATA=0x400000,VALID=1,KEY_TYPE=0x12,>
```

If the entry is present, decapsulation should occur.

Step 5 Make sure that the destination MAC address appears in the Layer 2 MAC address table.

Example:

```
bcm-shell.0> l2 show
mac=00:00:bb:01:00:03 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:cc:01:00:0a vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:bb:01:00:05 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:aa:01:00:0a vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:aa:01:00:07 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:cc:01:00:01 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:bb:01:00:08 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:aa:01:00:01 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:cc:01:00:07 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:cc:01:00:02 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:aa:01:00:04 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:cc:01:00:04 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:aa:01:00:02 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:cc:01:00:09 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:aa:01:00:09 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:bb:01:00:06 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:cc:01:00:06 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:aa:01:00:06 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:bb:01:00:09 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:bb:01:00:04 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:bb:01:00:02 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:aa:01:00:08 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:bb:01:00:07 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:cc:01:00:08 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:bb:01:00:01 vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:cc:01:00:05 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:aa:01:00:03 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:bb:01:00:0a vlan=28772 GPORT=0x80000215Unknown GPORT format
mac=00:00:cc:01:00:03 vlan=28772 GPORT=0x80003401Unknown GPORT format
mac=00:00:aa:01:00:05 vlan=28772 GPORT=0x80003401Unknown GPORT format
```

If the destination MAC address is present, Layer 2 forwarding occurs. Otherwise, packets will be flooded using the decapsulation flood list.

Understanding Broadcom Shell Tables

This section provides information on Broadcom shell tables with respect to VXLAN.

MPLS Entry Table

The MPLS entry (mpls_entry) table contains the following information:

- The IP address of the remote VTEP (SIP)
- The tunnel encapsulation port (SVP)
- The mapping between the VLAN and the VNID (VFI, VN_ID)

When the SIP entry is missing in the mpls_entry table, the packets are sent to the supervisor for VTEP learning. Once the entry is installed in the hardware, the packets should no longer be sent to the supervisor.

Note Some packets will be dropped during the learning phase because software forwarding is not performed for VXLAN packets.

Note Packets that are sent to the supervisor use the class-default CPU queue. There is not currently a dedicated COPP class for VxLAN.

The following example shows a table where the remote VTEP IP address is 100.100.100.1 and VLAN 100 maps to VNID 10000.

```
bcm-shell.0> d chg mpls_entry
Private image version: R
MPLS_ENTRY.ipipe0[6816]: <VXLAN_SIP:SVP=8,VXLAN_SIP:SIP=0x64646401,VXLAN_SIP:KEY=0x646464018
VXLAN_SIP:HASH_LSB=0x401,VXLAN_SIP:DATA=8,VALID=1,KEY_TYPE=8,>
MPLS_ENTRY.ipipe0[8680]:
<VXLAN_VN_ID:VN_ID=0x2710,VXLAN_VN_ID:VFI=0x64,VXLAN_VN_ID:KEY=0x27109
VXLAN_VN_ID:HASH_LSB=0x710,VXLAN_VN_ID:DATA=0x64,VALID=1,KEY_TYPE=9,>
```

In the output, you are looking for one entry per VLAN-VNID mapping. In this example, VN_ID=0x2710 is the VNID in hexadecimal notation, VFI=0x64 is the mapped VLAN in hexadecimal notation, and 0x64 = 100 maps to 0x2710 VNID 10000.

MAC Address Learning

MAC addresses that are learned in VXLAN VLANs appear as learned over an internal translated VLAN (for example, VLAN 100 appears as VLAN 28772).

GPORT refers to the port or virtual port that the MAC address was learned against. For local MAC addresses, there is mapping between the GPORT# and the front panel port#. Remote MAC addresses should be learned against the SVP that is pointing to the tunnel port.

A miss in this table means flood the packet to local ports in the VLAN and the tunnel port. A hit in this table means forward the packet to the corresponding GPORT. If GPORT is the tunnel port, you need to encapsulate the packet in VXLAN. If GPORT is the local port, then regular Layer 2 learned MAC address forwarding occurs.

Note To get the mapping between the GPORT and the front-panel port number, see the Getting the GPORT to Front-Panel Port Number Mapping, on page 14 section.

Ingress DVP Table

The ingress DVP table maps the virtual port to the next-hop index. It is used in the unicast encapsulation path and is indexed by the virtual port. In the case of ECMP, the ECMP=1 field is needed.

The following example shows that for VP 0x1751 the next-hop index is 0x35.

```
bcm-shell.0> d chg ing_dvp_table 0x1751
Private image version: R
ING_DVP_TABLE.ipipe0[5969]:
<VP_TYPE=3,NEXT_HOP_INDEX=0x35,NETWORK_PORT=1,ECMP_PTR=0x35,DVP_GROUP_PTR=0x35,>
```

Ingress Layer 3 Next Hop

The ingress Layer 3 next hop gives the port number for a given next-hop index. It is used in the unicast encapsulation path. You can use the phy_info to get the mapping between the port number and the actual front-panel port number.

```
bcm-shell.0> d chg ing_l3_next_hop
ING_L3_NEXT_HOP.ipipe0[16]:
<VLAN_ID=0xfff,TGID=0x9f,PORT_NUM=0x1f,MTU_SIZE=0x3fff,MODULE_ID=1,L3_OIF=0x1fff,ENTRY_TYPE=2
ENTRY_INFO_UPPER=3,DVP_RES_INFO=0x7f,>
```

VLAN Translate Table

The VLAN translate table is used in the decapsulation path for both VXLAN multicast and unicast. It contains three types of entries:

- One entry per outer multicast group (multicast DIP)
- One entry for the local VTEP (unicast DIP)
- One entry per VLAN, per port

The following example shows a multicast DIP entry.

```
bcm-shell.0> d chg vlan_xlate | grep 0xe1000003
VLAN_XLATE.ipipe0[8464]:
<VXLAN_DIP:KEY=0x7080000192,VXLAN_DIP:IGNORE_UDP_CHECKSUM=1,VXLAN_DIP:HASH_LSB=3
VXLAN_DIP:DIP=0xe1000003,VXLAN_DIP:DATA=0x400000,VALID=1,KEY_TYPE=0x12,>
```

The following example shows a unicast DIP entry.

```
bcm-shell.0> d chg vlan_xlate | grep 0x65656565
VLAN_XLATE.ipipe0[14152]:
<VXLAN_DIP:KEY=0x32b2b2b292,VXLAN_DIP:IGNORE_UDP_CHECKSUM=1,VXLAN_DIP:HASH_LSB=0x565
VXLAN_DIP:DIP=0x65656565,VXLAN_DIP:DATA=0x400000,VALID=1,KEY_TYPE=0x12,>
```

The following example shows one entry per VLAN, per port.

```
bcm-shell.0> d chg vlan_xlate | grep VLAN_ID=3
VLAN_XLATE.ipipe0[3216]:
<XLATE:VLAN_ID=3,XLATE:TGID=0xa0,XLATE:SVP_VALID=1,XLATE:SOURCE_VP=0x201,XLATE:SOURCE_FIELD=0xa0
XLATE:PORT_NUM=0x20,XLATE:OVID=3,XLATE:OTAG=3,XLATE:OLD_VLAN_ID=3,XLATE:MPLS_ACTION=1
XLATE:MODULE_ID=1,XLATE:KEY=0x1805024,XLATE:ITAG=3,XLATE:INCOMING_VIDS=3,XLATE:HASH_LSB=3
XLATE:GLP=0xa0,XLATE:DISABLE_VLAN_CHECKS=1,XLATE:DATA=0x100a000000000000000001,VLAN_ID=3
VALID=1,TGID=0xa0,SVP_VALID=1,SOURCE_VP=0x201,SOURCE_TYPE=1,SOURCE_FIELD=0xa0,PORT_NUM=0x20,OVID=3
OTAG=3,OLD_VLAN_ID=3,MPLS_ACTION=1,MODULE_ID=1,KEY_TYPE=4,KEY=0x1805024,ITAG=3,INCOMING_VIDS=3
HASH_LSB=3,GLP=0xa0,DISABLE_VLAN_CHECKS=1,DATA=0x100a000000000000000001>
```

EGR Port to NHI Mapping

EGR port to NHI mapping maps the next-hop index to the egress port. It is used in the unicast encapsulation path.

```
bcm-shell.0> g chg egr_port_to_nhi_mapping
EGR_PORT_TO_NHI_MAPPING.hg7[2][0x4001808]=0x36: <NEXT_HOP_INDEX=0x36>
```

VLAN Flood Index Table

The VLAN flood index (VFI) table shows the BC/UUC/UMC index for a given VLAN or VFI. The flood index can be used in the output of the **mc show** command to find the members of the VLAN, including the tunnel encapsulation port.

The following example shows how to get the port number.

```
bcm-shell.0> d chg vfi 3
Private image version: R
VFI.ipipe0[3]:
<VP_1=0xc01,VP_0=0x1803,UUC_INDEX=0x1803,UMC_INDEX=0x1803,RSVD_VP_0=1,BC_INDEX=0x1803>
```

The following example shows how to feed this port number into the phy_info to get the front-panel port number.

```
bcm-shell.0> d chg ing_l3_next_hop
ING_L3_NEXT_HOP.ipipe0[16]:
<VLAN_ID=0xfff,TGID=0x9f,PORT_NUM=0x1f,MTU_SIZE=0x3fff,MODULE_ID=1,L3_OIF=0x1fff,ENTRY_TYPE=2
ENTRY_INFO_UPPER=3,DVP_RES_INFO=0x7f,

bcm-shell.0> phy info
Phy mapping dump:
      port   id0   id1  addr iaddr                  name    timeout
  hg0(  1)  600d  8770   1b1   1b1             TSC-A0/31/4     250000
  hg1(  2)  600d  8770    81    81             TSC-A0/00/4     250000
  hg2(  3)  600d  8770   1ad   1ad             TSC-A0/30/4     250000
  hg3(  4)  600d  8770    85    85             TSC-A0/01/4     250000
  hg4(  5)  600d  8770   1a9   1a9             TSC-A0/29/4     250000
  hg5(  6)  600d  8770    89    89             TSC-A0/02/4     250000
  hg6(  7)  600d  8770   195   195             TSC-A0/26/4     250000
  hg7(  8)  600d  8770    a1    a1             TSC-A0/05/4     250000
  hg8(  9)  600d  8770   191   191             TSC-A0/25/4     250000
```

The following example shows the decapsulation route:

```
bcm-shell.0> d chg vlan_xlate
Private image version: R
VLAN_XLATE.ipipe0[768]:
<VXLAN_DIP:NETWORK_RECEIVERS_PRESENT=1,VXLAN_DIP:KEY=0x7080000092,VXLAN_DIP:IGNORE_UDP_CHECKSUM=1
```

```
VXLAN_DIP:HASH_LSB=1,VXLAN_DIP:DIP=0xe1000001,VXLAN_DIP:DATA=0x400001,VALID=1,KEY_TYPE=0x12,>
VLAN_XLATE.ipipe0[1472]:
<VXLAN_DIP:KEY=0x3232320112,VXLAN_DIP:IGNORE_UDP_CHECKSUM=1,VXLAN_DIP:HASH_LSB=0x402
VXLAN_DIP:DIP=0x64646402,VXLAN_DIP:DATA=0x400000,VALID=1,KEY_TYPE=0x12,>
```

Note The NETWORK_RECEIVERS_PRESENT must be set to 0.

Getting the GPORT to Front-Panel Port Number Mapping

Follow these steps to get the mapping between the GPORT and the front-panel port number.

SUMMARY STEPS

1. Use this formula to get the local target logic (LTL) from the GPORT#: LTL# = 0x10000 - 512 + GPORT#
2. Get the ifindex for a given LTL.
3. Get the ifindex to the front-panel port.
4. Display the GPORT to front-panel port number mapping.

DETAILED STEPS

Procedure

Step 1 Use this formula to get the local target logic (LTL) from the GPORT#: LTL# = 0x10000 - 512 + GPORT#

For a GPORT of 0x201, the LTL is 0x10000 + 0x201 (513) - 0x200 (512) = 0x10001.

Step 2 Get the ifindex for a given LTL.

Example:

```
switch# attach module 1
module-1# show system internal pixmc info sdb ltl 0x10001
```

Step 3 Get the ifindex to the front-panel port.

Example:

```
module-1# exit
switch# show int snmp-ifindex | grep 0x1a002e00
Eth1/24    436219392  (0x1a002e00)
```

Step 4 Display the GPORT to front-panel port number mapping.

Example:

```
switch# bcm-shell module 1
bcm-shell.0> l2 show
mac=00:00:00:00:00:00 vlan=0 GPORT=0xc000000 Trunk=0^M
mac=00:00:bb:01:00:03 vlan=28772 GPORT=0x80001751Unknown GPORT format ^M
mac=00:00:cc:01:00:0a vlan=28772 GPORT=0x80000201Unknown GPORT format ^M
```

```
mac=00:00:bb:01:00:05 vlan=28772 GPORT=0x80001751Unknown GPORT format ^M
mac=00:00:aa:01:00:0a vlan=28772 GPORT=0x80000202Unknown GPORT format ^M
```

In this example, MAC address 00:00:bb:01:00:05 is learned over the tunnel, so a GPORT of 0x1751 corresponds to the tunnel SVP. MAC address 00:00:aa:01:00:0a is learned locally, so a GPORT of 0x202 corresponds to the front-panel port.

Finding Which Interface Traffic Will Use for an Egress Port

The following example shows how to find the interface that traffic will use for a given egress port.

```
switch# show system internal ethpm info interface ethernet 2/3 | grep ns_pid
  IF_STATIC_INFO:
port_name=Ethernet2/3,if_index:0x1a006400,ltl=2543,slot=0,nxos_port=50,dmod=1,dpid=9,unit=0
queue=2064,xbar_unitbmp=0x0
ns_pid=8

- dpid=9 is higig8

switch# bcm-shell module 1
bcm-shell.0> g chg egr_port_to_nhi_mapping
EGR_PORT_TO_NHI_MAPPING.hg7[2][0x4001808]=0x36: <NEXT_HOP_INDEX=0x36>
bcm-shell.0> d chg egr_l3_next_hop 0x36
Private image version: R
EGR_L3_NEXT_HOP.epipe0[54]:
<OVID=0x65,MAC_ADDRESS=0x60735cde6e41,L3MC:VNTAG_P=1,L3MC:VNTAG_FORCE_L=1,L3MC:VNTAG_DST_VIF=0x18
L3MC:RSVD_DVP=1,L3MC:INTF_NUM=0x1065,L3MC:FLEX_CTR_POOL_NUMBER=3,L3MC:FLEX_CTR_OFFSET_MODE=3
L3MC:FLEX_CTR_BASE_COUNTER_IDX=0xe41,L3MC:ETAG_PCP_DE_SOURCE=3,L3MC:ETAG_PCP=1
L3MC:ETAG_DOT1P_MAPPING_PTR=1,L3MC:DVP=0x2b9b,L3:OVID=0x65,L3:MAC_ADDRESS=0x60735cde6e41
L3:IVID=0xc83,L3:INTF_NUM=0x1065,IVID=0xc83,INTF_NUM=0x1065,>
```

Finding the Flood List for a VLAN

The following example shows how to find the flood list for a given VLAN.

```
bcm-shell.0> d chg vfi 3
Private image version: R
VFI.ipipe0[3]:
<VP_1=0xc01,VP_0=0x1803,UUC_INDEX=0x1803,UMC_INDEX=0x1803,RSVD_VP_0=1,BC_INDEX=0x1803>
```

Determining if the Encapsulation Port is Part of the Flood List

The following example shows how to determine if the encapsulation port is part of the flood list in the access to network direction.

```
bcm-shell.0> mc show 0x1803
Group 0xc001803 (VXLAN)
        port hg7, encap id 400053
        port xe23, encap id 400057
```