EX Hardware: ACI Packet Forwarding Deep Dive.

Contents

Introduction

This document describes different Forwarding Scenarios using the new Generation ASIC hardware in Application Centric Infrastructure (ACI). It will show how to verify hardware is programmed correctly and we are forwarding packets to the correct destination Endpoints (EP's) in the appropriate Endpoint Groups (EPGs).

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

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

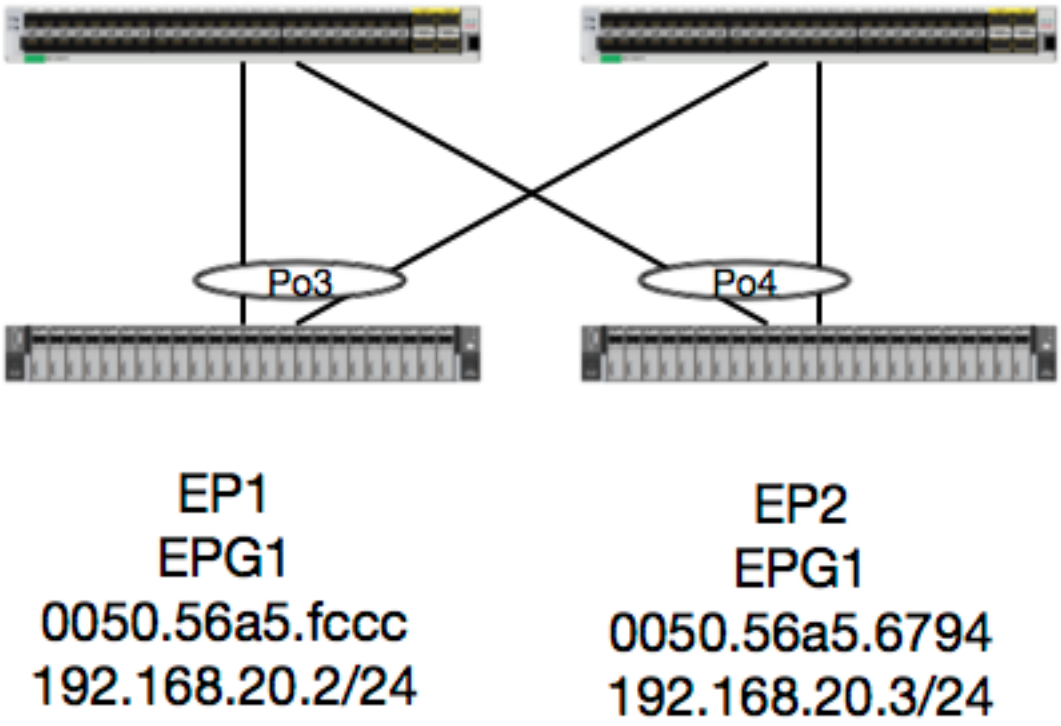
- An ACI Fabric that consists of two Spine switches and two Leaf switches using EX Hardware
- An ESXi host with two uplinks that go to each of the Leaf switches
- Nexus 5000 Device acting as a router.
- An Application Policy Infrastructure Controller (APIC) that is used for initial setup

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, make sure that you understand the potential impact of any command.

Scenarios

2 EP's in same EPG/Same Leaf - Switched Frame

Topology



Given this topology, the flow from EP1 to EP2 is an L2 Flow and should be switched locally on whatever leaf the source traffic comes in on. The first thing to check with Layer 2 (L2) flows is the mac address-table to determine if and where the switch received frames:

```
leaf4# show mac address-table | grep fccc * 30 0050.56a5.fccc dynamic - F F po3 leaf4# show mac address-table | grep 6794 * 30 0050.56a5.6794 dynamic - F F po4
```

In order to see the encapsulation vlan, we can check the EP database as well:

```
leaf4# show endpoint mac 0050.56a5.fccc Legend: O - peer-attached H - vtep a - locally-aged S - static V - vpc-attached p - peer-aged L - local M - span s - static-arp B - bounce +-----------------------+---------------+-----------------+--------------+-------------+ VLAN/ Encap MAC Address MAC Info/ Interface Domain VLAN IP Address IP Info +-------------------------
```

```
---------+--------------+-----------------+-------------+-------------+ 30 vlan-2268
0050.56a5.fccc LV po3 Joey-Tenant:Joey-Internal vlan-2268 192.168.20.2 LV po3 calo2-leaf4# show
endpoint mac 0050.56a5.6794 Legend: O - peer-attached H - vtep a - locally-aged S - static V -
vpc-attached p - peer-aged L - local M - span s - static-arp B - bounce +----------------------
------------+--------------+-----------------+-------------+-------------+ VLAN/ Encap MAC
Address MAC Info/ Interface Domain VLAN IP Address IP Info +---------------------------------
+--------------+-----------------+-------------+-------------+ 30 vlan-2268 0050.56a5.6794 LV
po4 Joey-Tenant:Joey-Internal vlan-2268 192.168.20.3 LV po4
```

We know the FD_VLAN 30 matches, but we can always validate the mapping in software:

```
leaf4# show vlan extended | grep 2268 30 enet CE vlan-2268
```

And of course, we can check the hardware to make sure VLAN 30 maps to VLAN 2268 as the front panel encapsulation.

```
leaf4# vsh_lc
module-1# show system internal eltmc info vlan 30 vlan_id: 30 ::: hw_vlan_id: 22 vlan_type:
FD_VLAN ::: bd_vlan: 28 access_encap_type: 802.1q ::: access_encap: 2268 fabric_encap_type:
VXLAN ::: fabric_encap: 11960 sclass: 32778 ::: scope: 11 untagged: 0 acess_encap_hex: 0x8dc :::
fabric_enc_hex: 0x2eb8 pd_vlan_ft_mask: 0x8 fd_learn_disable: 0 qos_class_id: 0 ::: qos_pap_id:
0 qq_met_ptr: 25 ::: ipmc_index: 0 ingressBdAclLabel: 0 ::: ingBdAclLblMask: 0 egressBdAclLabel:
0 ::: egrBdAclLblMask: 0 qos_map_idx: 0 ::: qos_map_pri: 0 qos_map_dscp: 0 ::: qos_map_tc: 0
vlan_ft_mask: 0xe30 hw_bd_idx: 0 ::: hw_epg_idx: 11267 intf_count: 2 ::: glbl_scp_if_cnt: 2
<SNIPPED>
```

Given that the EP's are learnt in software, we can also validate that hardware programmed the L2 information of these EP's as well. In the new hardware, there is the Hardware Abstraction Layer (HAL) that is the software state of the hardware. HAL's job is to take a software programming requests and push them to hardware.

In order to view L2 hardware information about an endpoint, we can look at the L2 table in HAL for given mac addresses:

```
leaf4# vsh_lc
module-1# show platform internal hal ep l2 mac 0050.56a5.fccc LEGEND: ------- BDId: BD Id BD
Name: BD Name T: EP Type (Pl: Physical Vl: Virtual Xr: Remote EP Mac: Mac L2 IfId: L2 Interface
L2 IfName: L2 IfName FDId: FD Id FD Name: FD Name S Class: S Class Age Intvl: Age Interval P A:
Packet Action (F: Forward, T: Trap to CPU, L: Log & Forward, D: Drop, N: None) S T: Static Ep S
E: Secure EP L D: Learn Disable B N D: Bind Notify Disable E N D: Epg Notify Disable B E: Bounce
Enable I D L: IVxlan Dont Learn SPI: Source Policy Incomplete DPI: Dest Policy Incomplete SPA:
Source Policy Applied DPA: Dest Policy Applied DSS: Dest Shared Service IL: Is Local VUB: Vnid
Use Bd SO: SA Only L2 EP Count: 1
==========================================================================================
====================== B E I S D S D D V BD EP L2 L2 FD S Age P S S L N N B D P P P P S I U S
BdId Name T Mac IfId Ifname FDId Name Class Intvl A T E D D D E L I I A A S L B O
==========================================================================================
====================== 1c BD-28 Pl 00:50:56:a5:fc:cc 16000002 Po3 1e FD-30 800a 29f F 0 0 0 1 0
0 0 0 0 0 0 0 1 0 0 module-1# show platform internal hal ep l2 mac 0050.56a5.6794
==========================================================================================
====================== B E I S D S D D V BD EP L2 L2 FD S Age P S S L N N B D P P P P S I U S
BdId Name T Mac IfId Ifname FDId Name Class Intvl A T E D D D E L I I A A S L B O
==========================================================================================
====================== 1c BD-28 Pl 00:50:56:a5:67:94 16000003 Po4 1e FD-30 800a 29f F 0 0 0 1 0
0 0 0 0 0 0 0 1 0 0
```

Now that we have mapped out the hardware, let's do an ELAM and see where the packet should go.

ELAM

```
leaf4# vsh_lc
module-1# debug platform internal tah elam asic 0 module-1(DBG-TAH-elam)# trigger reset module-
```

```
1(DBG-TAH-elam)# trigger init in-select 6 out-select 0 module-1(DBG-TAH-elam-insel6)# set outer
l2 src_mac 0050.56a5.fccc dst_mac 0050.56a5.6794 module-1(DBG-TAH-elam-insel6)# start module-
1(DBG-TAH-elam-insel6)# stat ELAM STATUS ========== Asic 0 Slice 0 Status Armed Asic 0 Slice 1
Status Triggered module-1(DBG-TAH-elam-insel6)# report | grep ovec
sug_elam_out_sidebnd_no_spare_vec.ovector_idx: 0x9E
```

Great, so Leaf4 received the frame on Asic 0 Slice 1. With ELAM on the new hardware, there is a new field that is very important when troubleshooting: **ovector_idx**. This index is the physical port index that the frame/packet should be forwarded out of. Once you have the ovector_idx, we can use this command to find what port it maps to:

```
module-1(DBG-TAH-elam-insel6)# show platform internal hal l2 port gpd Legend: ------- IfId:
Interface Id IfName: Interface Name I P: Is PC Mbr IfId: Interface Id Uc PC Cfg: UcPcCfg Idx Uc
PC MbrId: Uc Pc Mbr Id As: Asic AP: Asic Port Sl: Slice Sp: Slice Port Ss: Slice SrcId Ovec:
Ovector (slice | srcid) L S: Local Slot Reprogram: L3: Is L3 P: PifTable Xla Idx: Xlate Idx RP:
Rw PifTable Ovx Idx: OXlate Idx IP: If Profile Table N L3: Num. of L3 Ifs RS: Rw SrcId Table NI
L3: Num. of Infra L3 Ifs DP: DPort Table Vif Tid: Vif Tid SP: SrcPortState Table RwV Tid: RwVif
Tid RSP: RwSrcPortstate Table Ing Lbl: Ingress Acl Label UC: UCPcCfg Egr Lbl: Egress Acl Label
UM: UCPcMbr Reprogram: PROF ID: Lport Profile Id VS: VifStateTable HI: LportProfile Hw Install
RV: Rw VifTable Num. of Sandboxes: 1 Sandbox_ID: 0, BMP: 0x0 Port Count: 8
================================================================================================
========================================== Uc Uc | Reprogram | | Rep | I PC Pc L | R I R D R U
U X | L Xla Ovx N NI Vif RwV Ing Egr | V R | PROF H IfId Ifname P Cfg MbrID As AP Sl Sp Ss Ovec
S | P P P S P Sp Sp C M L | 3 Idx Idx L3 L3 Tid Tid Lbl Lbl | S V | ID I
================================================================================================
========================================== 1a004000 Eth1/5 1 0 1d 0 d 0 c 18 18 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 - - 800 0 0 1 0 0 1a005000 Eth1/6 1 0 b 0 e 0 d 1a 1a 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 - - 800 0 0 1 0 0 1a006000 Eth1/7 0 26 5 0 f 0 e 1c 1c 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
D-256 - 800 0 0 1 e 0 1a007000 Eth1/8 0 2e 7 0 10 0 f 1e 1e 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D-84
- 800 0 0 1 30 0 1a01e000 Eth1/31 1 0 2d 0 37 1 e 1c 9c 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 - - 0 0
0 1 0 0 1a01f000 Eth1/32 1 0 3d 0 38 1 f 1e 9e 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 - - 0 0 0 1 0 0
1a030000 Eth1/49 0 2 1 0 49 1 20 38 b8 1 0 0 0 0 0 0 0 0 0 0 1 8 6 2 2 D-24d - 400 0 0 0 1 0
1a031000 Eth1/50 0 3 3 0 29 1 0 0 80 1 0 0 0 0 0 0 0 0 0 0 1 9 7 2 2 D-350 - 400 0 0 0 1 0
```

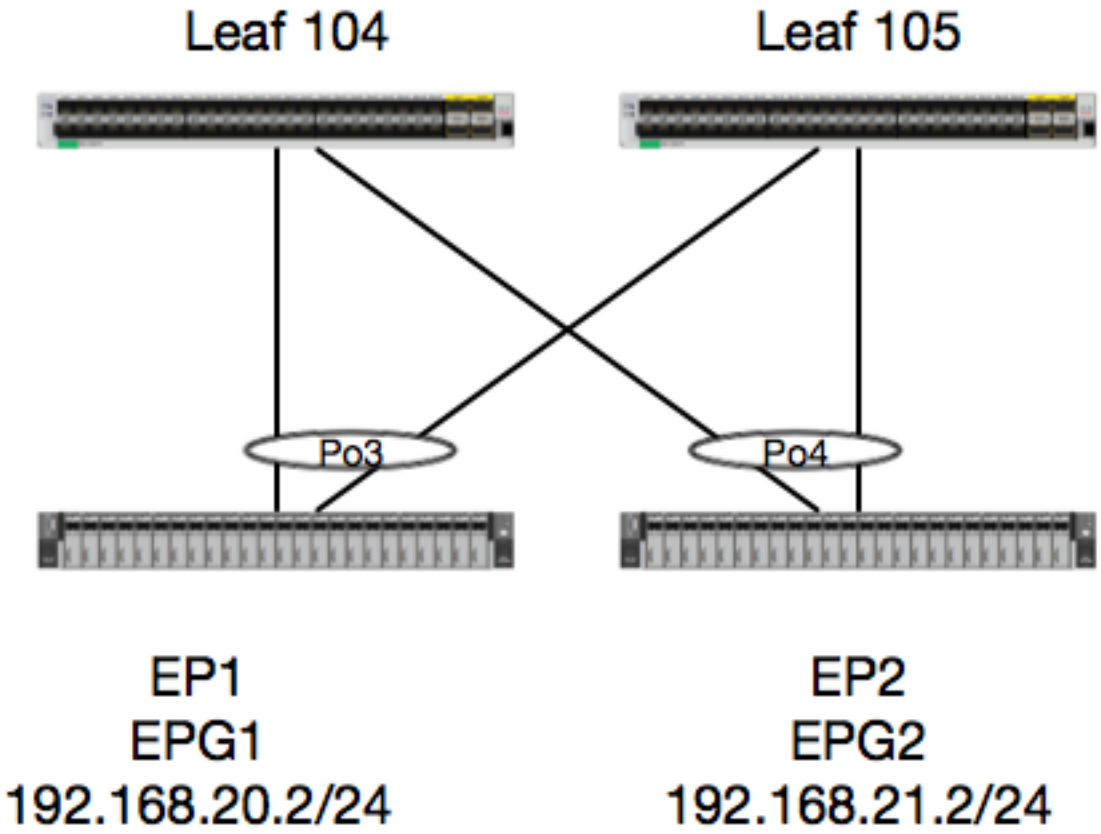
The switch thinks the packet should be forwarded out of interface Ethernet 1/32. Is that PO4 where we have learned that mac addess?

```
leaf4# show port-channel summary
Flags:  D - Down        P - Up in port-channel (members)
        I - Individual  H - Hot-standby (LACP only)
        s - Suspended   r - Module-removed
        S - Switched    R - Routed
        U - Up (port-channel)
        M - Not in use. Min-links not met
        F - Configuration failed
--------------------------------------------------------------------------------
Group Port-       Type     Protocol  Member Ports
      Channel
--------------------------------------------------------------------------------
1     Po1(SU)     Eth      LACP      Eth1/5(P)
2     Po2(SU)     Eth      LACP      Eth1/6(P)
3     Po3(SU)     Eth      LACP      Eth1/31(P)
4 Po4(SU) Eth LACP Eth1/32(P)
```

Yes, so the packet will br forwarded out of Interface 1/32 to the Destination Host.

2 EP's in different EPG/Same Leaf - Routed Packet

Topology



In this example, we will track the packet flow of a packet from EP1 to EP2 where they exist on the same vPC leaf pair. The two EP's are in different EPG's using different BD's.

The first thing to always do is check the EP database to see if we have learned the EP's:

```
leaf4# show endpoint ip 192.168.20.2 Legend: O - peer-attached H - vtep a - locally-aged S -
static V - vpc-attached p - peer-aged L - local M - span s - static-arp B - bounce +-----------
----------------------+--------------+----------------+-------------+-------------+ VLAN/
Encap MAC Address MAC Info/ Interface Domain VLAN IP Address IP Info +-------------------------
---------+--------------+----------------+-------------+-------------+ 30 vlan-2268
0050.56a5.fccc LV po3 Joey-Tenant:Joey-Internal vlan-2268 192.168.20.2 LV po3 calo2-leaf4# show
endpoint ip 192.168.21.2 Legend: O - peer-attached H - vtep a - locally-aged S - static V - vpc-
attached p - peer-aged L - local M - span s - static-arp B - bounce +-------------------------
--------+--------------+----------------+-------------+-------------+ VLAN/ Encap MAC Address
MAC Info/ Interface Domain VLAN IP Address IP Info +---------------------------------+--------
-------+----------------+-------------+-------------+ 8 vlan-2200 0050.56a5.0c11 LV po4 Joey-
Tenant:Joey-Internal vlan-2200 192.168.21.2 LV po4
```

Since we have learned the EP's and know the IP information, we should be able to look at the EP learning info in hardware:

```
leaf4# vsh_lc module-1# show platform internal hal ep l3 all LEGEND: ------- VrfName: Vrf Name
T: Type (Pl: Physical, Vl: Virtual, Xr: Remote) EP IP: Endpoint IP S Class: S Class Age Intvl:
Age Interval S T: Static Ep S E: Secure EP L D: Learn Disable B N D: Bind Notify Disable E N D:
Epg Notify Disable B E: Bounce Enable I D L: IVxlan Dont Learn SPI: Source Policy Incomplete
DPI: Dest Policy Incomplete SPA: Source Policy Applied DPA: Dest Policy Applied DSS: Dest Shared
Service IL: Is Local VUB: Vnid Use Bd SO: SA Only EP NH L3IfName: EP Next Hop L3 If Name NHT:
Next Hop Type (L2: L2 Entry L3: L3 Next Hop) BD Name: L2 NH BD Name EP Mac: EP Mac L3 IfName: L3
NH If Name L2 IfName: L2 If Name FD Name: L2 Entry FD Name IP: L3 NH IP L3 EP Count: 12
================================================================================================
================================================================================ B E
I S D S D D V EP-NH N | Vrf EP S Age S S L N N B D P P P P S I U S L3 H | BD EP L3 L2 FD Name T
IP Class Intvl T E D D D E L I I A A S L B O IfName T | Name Mac IfName Ifname Name IP
================================================================================================
```

```
================================================================================================
common*rewall Pl 10.6.112.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 - L3 - 00:00:00:00:00:00 - - -
0.0.0.0 common*rewall Pl 10.6.114.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 - L3 - 00:00:00:00:00:00 -
- - 0.0.0.0 common*rewall Pl 10.6.114.129 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 - L3 -
00:00:00:00:00:00 - - - 0.0.0.0 common*efault Pl 100.100.101.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0
- L3 - 00:00:00:00:00:00 - - - 0.0.0.0 Joey-T*ternal Pl 192.168.1.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0
1 0 0 - L3 - 00:00:00:00:00:00 - - - 0.0.0.0 Joey-T*ternal Xr 192.168.1.100 8013 128 0 0 0 1 0 0
0 0 0 0 0 0 0 1 0 - L3 - 00:0c:0c:0c:0c:0c Tunnel2 Tunnel2 - 0.0.0.0 Joey-T*ernal2 Pl
192.168.3.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 - L3 - 00:00:00:00:00:00 - - - 0.0.0.0 Joey-
T*ternal Pl 192.168.20.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 - L3 - 00:00:00:00:00:00 - - -
0.0.0.0 Joey-T*ternal Pl 192.168.20.2 800a 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 - L2 BD-28
00:50:56:a5:fc:cc - Po3 FD-30 - Joey-T*ternal Pl 192.168.21.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0
- L3 - 00:00:00:00:00:00 - - - 0.0.0.0 Joey-T*ternal Pl 192.168.21.2 800c 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 - L2 BD-7 00:50:56:a5:0c:11 - Po4 FD-8 - Joey-T*ternal Pl 2001:0:0:100::1 1 0 1 0 0 0
0 0 1 1 0 0 0 0 1 0 0 - L3 - 00:00:00:00:00:00 - - - 0.0.0.0
```

The HAL Layer3 (l3) table is very uself since it gives us VLAN/Port information for l3 learned EP's. We know that the destination exists of a Po4, so the packet should be forwarded out of any port in Po4.

Let's run an ELAM and see what we get!

ELAM

```
leaf4# vsh_lc
module-1# debug platform internal tah elam asic 0 module-1(DBG-TAH-elam)# trigger init in-select
6 out-select 0 module-1(DBG-TAH-elam-insel6)# set outer ipv4 src_ip 192.168.20.2 dst_ip
192.168.21.2 module-1(DBG-TAH-elam-insel6)# start module-1(DBG-TAH-elam-insel6)# stat ELAM
STATUS =========== Asic 0 Slice 0 Status Armed Asic 0 Slice 1 Status Armed module-1(DBG-TAH-
elam-insel6)# stat ELAM STATUS =========== Asic 0 Slice 0 Status Armed Asic 0 Slice 1 Status
Triggered module-1(DBG-TAH-elam-insel6)# report | grep ovec
sug_elam_out_sidebnd_no_spare_vec.ovector_idx: 0x9E
```

Great, so we triggered the packet, and we found that the "ovector_idx" is 0x9E. The ovector index is the outgoing phycial interface index that the packet should be forwarded out of. Let's see what port has that index:

```
module-1(DBG-TAH-elam-insel6)# show platform internal hal l2 port gpd Legend: ------- IfId:
Interface Id IfName: Interface Name I P: Is PC Mbr IfId: Interface Id Uc PC Cfg: UcPcCfg Idx Uc
PC MbrId: Uc Pc Mbr Id As: Asic AP: Asic Port Sl: Slice Sp: Slice Port Ss: Slice SrcId Ovec:
Ovector (slice | srcid) L S: Local Slot Reprogram: L3: Is L3 P: PifTable Xla Idx: Xlate Idx RP:
Rw PifTable Ovx Idx: OXlate Idx IP: If Profile Table N L3: Num. of L3 Ifs RS: Rw SrcId Table NI
L3: Num. of Infra L3 Ifs DP: DPort Table Vif Tid: Vif Tid SP: SrcPortState Table RwV Tid: RwVif
Tid RSP: RwSrcPortstate Table Ing Lbl: Ingress Acl Label UC: UCPcCfg Egr Lbl: Egress Acl Label
UM: UCPcMbr Reprogram: PROF ID: Lport Profile Id VS: VifStateTable HI: LportProfile Hw Install
RV: Rw VifTable Num. of Sandboxes: 1 Sandbox_ID: 0, BMP: 0x0 Port Count: 8
================================================================================================
========================================== Uc Uc | Reprogram | | Rep | I PC Pc L | R I R D R U
U X | L Xla Ovx N NI Vif RwV Ing Egr | V R | PROF H IfId Ifname P Cfg MbrID As AP Sl Sp Ss Ovec
S | P P P S P Sp Sp C M L | 3 Idx Idx L3 L3 Tid Tid Lbl Lbl | S V | ID I
================================================================================================
========================================== 1a004000 Eth1/5 1 0 1d 0 d 0 c 18 18 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 - - 800 0 0 1 0 0 1a005000 Eth1/6 1 0 b 0 e 0 d 1a 1a 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 - - 800 0 0 1 0 0 1a006000 Eth1/7 0 26 5 0 f 0 e 1c 1c 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
D-256 - 800 0 0 1 c 0 1a007000 Eth1/8 0 2f 7 0 10 0 f 1e 1e 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D-
199 - 800 0 0 1 2e 0 1a01e000 Eth1/31 1 0 2d 0 37 1 e 1c 9c 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 - -
0 0 0 1 0 0 1a01f000 Eth1/32 1 0 3d 0 38 1 f 1e 9e 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 - - 0 0 0 1 0
0 1a030000 Eth1/49 0 2 1 0 49 1 20 38 b8 1 0 0 0 0 0 0 0 0 0 0 1 6 4 2 2 D-24d - 400 0 0 0 1 0
1a031000 Eth1/50 0 3 3 0 29 1 0 0 80 1 0 0 0 0 0 0 0 0 0 0 1 5 3 2 2 D-350 - 400 0 0 0 1 0
```

Looks like we should send it out Port 1/32, is that correct?

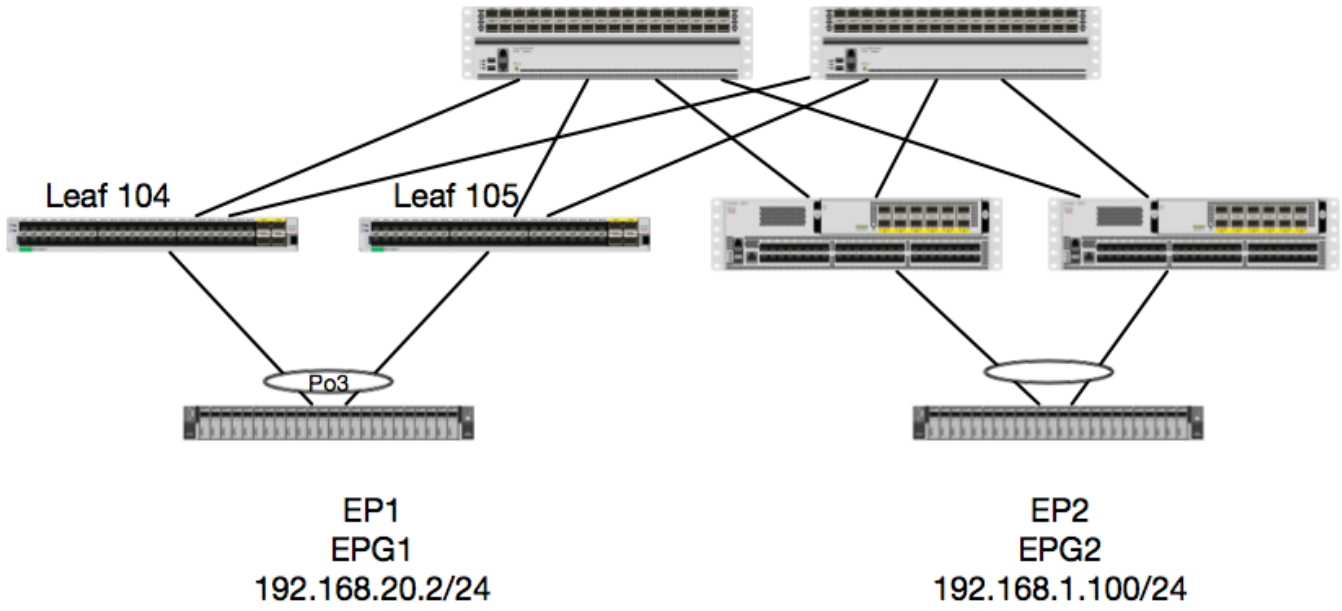
```
leaf4# show port-channel summary
```

```
Flags:  D - Down        P - Up in port-channel (members)
        I - Individual  H - Hot-standby (LACP only)
        s - Suspended   r - Module-removed
        S - Switched    R - Routed
        U - Up (port-channel)
        M - Not in use. Min-links not met
        F - Configuration failed
------------------------------------------------------------------------------
Group Port-       Type     Protocol  Member Ports
      Channel
------------------------------------------------------------------------------
1     Po1(SU)     Eth      LACP      Eth1/5(P)
2     Po2(SU)     Eth      LACP      Eth1/6(P)
3     Po3(SU)     Eth      LACP      Eth1/31(P)
4 Po4(SU) Eth LACP Eth1/32(P)
```

Yes, this is correct.

2 EP's in different EPG/Different Leaf - Routed Packet

Topology



In this example, we will track the packet flow of a packet from EP1 to EP2 where EP1 exists on a EX vPC pair and EP2 exists on a remote Generation 1 vPC Leaf pair. The two EP's are in different EPG's using different BD's.

Again, let's check where the EP's are learnt:

```
leaf4# show endpoint ip 192.168.20.2 Legend: O - peer-attached H - vtep a - locally-aged S -
static V - vpc-attached p - peer-aged L - local M - span s - static-arp B - bounce +------------
----------------------+--------------+-----------------+-------------+-------------+ VLAN/
Encap MAC Address MAC Info/ Interface Domain VLAN IP Address IP Info +--------------------------
---------+--------------+-----------------+-------------+-------------+ 30 vlan-2268
0050.56a5.fccc LV po3 Joey-Tenant:Joey-Internal vlan-2268 192.168.20.2 LV po3 calo2-leaf4# show
endpoint ip 192.168.1.100 Legend: O - peer-attached H - vtep a - locally-aged S - static V -
vpc-attached p - peer-aged L - local M - span s - static-arp B - bounce +----------------------
------------+--------------+-----------------+-------------+-------------+ VLAN/ Encap MAC
Address MAC Info/ Interface Domain VLAN IP Address IP Info +---------------------------------
+--------------+-----------------+-------------+------------+ Joey-Tenant:Joey-Internal
```

192.168.1.100 tunnel2

Now, let's verify what the hardware has programmed:

```
leaf4# vsh_lc
module-1# show platform internal hal ep l3 all LEGEND: ------- VrfName: Vrf Name T: Type (Pl: Physical, Vl: Virtual, Xr: Remote) EP IP: Endpoint IP S Class: S Class Age Intvl: Age Interval S T: Static Ep S E: Secure EP L D: Learn Disable B N D: Bind Notify Disable E N D: Epg Notify Disable B E: Bounce Enable I D L: IVxlan Dont Learn SPI: Source Policy Incomplete DPI: Dest Policy Incomplete SPA: Source Policy Applied DPA: Dest Policy Applied DSS: Dest Shared Service IL: Is Local VUB: Vnid Use Bd SO: SA Only EP NH L3IfName: EP Next Hop L3 If Name NHT: Next Hop Type (L2: L2 Entry L3: L3 Next Hop) BD Name: L2 NH BD Name EP Mac: EP Mac L3 IfName: L3 NH If Name L2 IfName: L2 If Name FD Name: L2 Entry FD Name IP: L3 NH IP L3 EP Count: 12
================================================================================================================================================================================ B E I S D S D D V EP-NH N | Vrf EP S Age S S L N N B D P P P P S I U S L3 H | BD EP L3 L2 FD Name T IP Class Intvl T E D D D E L I I A A S L B O IfName T | Name Mac IfName Ifname Name IP
================================================================================================================================================================================
common*rewall Pl 10.6.112.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 - L3 - 00:00:00:00:00:00 - - - 0.0.0.0 common*rewall Pl 10.6.114.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 - L3 - 00:00:00:00:00:00 - - - 0.0.0.0 common*rewall Pl 10.6.114.129 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 - L3 - 00:00:00:00:00:00 - - - 0.0.0.0 common*efault Pl 100.100.101.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 - L3 - 00:00:00:00:00:00 - - - 0.0.0.0 Joey-T*ternal Pl 192.168.1.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 - L3 - 00:00:00:00:00:00 - - - 0.0.0.0 Joey-T*ternal Xr 192.168.1.100 8013 128 0 0 0 1 0 0 0 0 0 0 0 0 1 0 - L3 - 00:0c:0c:0c:0c:0c Tunnel2 Tunnel2 - 0.0.0.0 Joey-T*ernal2 Pl 192.168.3.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 - L3 - 00:00:00:00:00:00 - - - 0.0.0.0 Joey-T*ternal Pl 192.168.20.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 - L3 - 00:00:00:00:00:00 - - - 0.0.0.0 Joey-T*ternal Pl 192.168.20.2 800a 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 - L2 BD-28 00:50:56:a5:fc:cc - Po3 FD-30 - Joey-T*ternal Pl 192.168.21.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 - L3 - 00:00:00:00:00:00 - - - 0.0.0.0 Joey-T*ternal Pl 192.168.21.2 800c 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 - L2 BD-7 00:50:56:a5:0c:11 - Po4 FD-8 - Joey-T*ternal Pl 2001:0:0:100::1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 - L3 - 00:00:00:00:00:00 - - - 0.0.0.0
```

Hardware thinks the EP exists on Tunnel 2. What is the destination for Tunnel 2?

```
module-1# show system internal eltmc info interface tunnel2 IfInfo: interface: Tunnel2 ::: ifindex: 402718722 iod: 66 ::: state: up Mod: 0 ::: Port: 0 Tunnel Index: 0 ::: Tunnel Dst ip: 0xc0a87843 Tunnel Encap: ivxlan ::: Tunnel VPC Peer: 0 Tunnel Dst ip str: 192.168.120.67 ::: Tunnel ept: 0x1 [SDK Info]: tunnl_name: vrf_id: 2 ::: if_index: 0x18010002 hwencapidx: 0 ::: encaptype: 1 mac_proxy: 0 ::: v4_proxy: 0 v6_proxy: 0 ::: ip_addr_type: 0 ipv4_address: 0xc0a87843 [SDB INFO]: iod: 66 pc_if_index: 0 fab_if_index: 0 sv_if: 0 src_idx: 0 int_vlan: 0 encap_vlan: 0 mod_port_status: 0x41620003 v6_tbl_id: 0x80000002 v4_tbl_id: 0x2 router_mac:00.00.00.00.00.00 unnumbered: 0 trunk_id: 0 tunnel_mod: 0 tunnel_port: 0 tep_ip: 0xc0a87843 ip_if_mode: 0 sdk_vrf_id: 2 mtu: 9366 ::: ipmtu_id: 0 is_fex_fabric: 0
```

Since the destination exists off of a vPC, that Destination IP should be the vPC Virtual IP of the remote leafs. Let's check on a remote leaf and see:

```
leaf1# show system internal epm vpc Local TEP IP : 192.168.160.95 Peer TEP IP : 192.168.160.93 vPC configured : Yes vPC VIP : 192.168.120.67 MCT link status : Up Local vPC version bitmap : 0x7 Peer vPC version bitmap : 0x7 Negotiated vPC version : 3 Peer advertisement received : Yes Tunnel to vPC peer : Up
```

Perfect, so it learnt the Destination EP from the remote vPC pair. Let's see what ELAM sees and verify we are forwarding the packet correectly:

ELAM

```
module-1# debug platform internal tah elam asic 0
module-1(DBG-TAH-elam)# trigger init in-select 6 out-select 0 module-1(DBG-TAH-elam-insel6)# set outer ipv4 src_ip 192.168.20.2 dst_ip 192.168.1.100 module-1(DBG-TAH-elam-insel6)# start module-1(DBG-TAH-elam-insel6)# stat ELAM STATUS ========== Asic 0 Slice 0 Status Armed Asic 0 Slice 1 Status Triggered
```

Now, with remote destinations on EX Hardware, there are 2 ELAM values that are very important when troubleshooting packet flow. The ovector_idx like before, and the encap_idx:

```
module-1(DBG-TAH-elam-insel6)# report | grep ovec sug_elam_out_sidebnd_no_spare_vec.ovector_idx: 0xB8 module-1(DBG-TAH-elam-insel6)# report | grep encap sug_lurw_vec.encap_l2_idx: 0x0 sug_lurw_vec.encap_pcid: 0x0 sug_lurw_vec.encap_idx: 0x6 sug_lurw_vec.encap_vld: 0x1
```

On EX Hardware, we do have the ability to drive the destination port the packet should be forwarded out of. Before, we usually just checked the encap idx and verified that the destination idx was the correct tunnel. Here we can verify what port maps to 8B:

```
module-1(DBG-TAH-elam-insel6)# show platform internal hal l2 port gpd Legend: ------- IfId: Interface Id IfName: Interface Name I P: Is PC Mbr IfId: Interface Id Uc PC Cfg: UcPcCfg Idx Uc PC MbrId: Uc Pc Mbr Id As: Asic AP: Asic Port Sl: Slice Sp: Slice Port Ss: Slice SrcId Ovec: Ovector (slice | srcid) L S: Local Slot Reprogram: L3: Is L3 P: PifTable Xla Idx: Xlate Idx RP: Rw PifTable Ovx Idx: OXlate Idx IP: If Profile Table N L3: Num. of L3 Ifs RS: Rw SrcId Table NI L3: Num. of Infra L3 Ifs DP: DPort Table Vif Tid: Vif Tid SP: SrcPortState Table RwV Tid: RwVif Tid RSP: RwSrcPortstate Table Ing Lbl: Ingress Acl Label UC: UCPcCfg Egr Lbl: Egress Acl Label UM: UCPcMbr Reprogram: PROF ID: Lport Profile Id VS: VifStateTable HI: LportProfile Hw Install RV: Rw VifTable Num. of Sandboxes: 1 Sandbox_ID: 0, BMP: 0x0 Port Count: 8
================================================================================================================================================ Uc Uc | Reprogram | | Rep | I PC Pc L | R I R D R U U X | L Xla Ovx N NI Vif RwV Ing Egr | V R | PROF H IfId Ifname P Cfg MbrID As AP Sl Sp Ss Ovec S | P P P S P Sp Sp C M L | 3 Idx Idx L3 L3 Tid Tid Lbl Lbl | S V | ID I
================================================================================================================================================ 1a004000 Eth1/5 1 0 1d 0 d 0 c 18 18 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 - - 800 0 0 1 0 0 1a005000 Eth1/6 1 0 b 0 e 0 d 1a 1a 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 - - 800 0 0 1 0 0 1a006000 Eth1/7 0 26 5 0 f 0 e 1c 1c 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D-256 - 800 0 0 1 c 0 1a007000 Eth1/8 0 2f 7 0 10 0 f 1e 1e 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D-199 - 800 0 0 1 2e 0 1a01e000 Eth1/31 1 0 2d 0 37 1 e 1c 9c 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 - - 0 0 0 1 0 0 1a01f000 Eth1/32 1 0 3d 0 38 1 f 1e 9e 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 - - 0 0 0 1 0 0 1a030000 Eth1/49 0 2 1 0 49 1 20 38 b8 1 0 0 0 0 0 0 0 0 0 1 6 4 2 2 D-24d - 400 0 0 0 1 0 1a031000 Eth1/50 0 3 3 0 29 1 0 0 80 1 0 0 0 0 0 0 0 0 0 1 5 3 2 2 D-350 - 400 0 0 0 1 0
```

Switch thinks it should forward it to the spine on interface Eth1/49. But how can we verify the encap is correct?

We first need to look at hardware information about the tunnel. We can do this by running this HAL command:

```
module-1(DBG-TAH-elam-insel6)# show platform internal hal tunnel rtep pi Non-Sandbox Mode LEGEND: ------- Tun Ifid: Tunnel Ifid IfName: Tunnel If Name Lid: Logical Id ET: Encap Type V: Vxlan I: IVxlan N: NVGRE VrfId: Vrf Id Vrf Name: Vrf Name IP: Tunnel's IP Hw Enc: Hw Encap Idx IVP: Is VPC Peer IL: Is Local P4: Proxy for v4 P6: Proxy for V6 PM: Proxy for Mac II: Is Ingress Only IC: Is Copy Service C OBd: Copy Service Outer Bd U D: Use DF NBT: Next Base Type E: ECMP N: Next-Hop NB Id: Next Base Id NH cnt: Next Hop Count VrfId: Vrf Id Vrf Name: Vrf Name IP: IP Address Mac: Mac L3 IfId: L3 IfId L3IfName: L3 If Name L2 IfId: L2 IfId L2IfName: L2 If Name Num. of Sandboxes: 1 Sandbox_ID: 0, BMP: 0x0 Remote Tep Count: 15
================================================================================================================================================
================================================================================================================================================
======= I N N | E Vrf Hw V I P P P I I C U B B NH | Vrf L3 L3 L2 L2 IfId Ifname T Lid VrfId Name IP Enc P L 4 6 M I C OBd D T Id Cnt | VrfId Name IP Mac IfId IfName IfId IfName
================================================================================================================================================
================================================================================================================================================
======= 18010002 Tunnel2 I 3005 2 overlay-1 192.168.120.670 0 0 0 0 0 0 0 1 0 E 2 2 2 overlay-1 0.0.0.0 0d:0d:0d:0d:0d:00 1a030001 Eth1/49.1 1a030000 Eth1/4 9 2 overlay-1 0.0.0.0 0d:0d:0d:0d:0d:00 1a031002 Eth1/50.2 1a031000 Eth1/5 0
```

This output gives us a few values we care about:

IfId - The interface ID allocated to the tunnel

IP - The IP of the destination. This should match ELTMC.

L3 IfId - The layer 3 interface(s) the switch can use to forward to the appropriate destination.

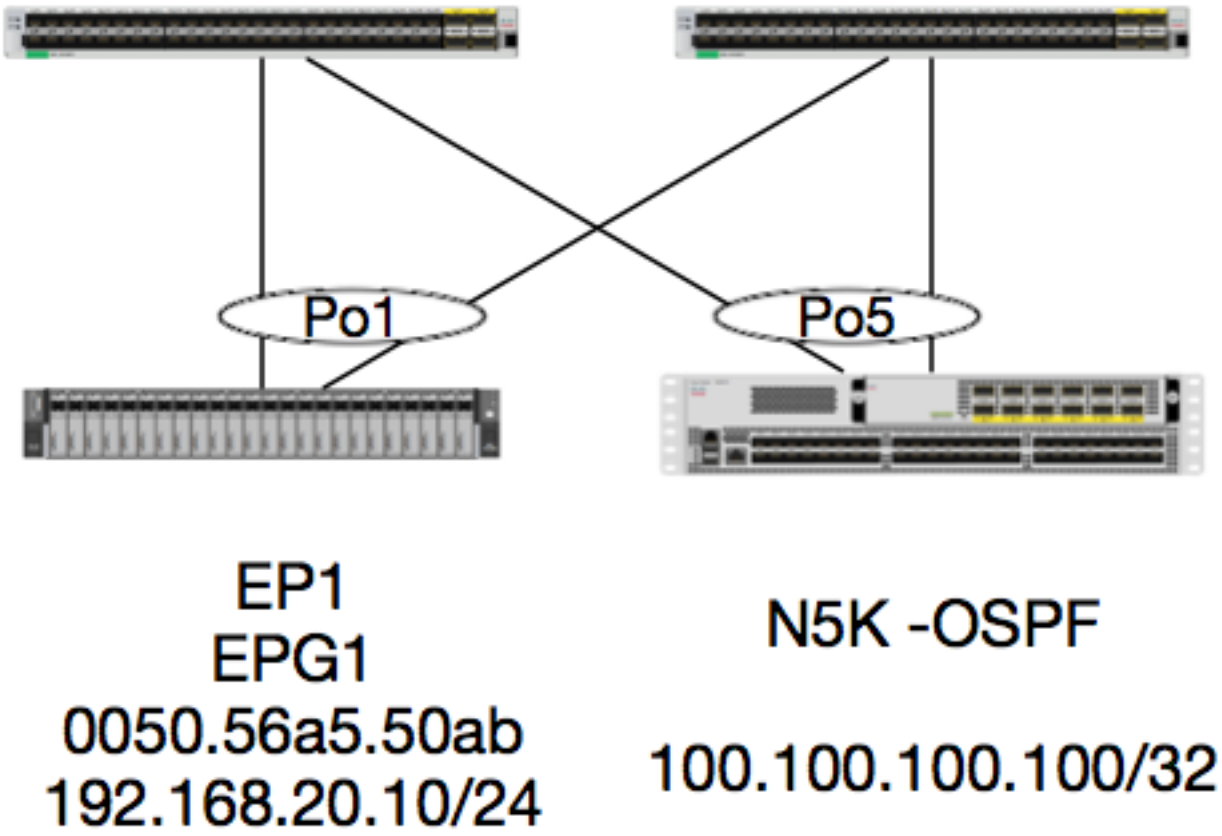
Once we know the IfId, we can verify that the encap we got in the elam matches the tunnel destination:

```
module-1(DBG-TAH-elam-insel9)# show platform internal hal tunnel rtep apd Non-Sandbox Mode
LEGEND: ------- ifId: Interface Id IP: IP address HwVrfId: Hardware Vrf Id SrcTepIdx: Source Tep
Index BDXlate: Egress BDXlate DstInfoIdx: Destination info index RwEncapIdx: Rw Encap Index
ECMPIdx: ECMP Index Num: Number of hops ECMPMbrIdx: ECMP member Index L2 Index: L2 Index
RwDmacIdx: Rw Dmax Index Num. of Sandboxes: 1 Sandbox_ID: 0, BMP: 0x0 Remote Tep Count: 15
================================================================================================
============================== ifId IP HwVrfId BDXlate SrcTepIdx DstInfoIdx RwEncapIdx ECMPIdx
ECMPMbrIdx Num L2Index RwDmacIdx
================================================================================================
============================== 18010002 192.168.120.67 2 1 3a9a 3005 6 0 0 2 1a030000 0 <----
RwEncapIdx is 6! Same as the "encap_idx" in the ELAM Report. 1a031000 1
```

This tunnel has a RwEncapIdx (Re-Write Encap Index) of 6, which is what was displayed in the elam.

1 EP --> L3 out - Routed Flow

Topology



In this example, we will track the packet flow of a packet from EP1 sending ICMP to a loopback on an N5K running OSPF. N5K is connected via an L3Out on the same pair of EX switches.

Since we have verified Local EP programming at the beginning of this document, let's assume the EP is learnt correctly in hardware and continue on to the Route verification.

First, let's check OSPF state and the routing table:

```
leaf6# show ip ospf neighbors vrf jr:sb OSPF Process ID default VRF jr:sb Total number of
neighbors: 2 Neighbor ID Pri State Up Time Address Interface 27.27.27.1 1 FULL/BDR 00:22:39
10.10.27.1 Vlan28 <---- Leaf5 27.27.27.3 1 FULL/DROTHER 00:22:37 10.10.27.3 Vlan28 <---- N5K
leaf6# show ip route vrf jr:sb 100.100.100.100 IP Route Table for VRF "jr:sb" '*' denotes best
ucast next-hop '**' denotes best mcast next-hop '[x/y]' denotes [preference/metric] '%<string>'
in via output denotes VRF <string> 100.100.100.100/32, ubest/mbest: 1/0 *via 10.10.27.3, vlan28,
[110/5], 00:16:58, ospf-default, intra
```

So we know that the routng table shows the next hop as the 5K at 10.10.27.3. Good start, but how can we verify what hardware has?

Let's first check the adjacency table in hardware to make sure we have ARP resolved to 10.10.27.3, and that it is programmed with the correct interface:

```
leaf6# vsh_lc module-1# show forwarding adjacency IPv4 adjacency information, adjacency count 20
next-hop rewrite info interface phy i/f -------------- -------------- -------------- ----------
----- 10.10.27.1 0022.bdf8.19ff Vlan28 Tunnel3 10.10.27.3 8c60.4f02.88fc Vlan28 port-channel5
```

MAC addresses matches that on the 5K:

```
ACI-5548-B# show interface vlan 3117 Vlan3117 is up, line protocol is up Hardware is EtherSVI,
address is 8c60.4f02.88fc Internet Address is 10.10.27.3/29 MTU 1500 bytes, BW 1000000 Kbit, DLY
10 usec
```

On EX Platforms, there is a "hw_vrf_idx" that is assigned to a VRF. This index will be referenced when we verify the hardware programming. Let's find the index:

```
module-1# show system internal eltmc info vrf jr:sb VRF-TABLE: jr:sb vrf_type: tenant :::
context_id: 6 overlay_index: 0 ::: vnid: 2129921 scope: 5 ::: sclass: 16386 v4_table_id: 0x5 :::
v6_table_id: 0x80000005 intf_count: 5 ::: intrn_vlan_id: 0 VRF Intf: Vlan11 ::: src_plcy_incomp:
0 vnid_hex: 0x208001 ::: ingress_policy: 0x1 vrf_intf_list:
Vlan28,Vlan16,Vlan9,Vlan11,loopback2, hw_vrf_idx: 4612 ::: nb_egr_outer_bd: 0 sb_egr_outer_bd: 0
vrf_bd_list: 28,16,11,9, sb_egr_outer_bd: 0 ::: sdk_vrf_id: 5 [SDK Info]: vrf_name: jr:sb
vrf_id: 5 ::: hw_vrf_idx: 4612 vrf_vnid: 2129921 ::: is_infra: 0 tornbinfrahwbd: 0 :::
torsbinfrahwbd: 0 ingressBdAclLabel: 0 ::: ingBdAclLblMask: 0 egressBdAclLabel: 0 :::
egrBdAclLblMask: 0 sg_label: 5 ::: sclass: 16386 sp_incomplete: 1 ::: sclassprio: 3 [SDB INFO]:
v4 table vrf type: 1 vrf id: 5 vnid: 2129921 internal infra vlan: 0 external router
mac:00:22:bd:f8:19:ff v6 table vrf type: 1 vrf id: 5 vnid: 2129921 internal infra vlan: 0
external router mac:00:22:bd:f8:19:ff ::::
```

After we detect the adjacency, HAL should program a route. We can check this using the following command:

```
module-1# show platform internal hal l3 routes | head ----------------------------------------
----------------------------------------------------------------------------------------------
------------------------------------------------- LEGEND: | ------------------------------
----------------------------------------------------------------------------------------------
------------------------------------------------------------- LID: Logical ID RID: Route ID
PID: Physical ID NB-ID:Next-Base ID HIT IDX: Next-Hop HitIndex CLP : Class Priority TBI: Trie
Base Index | SC : Sup-Copy SSR: Src Sup-Redirect DSR: Dst Sup-Redirect TDD :TTL Disable NB:
NextBaseType SDC : Src Direct Connect TRO: Trie Offset | SPI: Src Policy Inc DPI: Dst Policy Inc
DR : Default Route LE :Learn Enable [E:Ecmp/A:Adj] ILL : Is Link Local ISS: Is Shared Services |
RT : Route Type FWD: Forwarding HR : Host Routes EP :Ext Prefixes DLR: Default Lpm Route CLSS:
Class Id RDEL: Route in Deletion | BNE: Bind Notify Enable SNE: Sclass Notify Enable BE : Bounce
Enable IDL :Ivxlan DoNotLearn DL : Dest Local SA : Src Only AI : Age Interval | SF : Static Flag
SH : Src Hit DH: Dest Hit | module-1# show platform internal hal l3 routes ----------------------
----------------------------------------------------------------------------------------------
------------------------------------------------------------- LEGEND: | ----------
----------------------------------------------------------------------------------------------
------------------------------------------------------------------------------- LID:
Logical ID RID: Route ID PID: Physical ID NB-ID:Next-Base ID HIT IDX: Next-Hop HitIndex CLP :
Class Priority TBI: Trie Base Index | SC : Sup-Copy SSR: Src Sup-Redirect DSR: Dst Sup-Redirect
TDD :TTL Disable NB: NextBaseType SDC : Src Direct Connect TRO: Trie Offset | SPI: Src Policy
Inc DPI: Dst Policy Inc DR : Default Route LE :Learn Enable [E:Ecmp/A:Adj] ILL : Is Link Local
```

```
ISS: Is Shared Services | RT : Route Type FWD: Forwarding HR : Host Routes EP :Ext Prefixes DLR:
Default Lpm Route CLSS: Class Id RDEL: Route in Deletion | BNE: Bind Notify Enable SNE: Sclass
Notify Enable BE : Bounce Enable IDL :Ivxlan DoNotLearn DL : Dest Local SA : Src Only AI : Age
Interval | SF : Static Flag SH : Src Hit DH: Dest Hit | ----------------------------------------
----------------------------------------------------------------------------------------------
---------------------------------------------------- | | | | | LID |<-----------------
Trie ------------>|<Dleft Trie>| | | VRF | Prefix/Len | RT| RID | LID | Type| PID | FPID/| HIT
|N| NB-ID | NB Hw | PID | FPID/| TBI |TRO|Ifindex|CLSS|CLP| AI |SH|DH| Flags | |-----|----------
---------------------------------|---|-----|----------|-----| | TID | IDX |B| | Idx | | TID |---
------|---|-------|----|---|----|--|--|------------------| |-----|-----------------------------
-------------|---|-----|----------|-----|<------------------ DLEFT ------------>|-----|------|--
-------|---|-------|----|---|----|--|--|------------------| | | | | | | PID | FPID/| HIT |N|
NB-ID | NB Hw | | | | | | | | | | | | | | | | | | | TID | IDX |B| | Idx | | | | | | | | |
| | | | | | |<------------------ TCAM ------------>| | | | | | | | | | | | | | | | PID
| TCAM | HIT |N| NB-ID | NB Hw | | | | | | | | | | | | | | | | | | | ID | IDX |B| | Idx | | | |
| | | | | | | | | ----------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------
------------------ |Sandbox_ID: 0 Asic Bitmap: 0x0 ----------------------------------------
----------------------------------------------------------------------------------------------
------------------------------------------- module-1# show platform internal hal l3 routes |
egrep 100.100.100.100 | 4612| 100.100.100.100/ 32| UC| e4| 4a04| TRIE| 10| 5/ 0| 6010|A| 7567|
802e| 186a| 1/ 2| 10| 0| 0| f| 3| 0| 0| 0|spi,dpi
```

This output gives us information regarding the next hop route. 4612 is the hw_vrf_idx of the jr:sb VRF. In order for us the verify the Next Hop, the "NB Hw Idx" in TCAM will be used against the next table:

```
module-1# show platform internal hal l3 nexthops Non-Sandbox Mode LEGEND: ------- NHOP ID : Nhop
Identifier (Hex) CONS : H/W S/W info Consistency TYPE : Nexthop Type ACTN : Nexthop Action Vrf :
L3 Vrf of the Nhop L3 INTF : L3 interface index (Hex) L2 INTF : L2 interface index (Hex) BDID Or
RwVRF : Bridge Domain Id Or Rewrite Vrfid (Hex) INFR : ACI Infra valid PVRF : Preserve VRF LRN :
Learn Enabled VRFR : VRF Rewrite PID : Physical ID FPID : FP of this nexthop TLID : Tile Id
within FP HIT IDX : Location of this Nhop (Hex) Mac Entry: TYP : Type INTF : Interface related
Info (Hex) LRN : Learn Info DL : Destination Local MLD : Unused VNB : Vnid use BD DFL : Default
Entry VLD : MacKey Valid FT : FID Type FV : FID Valid FID : FID value (Hex) Mac : L2 MAC Address
L2 Ifabric Info: CLSS : Source Class CLP : Source Class Priority EPG : EndPoint Group BNE : Bind
Notification Enabled SNE : Source Address Notification Enabled CNE : Source class Notification
Enabled DL : iVxlan DL SPI : Source Policy Incomplete DPI : Dest Policy Incomplete IP Address :
IP address Sandbox_ID: 0 Asic Bitmap: 0x0 Summary info for 31 L3 Nexthop objects C T A BDID I P
V T |--------------------Mac Entry------------------|----L2 Ifabric Info----| NHOP O Y C L3 L2
Or N V L R L HIT|T L M V D V|-------------Mac Key-------| C B S C S D| ID N P T INTF INTF RwVRF
F R R F FP I IDX|Y INTF R D L N F L|F F FID | L N N N D P P| (Hex) S E N Vrf (H) (H) (H) R F N R
PID ID D (H)|P (H) N L D B L D|T V (H) Mac |CLSS P EPG E E E L I I| IP Address --------+-+-+-+--
--+--------+--------+----+-+-+-+-+----+--+-+----+-+------+-+-+-+-+-+-+-+-+------+--------------
--+----+-+----+-+-+-+-+-+-+---------------------------- module-1# show platform internal hal l3
nexthops | grep 802e 7567 N I F 5 901001c 16000004 1c 0 0 0 0 2e 9 0 802e 0 22 0 0 0 0 0 1 1 1
1214 8c:60:4f:02:88:fc 0 0 2c0d 0 0 0 0 0 0 10.10.27.3
```

Here, we take the "NB Hw Idx" and map it to the "HIT IDX". This shows us the entry corresponding to the Next Hop MAC/IP. This is the equivalent of looking at "l3 defip show" and "l3 egress show" in Broadcom on Generation 1 ACI Leaf Switches.

As we can see, the table has the correct info:

L2 INTF: 0x16000004 ---> The ifIndex of Port-channel 5

HIT IDX: The Index driven from the Nb Hw Idx in hal l3 routes

MAC: 8c:60:4f:02:88:fc --> MAC of next HOP SVI on 5K

EPG: SCLASS of L3 EPG

IP Address: 10.10.27.3 ---> Next Hop IP of SVI on 5K

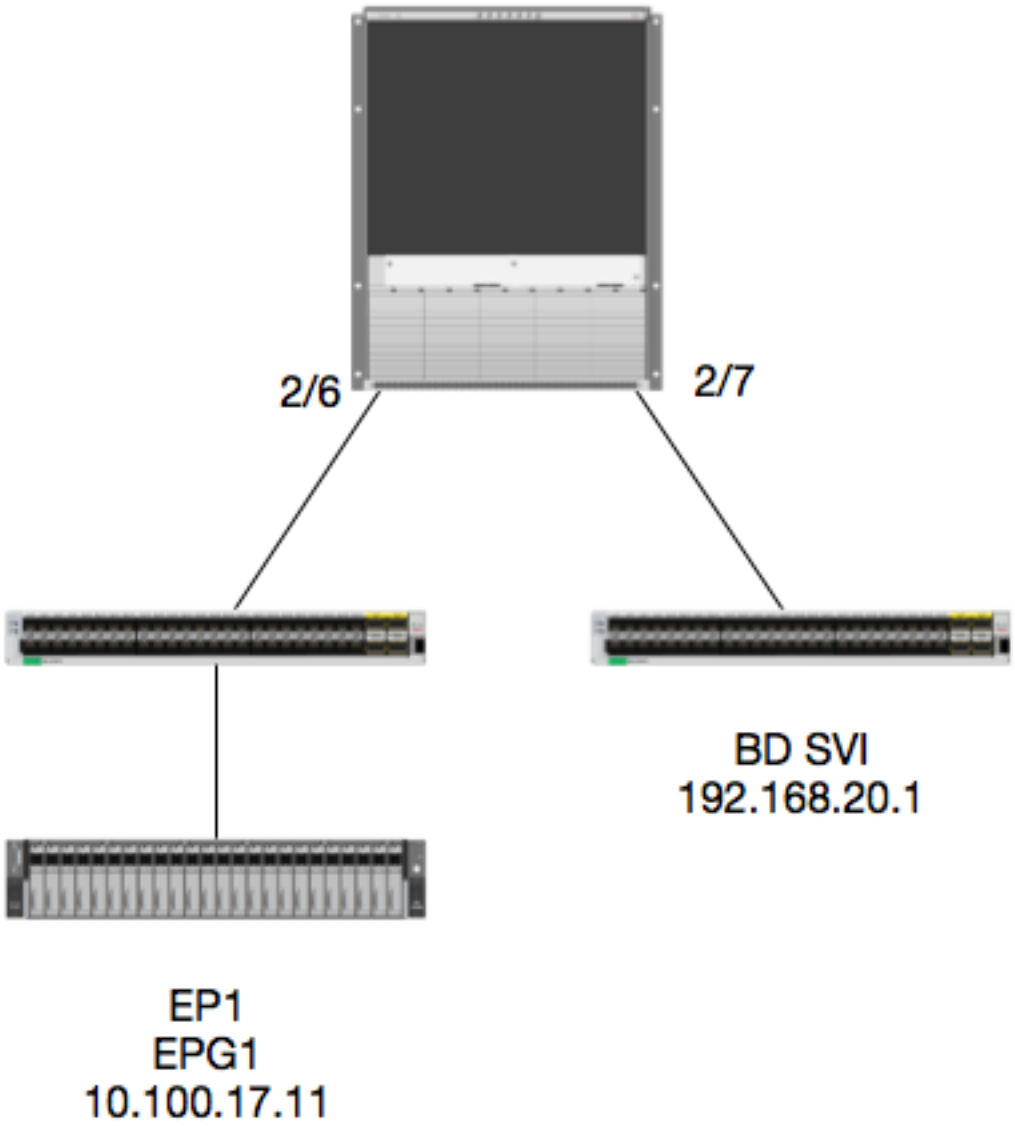
ELAM

```
leaf6# pwd
/var/sysmgr/tmp_logs

leaf6# cat elam_report.txt | grep ip.da
    sug_pr_lu_vec_l3v.ip.da: 0x0000000000000000064646464 leaf6# cat elam_report.txt | grep ip.sa
sug_pr_lu_vec_l3v.ip.sa: 0x0000000000000000C0A8140A leaf6# cat elam_report.txt | grep adj
sug_lurw_vec.dst_addr.adj: 0x8C604F0288FC sug_lurw_vec.dst_addr.adj.padfield: 0x04F0288FC
sug_lurw_vec.dst_addr.adj.idx: 0x2318 sug_lurw_vec.adj_vld: 0x0 leaf6# cat elam_report.txt |
grep macdarslt.hit_idx sug_fpc_lookup_vec.fplu_vec.rslt.macdarslt.hit_idx: 0x802E
```

1 EP --> Remote EP or SVI - Spine Verification

Topology



Logic

In this example, we will track the packet flow of a packet from EP1 destined to a Remote BD Switched Virtual Interface (SVI). The Purpose of this example will be to verify Spine Forwarding to

ensure the packet is sent to the correct Leaf. Let's assume the packet was sent to the Spine Proxy on the ingress Leaf.

On the Spine, let's first verify Council of Oracles Protocol (COOP) for the destination IP since the packet is sent to the Spine Proxy for a lookup:

```
calo1-spine1# show coop internal info ip-db | grep -A 10 192.168.20.1 <------- IP address :
192.168.20.1 Vrf : 2129921 Flags : 0 EP vrf vnid : 2129921 EP IP : 192.168.20.1 Publisher Id :
10.0.224.88 Record timestamp : 11 04 2016 16:41:16 422062712 Publish timestamp : 11 04 2016
16:41:16 424633605 Seq No: 0 Remote publish timestamp: 01 01 1970 00:00:00 0 URIB Tunnel Info
Num tunnels : 1 Tunnel address : 10.0.224.88 <---- REMOTE LEAF Tunnel ref count : 1
```

Let's verify what leaf has that TEP Address:

```
spine1# acidiag fnvread | grep 10.0.224.88 105 1 calo1-leaf5 FDO20160TPS 10.0.224.88/32 leaf
active 0
```

Since we know that the packet is coming into the Spine on Module 2, Port 6, we can attach to Module 2 and look at the Port Layout.

```
spine1# vsh Cisco iNX-OS Debug Shell This shell should only be used for internal commands and
exists for legacy reasons. User should use ibash infrastructure as this will be deprecated.
calo1-spine1# attach module 2 Attaching to module 2 ... To exit type 'exit', to abort type '$.'
No directory, logging in with HOME=/ Bad terminal type: "xterm-256color". Will assume vt100.
Cisco iNX-OS Debug Shell This shell should only be used for internal commands and exists for
legacy reasons. User should use ibash infrastructure as this will be deprecated. Loading parse
tree (LC). Please be patient... module-2# module-2# show platform internal hal l2 port gpd
Legend: ------- IfId: Interface Id IfName: Interface Name I P: Is PC Mbr IfId: Interface Id Uc
PC Cfg: UcPcCfg Idx Uc PC MbrId: Uc Pc Mbr Id As: Asic AP: Asic Port Sl: Slice Sp: Slice Port
Ss: Slice SrcId Ovec: Ovector (slice | srcid) L S: Local Slot Reprogram: L3: Is L3 P: PifTable
Xla Idx: Xlate Idx RP: Rw PifTable Ovx Idx: OXlate Idx IP: If Profile Table N L3: Num. of L3 Ifs
RS: Rw SrcId Table NI L3: Num. of Infra L3 Ifs DP: DPort Table Vif Tid: Vif Tid SP: SrcPortState
Table RwV Tid: RwVif Tid RSP: RwSrcPortstate Table Ing Lbl: Ingress Acl Label UC: UCPcCfg Egr
Lbl: Egress Acl Label UM: UCPcMbr Reprogram: PROF ID: Lport Profile Id VS: VifStateTable HI:
LportProfile Hw Install RV: Rw VifTable Num. of Sandboxes: 1 Sandbox_ID: 0, BMP: 0x0 Port Count:
7
================================================================================================
========================================== Uc Uc | Reprogram | | Rep | I PC Pc L | R I R D R U
U X | L Xla Ovx N NI Vif RwV Ing Egr | V R | PROF H IfId Ifname P Cfg MbrID As AP Sl Sp Ss Ovec
S | P P P S P Sp Sp C M L | 3 Idx Idx L3 L3 Tid Tid Lbl Lbl | S V | ID I
================================================================================================
========================================== 1f5 SpInBndMgmt 0 9de 1a 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 D-2d4 D-3e1 0 0 0 0 1 0 1a080000 Eth2/1 0 9a 1c 0 11 0 10 20 20 1 0 0 0 0 0 0
0 0 0 0 1 b b 1 1 D-f3 D-61 100 0 0 0 1 0 1a081000 Eth2/2 0 9b 22 0 d 0 c 18 18 1 0 0 0 0 0 0 0
0 0 1 c c 1 1 D-1ee D-30b 100 0 0 0 1 0 1a084000 Eth2/5 0 9e 1e 0 3d 1 14 28 a8 1 0 0 0 0 0 0
0 0 0 0 1 1 1 1 1 D-19a D-2ee 100 0 0 0 1 0 1a085000 Eth2/6 0 9f 24 0 39 1 10 20 a0 1 0 0 0 0 0
0 0 0 0 0 1 e e 1 1 D-87 D-184 100 0 0 0 1 0 <--- Interface that connects to Leaf 6 is on ASIC 0
SLICE 1 1a086000 Eth2/7 0 a0 26 0 35 1 c 18 98 1 0 0 0 0 0 0 0 0 0 0 1 d d 1 1 D-1d0 D-357 100 0
0 0 1 0 1a088000 Eth2/9 0 a2 20 1 d 0 c 18 18 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D-3ea D-1a9 100 0
0 0 1 0
```

Now we know which ASIC to Run our ELAM on. ASIC 0.

```
module-2# debug platform internal tah elam asic 0 module-2(DBG-TAH-elam)# trigger reset module-
2(DBG-TAH-elam)# trigger init in-select 13 out-select 0 module-2(DBG-TAH-elam-insel13)# set
inner ipv4 src_ip 10.100.17.11 dst_ip 192.168.20.1 module-2(DBG-TAH-elam-insel13)# start stat
module-2(DBG-TAH-elam-insel13)# stat ELAM STATUS ========== Asic 0 Slice 0 Status Armed Asic 0
Slice 1 Status Armed module-2(DBG-TAH-elam-insel13)# stat ELAM STATUS ========== Asic 0 Slice 0
Status Triggered <---- Packet triggered from FM Asic 0 Slice 1 Status Triggered <---- Packet
triggered from Front Panel
```

Looking at the ELAM, we can find the ovector Index:

```
Front Panel ELAM drove sug_elam_out_sidebnd_no_spare_vec.ovector_idx: 0xB8
```

Now, How do we map 0xb8 to a port? Since we know the packet should get sent to a Fabric Module (FM) for a lookup, we can look at the internal-port mapping to find the dest FM:

```
module-2# show platform internal hal l2 internal-port pi Num. of Sandboxes: 1 Legend: -------
IfId: Interface Id IfName: Interface Name As: Asic AP: Asic Port Sl: Slice SP: Slice Port Ss:
Slice SrcId Ovec: Ovector UcPcCfgId: Uc Pc CfgId Lb Mbrid: LB MbrId Sandbox_ID: 0, BMP: 0x0
Internal Port Count: 32 ========================================================================
UcPc Lb IfId IfName As AP Sl SP Ss Ovec CfgId MbrId
======================================================================== 7d - 0 21 0 20 38 38 0 4
7e - 0 29 1 0 0 80 0 8 7f - 1 21 0 20 38 38 0 c 80 - 1 29 1 0 0 80 0 10 81 - 2 21 0 20 38 38 0
14 82 - 2 29 1 0 0 80 0 18 83 - 3 21 0 20 38 38 0 1c 84 - 3 29 1 0 0 80 0 20 95 - 0 19 0 18 30
30 0 3 96 - 0 49 1 20 38 b8 0 7 <------- Using ASIC0 / Ovec B8, we get MbrId 7, Slice does not
matter 97 - 1 19 0 18 30 30 0 b 98 - 1 49 1 20 38 b8 0 f 99 - 2 19 0 18 30 30 0 13 9a - 2 49 1
20 38 b8 0 17 9b - 3 19 0 18 30 30 0 1b 9c - 3 49 1 20 38 b8 0 1f ad - 0 25 0 24 40 40 0 1 ae -
0 41 1 18 30 b0 0 6 af - 1 25 0 24 40 40 0 9 b0 - 1 41 1 18 30 b0 0 e b1 - 2 25 0 24 40 40 0 11
b2 - 2 41 1 18 30 b0 0 16 b3 - 3 25 0 24 40 40 0 19 b4 - 3 41 1 18 30 b0 0 1e dd - 0 15 0 14 28
28 0 2 de - 0 4d 1 24 40 c0 0 5 df - 1 15 0 14 28 28 0 a e0 - 1 4d 1 24 40 c0 0 d e1 - 2 15 0 14
28 28 0 12 e2 - 2 4d 1 24 40 c0 0 15 e3 - 3 15 0 14 28 28 0 1a e4 - 3 4d 1 24 40 c0 0 1d
```

This MbrId is the interface on the USD that maps to an interface on an FM. We can find out which FM by looking at the USD interfaces and inspecting Port 7:

```
module-2# show platform internal usd port info | grep -A 3 "Int 7" Port 73.0 (Int 7) : Admin UP
Link UP Remote slot22.asic0 slice:1 slice port:32 lcl srcid:56 gbl srcid:184 asic mrl:0xd07c010,
mac mrl:0x12c84010, mac:16, chan:0 speed 106G serdes: 0x328 0x329 0x32a 0x32b
```

The "slot" is 0 based, and the FM numbering is 1 based, so we need to add 1 to the number listed here. This means that the packet should be sent to FM 23.

Synthetic IP

Just like in Alpine, there is a synthetic IP used as the Outer IP address to determine the hash for the COOP lookup. In order to find this, you need to run this command and grep for the inner DST IP:

```
module-2(DBG-TAH-elam-insel7)# show forwarding route synthetic vrf all | grep 192.168.20.1
SYNTH-88 1.203.211.185/32 0x208001 192.168.20.1
```

This shows us that 1.203.211.185 is our synthetic IP. Based on this, we can also set the "Outer DST IP" on our FM elam to be this. We should trigger on the FM:

Fabric Module ELAM

```
module-23(DBG-TAH-elam-insel7)# trigger reset module-23(DBG-TAH-elam)# trigger init in-select 13
out-select 0 module-23(DBG-TAH-elam-insel13)# set outer ipv4 dst_ip 1.203.211.185 <----- DST IP
IS THE SYNTHETIC IP module-23(DBG-TAH-elam-insel13)# set inner ipv4 src_ip 10.100.17.11 dst_ip
192.168.20.1 module-23(DBG-TAH-elam-insel13)# start stat module-23(DBG-TAH-elam-insel13)# stat
ELAM STATUS ========== Asic 0 Slice 0 Status Armed Asic 0 Slice 1 Status Armed Asic 0 Slice 2
Status Armed Asic 0 Slice 3 Status Armed Asic 0 Slice 4 Status Armed Asic 0 Slice 5 Status Armed
module-23(DBG-TAH-elam-insel13)# stat ELAM STATUS ========== Asic 0 Slice 0 Status Armed Asic 0
Slice 1 Status Armed Asic 0 Slice 2 Status Triggered <---- Triggered on SLICE 2 Asic 0 Slice 3
Status Armed Asic 0 Slice 4 Status Armed Asic 0 Slice 5 Status Armed
```

Obviously, dump the full report, but let's look at the ovector_idx for this packet that we triggered:

lac_elam_out_sidebnd_no_spare_vec.ovector_idx: **0x20** <----- Ovector Index used in the command below

How do we figure out which interface has that ovector?. On the FM, run this:

```
module-23(DBG-TAH-elam-insel13)# show platform internal hal l2 port gpd Legend: ------- IfId:
Interface Id IfName: Interface Name I P: Is PC Mbr IfId: Interface Id Uc PC Cfg: UcPcCfg Idx Uc
PC MbrId: Uc Pc Mbr Id As: Asic AP: Asic Port Sl: Slice Sp: Slice Port Ss: Slice SrcId Ovec:
Ovector (slice | srcid) L S: Local Slot Reprogram: L3: Is L3 P: PifTable Xla Idx: Xlate Idx RP:
Rw PifTable Ovx Idx: OXlate Idx IP: If Profile Table N L3: Num. of L3 Ifs RS: Rw SrcId Table NI
L3: Num. of Infra L3 Ifs DP: DPort Table Vif Tid: Vif Tid SP: SrcPortState Table RwV Tid: RwVif
Tid RSP: RwSrcPortstate Table Ing Lbl: Ingress Acl Label UC: UCPcCfg Egr Lbl: Egress Acl Label
UM: UCPcMbr Reprogram: PROF ID: Lport Profile Id VS: VifStateTable HI: LportProfile Hw Install
RV: Rw VifTable Num. of Sandboxes: 1 Sandbox_ID: 1, BMP: 0x1 Port Count: 8
================================================================================================
========================================== Uc Uc | Reprogram | | Rep | I PC Pc L | R I R D R U
U X | L Xla Ovx N NI Vif RwV Ing Egr | V R | PROF H IfId Ifname P Cfg MbrID As AP Sl Sp Ss Ovec
S | P P P S P Sp Sp C M L | 3 Idx Idx L3 L3 Tid Tid Lbl Lbl | S V | ID I
================================================================================================
========================================== ae fc0-lc1:0-0 1 0 3 0 11 0 10 20 20 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 - - 0 0 0 0 0 0 <----------- Interface points to LC1 ASIC 0 / SLICE 0 af fc0-
lc1:0-1 1 0 4 0 3d 2 c 18 98 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 - - 0 0 0 0 0 0 b0 fc0-lc1:1-0 1 0
13 0 d 0 c 18 18 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 - - 0 0 0 0 0 0 b1 fc0-lc1:1-1 1 0 14 0 39 2 8
10 90 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 - - 0 0 0 0 0 0 b2 fc0-lc1:2-0 1 0 23 0 5d 3 14 28 e8 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 - - 0 0 0 0 0 0 b3 fc0-lc1:2-1 1 0 24 0 21 1 8 10 50 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 - - 0 0 0 0 0 0 b4 fc0-lc1:3-0 1 0 33 0 51 3 8 10 d0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 - - 0 0 0 0 0 0
```

That ovector maps to LC1 (Line card in slot 2, since it's 0 based), on ASIC 0 / SLICE 0. As we know from the ELAM run originally on the LC, we triggered on this slice:

```
module-2# debug platform internal tah elam asic 0
module-2(DBG-TAH-elam)# trigger reset
module-2(DBG-TAH-elam)# trigger init in-select 13 out-select 0
module-2(DBG-TAH-elam-insel13)# set inner ipv4 src_ip 10.100.17.11 dst_ip 192.168.20.1
module-2(DBG-TAH-elam-insel13)# start
stat
module-2(DBG-TAH-elam-insel13)# stat
ELAM STATUS
===========
Asic 0 Slice 0 Status Armed
Asic 0 Slice 1 Status Armed

module-2(DBG-TAH-elam-insel13)# stat
ELAM STATUS
===========
Asic 0 Slice 0 Status Triggered <---- Packet triggered from FM Asic 0 Slice 1 Status Triggered
<---- Packet triggered from Front Panel
```

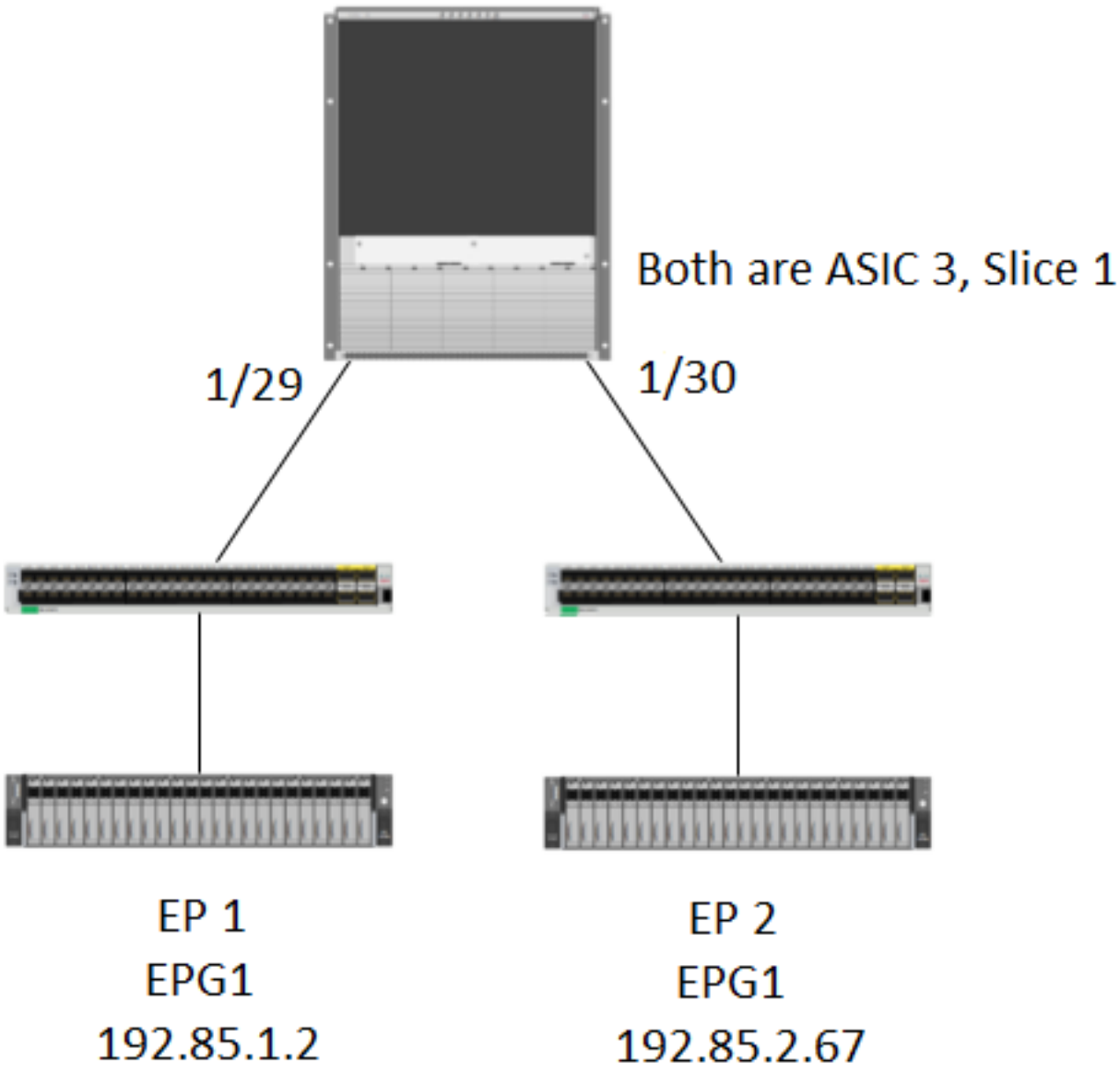
The ovector on this ELAM is sug_elam_out_sidebnd_no_spare_vec.ovector_idx: **0x98,** which we know from the "hal l2 port gpd", maps to the correct interface on the LC:

```
================================================================================================
==========================================
                      Uc   Uc                          |       Reprogram       |
| Rep |
                    I PC   Pc                        L |   R I R D    R  U U X | L Xla Ovx N
NI Vif    RwV    Ing  Egr  | V R | PROF H
IfId     Ifname      P Cfg  MbrID As AP Sl Sp Ss Ovec S | P P P S P Sp Sp C M L | 3 Idx Idx L3
L3 Tid    Tid    Lbl  Lbl  | S V | ID   I
================================================================================================
==========================================
1f5      SpInBndMgmt 0 9de  1a    0  0  0  0  0  0    0   0 0 0 0 0 0  0  0 0 0   0 0   0   0  0
D-2d4  D-3e1  0    0      0 0   1    0
1a080000 Eth2/1      0 9a   1c    0  11 0  10 20 20   1   0 0 0 0 0 0  0  0 0 0   1 b   b   1  1
D-f3   D-61   100  0      0 0   1    0
1a081000 Eth2/2      0 9b   22    0  d  0  c  18 18   1   0 0 0 0 0 0  0  0 0 0   1 c   c   1  1
D-1ee  D-30b  100  0      0 0   1    0
1a084000 Eth2/5      0 9e   1e    0  3d 1  14 28 a8   1   0 0 0 0 0 0  0  0 0 0   1 1   1   1  1
```

```
D-19a  D-2ee  100  0      0 0   1    0
1a085000 Eth2/6      0 9f   24    0  39 1  10 20 a0   1   0 0 0 0 0 0  0  0 0 0   1 e   e   1  1
D-87   D-184  100  0      0 0   1    0
1a086000 Eth2/7 0 a0 26 0 35 1 c 18 98 1 0 0 0 0 0 0 0 0 0 0 1 d d 1 1 D-1d0 D-357 100 0 0 0 1 0
<--- Interface that connects to Leaf 5 1a088000 Eth2/9 0 a2 20 1 d 0 c 18 18 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 D-3ea D-1a9 100 0 0 0 1 0
```

Extra Scenario: Getting an Ovector that is not in the "hal internal-port pi" output

Topology



Logic

There are some scenarios where we catch a packet that does not have an Ovector in the "**show platform internal hal l2 internal-port pi**" table. In the scenario below, we are actually catching the packet coming back from the FM, so we need to look at a different table to see which front panel port the packet is selecting.

Note that the topology above is a completely different environment where the transit traffic is learned (no proxy routing). The module is a N9K-X9732C-EX.

```
@module-1# debug platform internal tah elam asic 3
@module-1(DBG-elam)# trigger reset
@module-1(DBG-elam)# trigg init in-select 13 out-select 0
@module-1(DBG-elam-insel13)# set inner ipv4 src_ip 192.85.1.2 dst_ip 192.85.2.67
@module-1(DBG-elam-insel13)# star
@module-1(DBG-elam-insel13)# stat
ELAM STATUS
===========
Asic 3 Slice 0 Status Armed
Asic 3 Slice 1 Status Triggered

@module-1(DBG-elam-insel13)# report | grep ovector
 sug_elam_out_sidebnd_no_spare_vec.ovector_idx: 0xA0 <<<<<<<<<<<<<<<< now we look for this in
the "hal internal-port pi" command


@module-1# show platform internal hal l2 internal-port pi
No sandboxes exist
Num. of Sandboxes: 1
Legend:
-------
IfId:      Interface Id               IfName:     Interface Name
As:        Asic                       AP:         Asic Port
Sl:        Slice                      SP:         Slice Port
Ss:        Slice SrcId                Ovec:       Ovector
UcPcCfgId: Uc Pc CfgId                Lb Mbrid:   LB MbrId

Sandbox_ID: 0, BMP: 0x0
 Internal Port Count: 24

=====================================================================
                                                      UcPc  Lb
IfId     IfName                       As AP Sl SP Ss Ovec CfgId MbrId
=====================================================================
7d       -                            0  21 0  20 38 38   0     4
7e       -                            0  29 1  0  0  80   0     8
7f       -                            1  21 0  20 38 38   0     c
80       -                            1  29 1  0  0  80   0     10
81       -                            2  21 0  20 38 38   0     14
82       -                            2  29 1  0  0  80   0     18
83       -                            3  21 0  20 38 38   0     1c
84       -                            3  29 1  0  0  80   0     20
ad       -                            0  25 0  24 40 40   0     1
ae       -                            0  41 1  18 30 b0   0     6
af       -                            1  25 0  24 40 40   0     9
b0       -                            1  41 1  18 30 b0   0     e
b1       -                            2  25 0  24 40 40   0     11
b2       -                            2  41 1  18 30 b0   0     16
b3       -                            3  25 0  24 40 40   0     19
b4       -                            3  41 1  18 30 b0   0     1e
dd       -                            0  15 0  14 28 28   0     2
de       -                            0  4d 1  24 40 c0   0     5
df       -                            1  15 0  14 28 28   0     a
e0       -                            1  4d 1  24 40 c0   0     d
e1       -                            2  15 0  14 28 28   0     12
e2       -                            2  4d 1  24 40 c0   0     15
e3       -                            3  15 0  14 28 28   0     1a
e4       -                            3  4d 1  24 40 c0   0     1d  <<<<<<<  we cant find an
entry that matches 0xA0
```

```
@module-1# show platform internal hal l2 port gpd
Legend:
-------
<snip>

Sandbox_ID: 0, BMP: 0x0
 Port Count: 6

================================================================================================================
==========================================
                      Uc   Uc                            |       Reprogram
|                                          | Rep |
                    I PC   Pc                         L |   R I R D    R  U U X | L Xla Ovx N  NI
Vif    RwV    Ing  Egr  | V R | PROF H
IfId     Ifname      P Cfg  MbrID As AP Sl Sp Ss Ovec S | P P P S P Sp Sp C M L | 3 Idx Idx L3
L3 Tid    Tid    Lbl  Lbl  | S V | ID   I
================================================================================================================
==========================================
1f5      SpInBndMgmt 0 9de  1a    0  0  0  0  0  0    0   0 0 0 0 0 0  0  0 0 0   0 0   0   0
0  D-2d4  D-3e1  0    0      0 0   1    0
1a000000 Eth1/1      0 1b   1c    0  11 0  10 20 20   1   0 0 0 0 0 0  0  0 0 0   1 1   1   1
1  D-13b  D-33b  500  0      1 0   3    0
1a01c000 Eth1/29     0 37   1e    3  3d 1  14 28 a8   1   0 0 0 0 0 0  0  0 0 0   1 8   8   1
1  D-3f2  D-7a   100  0      0 0   2    0
1a01d000 Eth1/30     0 38   20    3  39 1  10 20 a0   1   0 0 0 0 0 0  0  0 0 0   1 5   5   1
1  D-36e  D-362  100  0      0 0   2    0  <<<<<<< 1/30 is the phys interface that connects to
leaf 102, verified by topology, ASIC 3, Slice 1
1a01e000 Eth1/31     0 39   22    3  35 1  c  18 98   1   0 0 0 0 0 0  0  0 0 0   1 9   9   1
1  D-273  D-8    100  0      0 0   2    0
1a01f000 Eth1/32     0 3a   24    3  31 1  8  10 90   1   0 0 0 0 0 0  0  0 0 0   1 a   a   1
1  D-154  D-5d   100  0      0 0   2    0
```