CHAPTER 23

VXLANs

This chapter describes how to identify and resolve problems that might occur when implementing Virtual Extensible Local Area Networks (VXLANs).

This chapter includes the following sections:

Information About VXLANs

Overview

A Virtual Extensibel LAN creates LAN segments by using an overlay approach with MAC-in-UDP encapsulation and a 24-bit segment identifier in the form of a VXLAN ID. The encapsulation carries the original Layer 2 (L2) frame from the Virtual Machine (VM) that is encapsulated from within the Virtual Ethernet Module (VEM). Each VEM is assigned a IP address that are used as the source IP address when encapsulating MAC frames to be sent on the network. You can have multiple vmknics per VEM that are used as sources for this encapsulated traffic. The encapsulation carries the VXLAN identifier used to scope the MAC address of the payload frame. The VXLAN ID to which a VM belongs is indicated within the port profile configuration of the vNIC and is applied when the VM connects to the network. A VXLAN supports three different modes for broadcast, multicast, and MAC distribution mode transport:

- Multicast Mode— A VXLAN uses an IP multicast network to send broadcast, multicast, and unknown unicast flood frames. When a VM joins a VXLAN segment, the server joins a multicast group. Broadcast traffic from the VM is encapsulated and is sent using the multicast outer destination IP address to all the servers in the same multicast group. Subsequent unicast packets are encapsulated and unicast directly to the destination server without multicast IP address.
- Unicast-only Mode— A VXLAN uses each VEM's single unicast IP address as the destination IP address to send broadcast, multicast, and unknown unicast flood frames. Broadcast traffic from the VM is replicated to each VEM by encapsulating it with a VXLAN header and the designated IP address as the outer destination IP address.
    - MAC Distribution Mode(supported only in unicast mode)—In this mode, the unknown unicast flooding is reduced because the Virtual Supervisor Module (VSM) learns all the MAC addresses from the VEMs in all VXLANs and distributes those MAC addresses with VXLAN Tunnel Endpoint (VTEP) IP mappings to other VEMs.

The VXLAN creates LAN segments by using an overlay approach with MAC in IP encapsulation. The encapsulation carries the original Layer 2 (L2) frame from the Virtual Machine (VM) which is encapsulated from within the Virtual Ethernet Module (VEM). Each VEM is assigned an IP address which is used as the source IP address when encapsulating MAC frames to be sent on the network. You can have multiple vmknics per VEM that are used as sources for this encapsulated traffic. The encapsulation carries the VXLAN identifier which is used to scope the MAC address of the payload frame.

VXLAN Tunnel EndPoint

Each VEM requires at least one IP/MAC pair to terminate VXLAN packets. This IP/MAC address pair is known as the VXLAN Tunnel End Point (VTEP) IP/MAC addresses. The VEM supports IPv4 addressing for this purpose. The IP/MAC address that the VTEP uses is configured when you enter the capability vxlan command. You can have a maximum of four VTEPs in a single VEM.

One VTEP per VXLAN segment is designated to receive all broadcast, multicast, and unknown unicast flood traffic for the VEM.

When encapsulated traffic is destined to a VEM that is connected to different subnet, the VEM does not use the VMware host routing table. Instead, the VTEPs initiate the Address Resolution Protocol (ARP) for remote VEM IP addresses. If the VTEPs in the different VEMs are in different subnets, you must configure the upstream router to respond by using the Proxy ARP.

VXLAN Gateway

VXLAN termination (encapsulation and decapsulation) is supported on virtual switches. As a result, the only endpoints that can connect into VXLANs are VMs that are connected to a virtual switch. Physical servers cannot be in VXLANs and routers or services that have traditional VLAN interfaces cannot be used by VXLAN networks. The only way that VXLANs can currently interconnect with traditional VLANs is through VM-based software routers.

VXLAN Trunks

A VXLAN trunk allows you to trunk multiple VXLANs on a single virtual Ethernet interface. In order to achieve this configuration, you must encapsulate a VXLAN-VLAN mapping on the virtual Ethernet interface.

VXLAN-VLAN mappings are configured through the VSM and must always be a 1:1 mapping for each Layer 2 domain. VXLAN-VLAN mappings are applied on a virtual Ethernet interface using a port-profile. A single port profile can support multiple VLAN-VXLAN mappings.

Multi-MAC Capability

You can use multi-MAC addresses to mark a virtual Ethernet interface as capable of sourcing packets from multiple MAC addresses. For example, you can use this feature if you have a virtual Ethernet port and you have enabled VXLAN trunking on it and the VM that is connected to the port bridges packets that are sourced from multiple MAC addresses.

By using this feature, you can easily identify such multi-MAC capable ports and handle live migration scenarios correctly for those ports.

Fragmentation

The VXLAN encapsulation overhead is 50 bytes. In order to prevent performance degradation due to fragmentation, the entire interconnection infrastructure between all VEMs exchange VXLAN packets must be configured to carry 50 bytes more than what the VM VNICs are configured to send. For example, if the default VNIC configuration is 1500 bytes, the VEM uplink port profile, upstream physical switch port, and interswitch links, and any routers if present, must be configured to carry a maximum transmission unit (MTU) of at least 1550 bytes. If that is not possible, we recommend that the MTU within the guest VMs you configure to be smaller by 50 bytes.

If you do not configure a smaller MTU, the VEM attempts to notify the VM if it performs Path MTU (PMTU) Discovery. If the VM does not send packets with a smaller MTU, the VM fragments the IP packets. Fragmentation occurs only at the IP layer. If the VM sends a frame that is too large, the frame will be dropped after VXLAN encapsulation and if the frame does not contain an IP packet.

Scalability

Maximum Number of VXLANs

The Cisco Nexus 1000V supports a total of 4096 VLANs or VXLANs (or a maximum of 2048 VLANs or 2048 VXLANs in any combination that totals 4096).

Supported Features

This section contains the following topics:

- Jumbo Frames, page 23-3
- Disabling the VXLAN Feature Globally, page 23-4

Jumbo Frames

Jumbo frames are supported by the Cisco Nexus 1000V if there is space on the frame to accommodate the VXLAN encapsulation overhead of at least 50 bytes, and the physical switch/router infrastructure has the capability to transport these jumbo-sized IP packets.

Disabling the VXLAN Feature Globally

As a safety precaution, do not use the no feature segmentation command if there are any ports associated with a VXLAN port profile. You must remove all associations before you can disable this feature. You can use the no feature segmentation command to remove all the VXLAN bridge domain configurations on the Cisco Nexus 1000V.

VXLAN Troubleshooting Commands

Use the following commands to display VXLAN attributes.

This section contains the following topics:

- VSM Commands, page 23-4
- VXLAN Gateway Commands, page 23-5

VSM Commands

To display ports belonging to a specific segment:

```
switch(config)# show system internal seg_bd info segment 10000
Bridge-domain: A
Port Count: 11
Veth1
Veth2
Veth3
```

To display the vEthernet bridge domain configuration:

```
switch(config)# show system internal seg_bd info port vethernet 1
Bridge-domain: A
segment_id = 10000
Group IP: 225.1.1.1
```

To display the vEthernet bridge configuration with ifindex as an argument:

```
switch(config)# show system internal seg_bd info port ifindex 0x1c000050
Bridge-domain: A
segment_id = 10000
Group IP: 225.1.1.1
```

To display the total number of bridge domain ports:

```
switch(config)# show system internal seg_bd info port_count
Number of ports: 11
```

To display the bridge domain internal configuration:

```
switch(config)# show system internal seg_bd info bd vxlan-home

Bridge-domain vxlan-home (2 ports in all)
Segment ID: 5555 (Manual/Active)
Group IP: 235.5.5.5
State: UP               Mac learning: Enabled
is_bd_created: Yes
current state: SEG_BD_FSM_ST_READY
pending_delete: 0
port_count: 2
action: 4
```

```
hwbd: 28
pa_count: 0
Veth2, Veth5
switch(config)#
```

To display VXLAN vEthernet information:

```
switch# show system internal seg_bd info port
if_index = <0x1c000010>
Bridge-domain vxlan-pepsi
rid = 216172786878513168
swbd = 4098

if_index = <0x1c000040>
Bridge-domain vxlan-pepsi
rid = 216172786878513216
swbd = 4098

switch#
```

Additional **show** commands:

show system internal seg_bd info {pss | sdb | global | all}

show system internal seg_bd {event-history | errors | mem-stats | msgs}

show system internal seg_bd info (sdb | bd)

VXLAN Gateway Commands

To display VXLAN Gateway information attached to VSM:

```
switch# show module vem
Mod  Ports  Module-Type                       Model               Status
---  -----  --------------------------------  ------------------  ------------
3    7      Virtual Service Module            VXLAN Gateway       ok
```

To display VXLAN Gateway information that is not attached to the VSM:

```
VXLANGW# attach vem
VXLANGW(vem-attach)# ?
  vemcmd     Execute vem command
  vemdpa     Execute vemdpa command
  vemdpalog  Execute vemdpalog command
  vemlog     Execute vemlog command
  vempkt     Execute vempkt command
  vemset     Execute vemset command
switch(vem-attach)#
```

To display VXLAN Gateway statistics:

```
switch(vem-attach)# vemcmd show vxlan-stats
  LTL  Ucast   Mcast/Repl   Ucast   Mcast    Total
       Encaps  Encaps       Decaps  Decaps   Drops
   17    8717          173    8334       0     242
switch(vem-attach)#

switch(vem-attach)# vemcmd show vxlan-stats ltl 17
VXLAN Port Stats for LTL 17
Unicast Encapsulations: 8756
```

```
Multicast Encapsulations/HeadEnd Replications: 173
Unicast Decapsulations: 8372
Multicast Decapsulations: 0
IP Pre-fragmentations: 0
TSO Processed Packets: 0
ICMP Pkt Too Big msgs from upstream: 0
ICMP Pkt Too Big msgs sent to VM: 0
Packets generated by Head End Replication: 172
```

To display the VXLAN Gateway packet path:

```
switch(vem-attach)# vemlog show all
```

To display the bridge-domain configuration on VSM:

```
switch# show bridge-domain
Note - This command is common for both gateway and VEM.

Global Configuration:
Mode: Unicast-only
MAC Distribution: Disable
Note - If you have enabled MAC distribution, the above command will display Enable.
Bridge-domain segment-cisco (3 ports in all)
Segment ID: 9001 (Manual/Active)
Mode: Unicast-only (default)
MAC Distribution: Disable (default)
Group IP: NULL
State: UP               Mac learning: Enabled
Veth2, Veth3, Veth5
```

To display the vlan-vxlan mappings programmed on the VSM:

```
switch# show bridge-domain mapping
```

To display the interfaces on the VSM:

```
switch# show module vtep
```

To display the the bridge domain-to-VTEP mappings that are maintained by the VSM and are pushed to all VEMs:

```
switch# show bridge-domain vtep
```

To displays the MACs learnt on VSM through VEM distribution:

```
switch# sho bridge-domain mac

Bridge-domain: segment-cisco
MAC TABLE Version: 1
Note: You can compare with VEM output using the echo show vxlan version-table command.
MAC Address      Module     Port        VTEP-IP Address  VM-IP Address
------------------------------------------------------------------------------
0050.5683.014e   5          Veth5       10.106.199.117   -
0050.5683.0160   4          Veth2       10.106.199.116   -
0050.5683.0161   4          Veth3       10.106.199.116   -
```

To verify the port configuration on VSM:

```
switch# show int switchport | begin Vethernet2
Name: Vethernet2
  Switchport: Enabled
  Switchport Monitor: Not enabled
  Operational Mode: access
  Access Mode VLAN: 0 (none)
  Access BD name: segment-cisco
```

To verify the VTEP distribution on VSM:

```
switch# show bridge-domain segment-cisco vteps

D: Designated VTEP     I:Forwarding Publish Incapable VTEP

Bridge-domain: segment-cisco
VTEP Table Version: 2
Note: You can compare the VTEP table version with the echo show vxlan version-table on VEM.
Ifindex     Module  VTEP-IP Address
-------------------------------------------------------------------------------
Veth4         4     10.106.199.116(D)
Veth1         5     10.106.199.117(D)
switch#
```

Additional **show** commands:

```
show platform fwm errors

show platform fwm info (vtep | trace | error history)

show platform fwm info error history

show platform fwm event-history msgs

show platform fwm info vlan (all|swbd)
```

VEM Commands

To verify VXLAN vEthernet programming:

```
~ # vemcmd show port segments
                          Native  Seg
  LTL   VSM Port  Mode     SegID   State
   50      Veth5   A        5555   FWD
   51      Veth9   A        8888   FWD
~ #
```

To verify VXLAN vmknic programming:

```
~ # vemcmd show vxlan interfaces
LTL           IP       Seconds since Last
                       IGMP Query Received
(* Interface on which IGMP Joins are sent)
------------------------------------------
 49        10.3.3.3        50         *
 52        10.3.3.6        50
~ #
Use "vemcmd show port vlans" to verify that the vmknics are in the correct transport VLAN.
```

To verify bridge domain creation on the VEM:

```
~ # vemcmd show bd  bd-name vxlan-home
BD 31, vdc 1, segment id 5555, segment group IP 235.5.5.5, swbd 4098, 1 ports,
"vxlan-home"
Portlist:
     50  RedHat_VM1.eth0
```

```
~ #
```

To verify remote IP learning:

```
~ # vemcmd show l2 bd-name vxlan-home
Bridge domain   31 brtmax 4096, brtcnt 2, timeout 300
Segment ID 5555, swbd 4098, "vxlan-home"
Flags:  P - PVLAN  S - Secure  D - Drop
       Type         MAC Address   LTL   timeout   Flags    PVLAN    Remote IP
    Dynamic   00:50:56:ad:71:4e   305         2                     10.3.3.100
     Static   00:50:56:85:01:5b    50         0                      0.0.0.0

~ #
```

To display statistics:

```
~ # vemcmd show vxlan-stats
  LTL  Ucast   Mcast   Ucast   Mcast    Total
       Encaps  Encaps  Decaps  Decaps   Drops
   49       5   14265       4      15       0
   50       6   14261       4      15     213
   51       1      15       0       0      10
   52       0      11       0       0      15

~ #
```

To display detailed per-port statistics for a VXLAN vEthernet/vmknic:

```
~ # vemcmd show vxlan-stats ltl 51
```

To display detailed per-port-per-bridge domain statistics for a VXLAN vmknic for all bridge domains:

```
~ # vemcmd show vxlan-stats ltl <vxlan_vmknic_ltl> bd-all
```

To display detailed per-port-per-bridge domain statistics for a VXLAN vmknic for a specified bridge domain:

```
~ # vemcmd show vxlan-stats ltl vxlan_vmknic_ltl bd-name bd-name
```

To verify the bridge-domain configuration on VEM:

```
switch# vemcmd show bd bd-name segment-cisco
Note - Use the module command to check the details of VEM and gateway on the VSM.

BD 26, vdc 1, segment id 9001, segment group IP 0.0.0.0, swbd 4102, 2 ports,
"segment-cisco"
Segment Mode: Unicast
Note: If MAC distribution is enabled, the above command will displays Segment moode as
Unicast MAC distribution
VTEP DSN: 1 , MAC DSN: 1
Note: You can check the VTEP and MAC download sequence numbers using the vemcmd show
vxlan-vteps and vemcmd show l2 bd bd-name commands.
Portlist:
     53  RedHat_VM1_112.eth4
     54  RedHat_VM1_112.eth5
~ #
```

To display the MAC address table that shows the MACs pushed by the VSM:

```
switch# vemcmd show l2 bd-name segment-cisco
Bridge domain   26 brtmax 4096, brtcnt 3, timeout 300
Segment ID 9001, swbd 4102, "segment-cisco"
Flags:  P - PVLAN  S - Secure  D - Drop
       Type         MAC Address   LTL   timeout   Flags    PVLAN    Remote IP    DSN
    SwInsta   00:50:56:83:01:4e   561         0               10.106.199.117   1
```

```
     Static   00:50:56:83:01:61    54         0                      0.0.0.0   1
     Static   00:50:56:83:01:60    53         0                      0.0.0.0   1

switch#
```

To verify the port configuration on VEM:

```
switch# vemcmd show port
  LTL   VSM Port  Admin Link  State  PC-LTL  SGID  Vem Port  Type
   17     Eth4/1     UP   UP    F/B*    561     0    vmnic0
   49              DOWN   UP    BLK       0        RedHat_VM1_112 ethernet7
   50      Veth8   DOWN   UP    BLK       0        RedHat_VM1_112.eth8
   51      Veth4     UP   UP    FWD       0     0      vmk1  VXLAN
   52              DOWN   UP    BLK       0        RedHat_VM1_112.eth6
   53      Veth2     UP   UP    FWD       0        RedHat_VM1_112.eth4
   54      Veth3     UP   UP    FWD       0        RedHat_VM1_112.eth5
  561        Po2     UP   UP    F/B*      0
```

To verify the VTEP distribution on VEM:

```
switch# vemcmd show vxlan-vteps
Bridge-Domain: segment-cisco Segment ID: 9001
Designated Remote VTEP IPs (*=forwarding publish incapable):
10.106.199.117(DSN: 1),
Note: You can compare the download sequence number against the VTEP download sequence
number using the vemcmnd show bd bd-name.
```

To verify if the MAC address table displays the remote IP learning in the segment-cisco bridge domain:

```
switch# vemcmd show l2 bd-name segment-cisco
Note - Use the module command to check the details of VEM and gateway on the VSM.

Bridge domain   26 brtmax 4096, brtcnt 3, timeout 300
Segment ID 9001, swbd 4102, "segment-cisco"
Flags:  P - PVLAN  S - Secure  D - Drop
       Type         MAC Address   LTL   timeout   Flags    PVLAN    Remote IP    DSN
    Dynamic   00:50:56:83:01:4e   561         1               10.106.199.117   0
     Static   00:50:56:83:01:61    54         0                      0.0.0.0   0
     Static   00:50:56:83:01:60    53         0                      0.0.0.0   0
```

To display the vlan-vxlan mappings programmed on a VEM:

```
switch# vemcmd show vlan-vxlan mapping
Note - Use the module command to check the details of VEM and gateway on the VSM.
```

To display the multi-MAC capable interfaces on a VEM:

```
Note - Use the module command to check the details of VEM and gateway on the VSM.
switch# vemcmd show multi-mac-capable interfaces
```

VEM Packet Path Debugging

Use the following commands to debug VXLAN traffic from a VM on VEM1 to a VM on VEM2.

- VEM1: Verify that packets are coming into the switch from the segment vEthernet.

  vempkt capture ingress ltl *vxlan_veth*

- VEM1: Verify VXLAN ecapsulation.

  vemlog debug sflisp all
  vemlog debug sfvnsegment all

- VEM1: Verify remote IP is learned:

  ```
  vemcmd show l2 bd-name segbdname
  ```

  If the remote IP is not learned, packets are sent multicast encapsulated. For example, an initial ARP request from VM is sent in this manner.

- VEM1: Verify encapsulated packets go out uplink.

  Use the **vemcmd show vxlan-encap ltl** *ltl* command or the **vemcmd show l2lisp-encap mac** *mac* to find out which uplink is being used.

  ```
  vempkt capture egress ltl uplink
  ```

- VEM1: Look at statistics for any failures.

  ```
  vemcmd show vxlan-stats all
  vemcmd show vxlan-stats ltl veth/vxlanvmknic
  ```

- VEM2: Verify encapsulated packets are arriving on the uplink.

  ```
  vempkt capture ingress ltl uplink
  ```

- VEM2: Verify VXLAN decapsulation.

  ```
  vemlog debug sflisp all
  vemlog debug sfvnsegment all
  ```

- VEM2: Verify decapsulated packets go out on VXLAN vEthernet.

  ```
  vempkt capture egress ltl vxlan_veth
  ```

- VEM2: Look at statistics for any failures:

  ```
  vemcmd show vxlan-stats all
  vemcmd show vxlan-stats ltl veth/vxlanvmknic
  ```

Use the following commands to debug the VXLAN packet path:

```
switch# module vem 4 execute vemlog debug vssnet all
switch# module vem 4 execute vemlog debug sfsched all
switch# module vem 4 execute vemlog debug sfport all
switch# module vem 4 execute vemlog debug sflisp all
switch# module vem 4 execute vemlog debug sfvnsegment all
```

Use the following commands to debug the VXLAN packet path from the VSM:

```
switch# module vem 4 execute vemdpalog debug if_bridge_rt all
switch# module vem 4 execute vemdpalog debug sfbd all
switch# module vem 4 execute vemdpalog debug sf_dp_threads all
switch# module vem 4 execute vemdpalog debug sfl2agent all
switch# module vem 4 execute vemlog debug sfporttable all
```

You can view the output for all the above logs by using the **module vem 4 execute vemlog show all** command.

VEM Multicast Debugging

Use the following command to debug VEM multicast.

- IGMP state on the VEM:

  ```
  vemcmd show igmp vxlan_transport_vlan detail
  ```

Note This command does not show any output for the segment multicast groups. To save multicast table space, segment groups are not tracked by IGMP snooping on the VEM.

- IGMP queries:

Use the **vemcmd show vxlan interfaces** command to verify that IGMP queries are being received.

- IGMP joins from vmknic:

Use the **vempkt capture ingress ltl** *first_vxlan_vmknic_ltl* command to see if the VMware stack is sending joins.

Use the **vempkt capture egress ltl** *uplink_ltl* command to see if the joins are being sent out to the upstream switch.

VXLAN Datapath Debugging

Use the commands listed in this section to troubleshot VXLAN problems.

This section contains the following topics:

- Vemlog Debugging, page 23-11
- Vempkt, page 23-12
- Statistics, page 23-12
- Show Commands, page 23-13

Vemlog Debugging

To debug the bridge domain setup or configuration, use the following command:

```
vemlog debug sfbd all
```

To debug port configuration/CBL/vEthernet LTL pinning, use the following command:

```
vemlog debug sfporttable all
```

(for encap/decap setup and decisions)

```
vemlog debug sfvnsegment all
```

To debug for actual packet editing, VXLAN interface handling, and multicast handling, use the following command:

```
vemlog debug sflisp all
```

To debug multicast joins or leaves on the DPA socket, use the following command:

```
echo "debug dpa_allplatform all" > /tmp/dpafifo
```

To debug the bridge domain configuration, use the following command:

```
echo “debug sfl2agent all” > /tmp/dpafifo
```

To debug port configuration, use the following command:

```
echo “debug sfportagent all” > /tmp/dpafifo
```

To debug hitless reconnect (HR) for capability l2-lisp, use the following command:

```
echo “debug sfportl2lisp_cache all” > /tmp/dpafifo
```

To debug CBL programming.

```
echo “debug sfpixmagent all” > /tmp/dpafifo
```

To debug VXLAN agent interacting with the VSM, use the following command:

```
echo "debug sfvxlanagent all" > /tmp/dpafifo
```

Tocheck the VTEP and MAC version, use the following command:

```
~ # echo "show vxlan version-table" > /tmp/dpafifo

Content written to /var/log/vemdpa.log


     Slot       SWBD      VTEP Version      MAC Version
      0         4096           0             0
      6         4102           2             1
      7         4103           0             1
Note: You can compare the MAC version output on the VSM using the show bridge-domain mac
command and VTEP version output on the VSM using the show bridge-domain vtep command.
```

To check the MACs to be distributed on the VSM, use the following command:

```
~ # echo "show vxlan mac-table" > /tmp/dpafifo
Content written to /var/log/vemdpa.log

Flags:  R - Report to VSM  I - VSM Informed
Del* - Stale entry in VSM
No VTEP - NO VTEP. Entry to be removed from VSM
     BD       MAC Address       Ifindex         Id          VTEP           Flags    VM-IP
Count    Retry Count
   4102  00:50:56:83:01:61  0x1c000020  5422209  10.106.199.116      I      0
0
   4102  00:50:56:83:01:60  0x1c000010  5422209  10.106.199.116      I      0
0
```

Vempkt

Vempkt has been enhanced to display VLAN/SegmentID. Use vempkt to trace the packet path through VEM.

- Encap: Capture ingress on Seg-VEth LTL – Egress on uplink
- Decap: Capture ingress on uplink – Egress on Seg-VEth LTL

Statistics

To display a summary of per-port statistics, use the following command:

```
vemcmd show vxlan-stats
```

To display detailed per-port statistics for VXLAN vmknic, use the following command:

vemcmd show vxlan-stats ltl *vxlan_vmknic_ltl*

To display detailed per-port statistics for vEthernet in a VXLAN, use the following command:

```
vemcmd show vxlan-stats ltl vxlan_veth_ltl
```

To display detailed per-port-per-bridge domain statistics for a VXLAN vmknic for all bridge domains, use the following command:

```
vemcmd show vxlan-stats ltl vxlan_vmknic_ltl bd-all
```

To display detailed per-port-per-bridge domain statistics for a VXLAN vmknic for the specified bridge domain, use the following command:

```
vemcmd show vxlan-stats ltl vxlan_vmknic_ltl bd-name bd-name
```

To display which VXLAN vmknic used for encap and subsequent pinning to uplink PC for static MAC learned on port, use the following command:

```
vemcmd show vxlan-encap ltl vxlan_veth_ltl
```

To display which VXLAN vmknic used for encapsulation and subsequent pinning to uplink PC, use the following command:

```
vemcmd show vxlan-encap mac vxlan_vm_mac
```

Show Commands

Table 23-1 lists available **vemcmd show** commands.

Table 23-1 vemcmd Show Commands

Command	Result
vemcmd show vxlan interfaces	Displays the VXLAN encapsulated interfaces.
vemcmd show port vlans	Checks the port programming and CBL state for the bridge domain.
vemcmd show bd	Displays the bridge domain segmentId/group/list of ports.
vemcmd show bd bd-name *bd-name-string*	Displays one segment bridge domain.
vemcmd show l2 all	Displays the remote IP being learned.
vemcmd show l2 bd-name *bd-name-string*	Displays the Layer 2 table for one segment bridge domain.
vemcmd show arp all	Displays the IP-MAC mapping for the outer encapsulated header.