

Hardware EX: Mergulho profundo do encaminhamento de pacote ACI.

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Encenações](#)

[2 EP na mesma folha EPG/Same - Quadro comutado](#)

[Topologia](#)

[ELAM](#)

[2 EP na folha diferente EPG/Same - Pacote roteado](#)

[Topologia](#)

[ELAM](#)

[2 EP na folha diferente EPG/Different - Pacote roteado](#)

[Topologia](#)

[ELAM](#)

[1 EP --> L3 para fora - Fluxo roteado](#)

[Topologia](#)

[ELAM](#)

[1 EP --> EP ou SVI remoto - Verificação da espinha](#)

[Topologia](#)

[Lógica](#)

[IP sintético](#)

[Módulo de construção ELAM](#)

[Encenação extra: Obtendo um Ovector que não esteja “na saída do pi hal da porta interna”](#)

[Topologia](#)

[Lógica](#)

Introdução

Este documento descreve encenações diferentes da transmissão usando o hardware ASIC da nova geração na infraestrutura céntrica do aplicativo (ACI). Mostrará como verificar que o hardware está programado corretamente e nós estamos enviando pacotes aos valores-limite do destino correto (EP) nos grupos apropriados do valor-limite (EPGs).

Pré-requisitos

Requisitos

Não existem requisitos específicos para este documento.

Componentes Utilizados

As informações neste documento são baseadas nas seguintes versões de hardware e software:

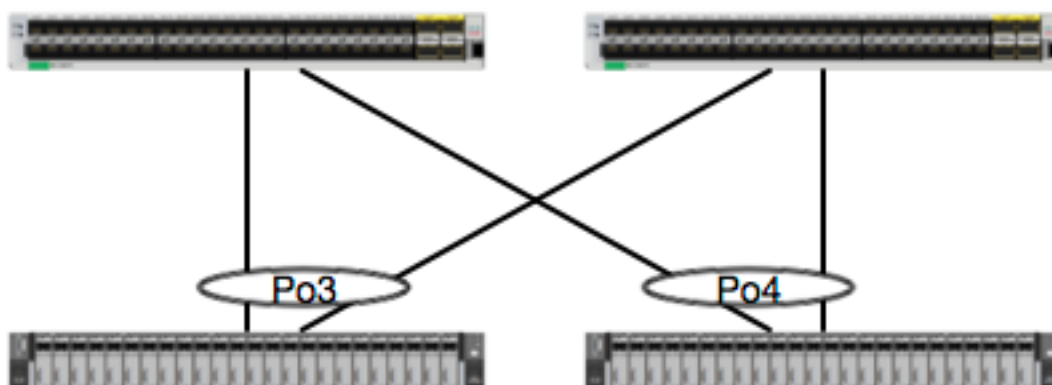
- Uma tela ACI que consista em dois Switches da espinha e em dois Switches da folha usando o hardware EX
- Um host de ESXi com dois uplinks que vão a cada um da folha comuta
- Dispositivo do nexa 5000 que atua como um roteador.
- Um controlador da infraestrutura da política do aplicativo (APIC) que seja usado para a instalação inicial

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

Encenações

2 EP na mesma folha EPG/Same - Quadro comutado

Topologia



EP1
EPG1
0050.56a5.fccc
192.168.20.2/24

EP2
EPG1
0050.56a5.6794
192.168.20.3/24

Dado esta topologia, o fluxo de EP1 a EP2 é um fluxo L2 e deve ser ligado localmente o que folha o tráfego de origem vem dentro. A primeira coisa a verificar com os fluxos da camada 2 (L2) é a tabela de endereços MAC para determinar se e onde os frames recebidos do interruptor:

```
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
```

A fim ver o encapsulamento vlan, nós podemos verificar o base de dados EP também:

```
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
```

Nós conhecemos-nos os fósforos FD_VLAN 30, mas podemos sempre validar o mapeamento no software:

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

E naturalmente, nós podemos verificar o hardware para certificar-se de mapas do VLAN 30 a VLAN 2268 como o encapsulamento do painel dianteiro.

```
leaf4# vsh lc
module-1# show system internal eltc 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 access_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>
```

Dado que os EP estão aprendidos no software, nós podemos igualmente validar que o hardware programou a informação L2 destes EP também. No hardware novo, há a camada de abstração de hardware (HAL) que é o estado do software do hardware. O trabalho de HAL é tomar pedidos de programação do software e empurrá-los para o hardware.

A fim ver a informação de hardware L2 sobre um valor-limite, nós podemos olhar a tabela L2 em HAL para endereços dados do Mac:

```
leaf4# vsh lc
module-1# show platform internal hal ep 12 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 FDIId: 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 FDIId 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 1 0 0 module-1# show platform internal hal ep 12 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 FDIId 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 1 0 0
```

Agora que nós traçamos para fora o hardware, nos deixamos fazer um ELAM e ver onde o pacote deve ir.

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
12 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
```

Grande, assim que Leaf4 recebeu o quadro em Asic 0 fatias 1. Com o ELAM no hardware novo, há um campo novo que seja muito importante ao pesquisar defeitos: **ovector_idx**. Este deslocamento predeterminado é o deslocamento predeterminado da porta física que o quadro/pacote deve ser enviado fora de. Uma vez que você tem o **ovector_idx**, nós podemos usar este comando encontrar que porta traça:

```
module-1(DBG-TAH-elam-insel6)# show platform internal hal 12 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: UCPCfg 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 - - 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 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 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
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 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
```

O interruptor pensa que o pacote deve ser enviado fora dos Ethernet de interface 1/32. É esse PO4 onde nós aprendemos que address do Mac?

```
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
```

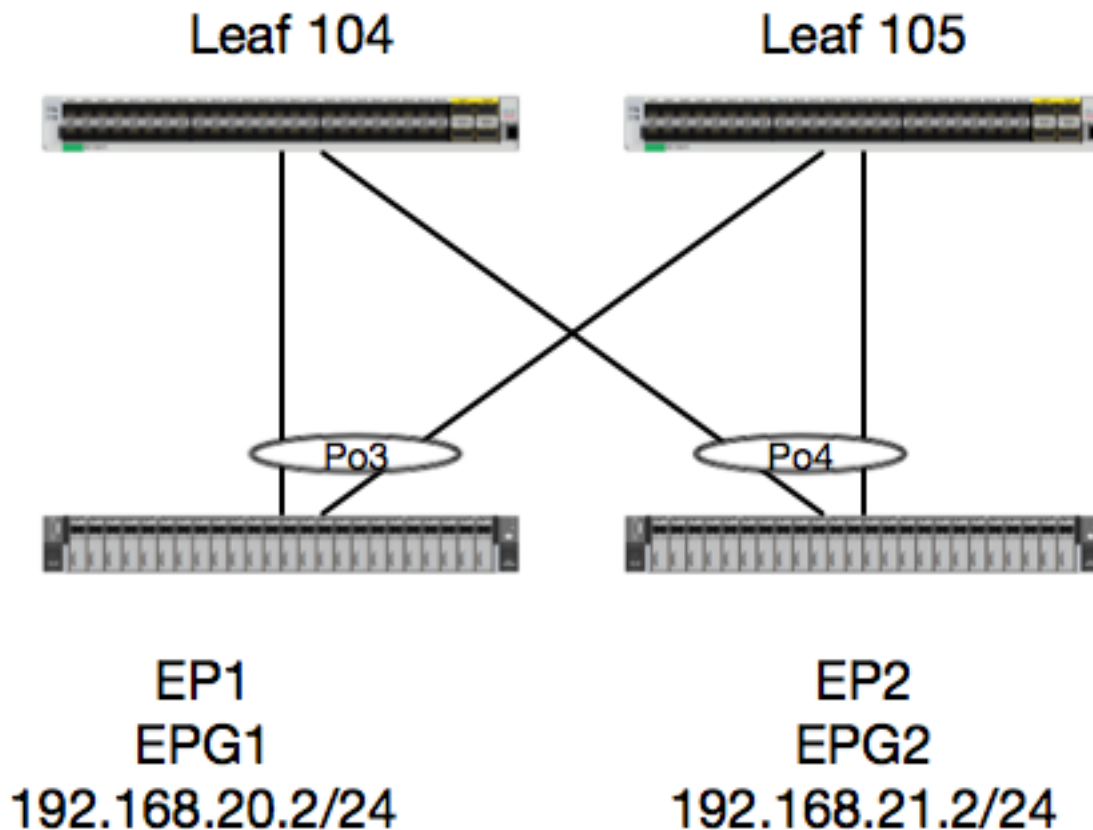
```
-----
Group Port-      Type      Protocol  Member Ports
Channel
```

```
-----
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)
```

Sim, assim que o pacote Br enviado fora da relação 1/32 ao host de destino.

2 EP na folha diferente EPG/Same - Pacote roteado

Topologia



Neste exemplo, nós seguiremos o fluxo de pacote de informação de um pacote de EP1 a EP2 onde existem nos mesmos pares da folha do vPC. Os dois EP estão nos BD diferentes de utilização de EPG diferente.

A primeira coisa a fazer sempre é verificar o base de dados EP para considerar se nós aprendemos os EP:

```
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
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Encap MAC Address MAC Info/ Interface Domain VLAN IP Address IP Info +-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
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
```

Desde que nós aprendemos os EP e conhecemos a informação IP, nós devemos poder olhar a informação de aprendizagem EP no hardware:

```
leaf4# vsh_lc module-1# show platform internal hal ep 13 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 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*ternal2 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

```

A tabela HAL Layer3 (I3) é muito uself desde que nos dá a informação VLAN/Port para EP aprendidos I3. Nós sabemos que o destino existe de um Po4, assim que o pacote deve ser enviado fora de toda a porta em Po4.

Deixe-nos executar um ELAM e ver o que nós obtemos!

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

```

Grande, assim que nós provocamos o pacote, e nós encontrou que o “ovector_idx” é 0x9E. O deslocamento predeterminado do ovector é o deslocamento predeterminado phycial que parte da relação que o pacote deve ser enviado fora de. Deixe-nos ver que porta tem esse deslocamento predeterminado:

```

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 - - 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
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 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 - -

```



```

0050.56a5.fc5c 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 +-----+
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Address MAC Info/ Interface Domain VLAN IP Address IP Info +-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
192.168.1.100 tunnel2

```

Agora, deixe-nos verificam o que o hardware programou:

```

leaf4# vsh_lc
module-1# show platform internal hal ep 13 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 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*ternal2 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

```

O hardware pensa que o EP existe no túnel 2. Que é o destino para o túnel 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

```

Desde que o destino existe fora de um vPC, esse IP de destino deve ser o IP virtual do vPC das folhas remotas. Deixe-nos verificar em uma folha remota e ver:

```

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

```

Aperfeiçoe, assim que aprendeu o destino EP dos pares remotos do vPC. Deixe-nos ver que ELAM vê e verifique-os que nós estamos enviando o pacote correctly:

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

```

Agora, com destinos remotos no hardware EX, há 2 valores ELAM que são muito importantes ao pesquisar defeitos o fluxo de pacote de informação. O ovector_idx gosta antes, e o 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

```

No hardware EX, nós temos a capacidade para conduzir a porta do destino que o pacote deve ser enviado fora de. Antes, nós geralmente apenas verificamos o idx do encap e verificamos que o idx do destino era o túnel correto. Aqui nós podemos verificar que mapas de portas a 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: UCPCfg 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 - - 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 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 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

```

O interruptor pensa que deve o enviar à espinha na relação Eth1/49. Mas como podemos nós verificar o encap estamos corretos?

Nós precisamos primeiramente de olhar a informação de hardware sobre o túnel. Nós podemos fazer este executando este comando HAL:

```

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 Tunnel12 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

```

Esta saída dá-nos alguns valores que nós nos importamos aproximadamente:

lflid - A relação ID atribuída ao túnel

IP - O IP do destino. Isto deve combinar ELTMC.

L3 lflid - As relações que da camada 3 o interruptor pode se usar para enviar ao destino apropriado.

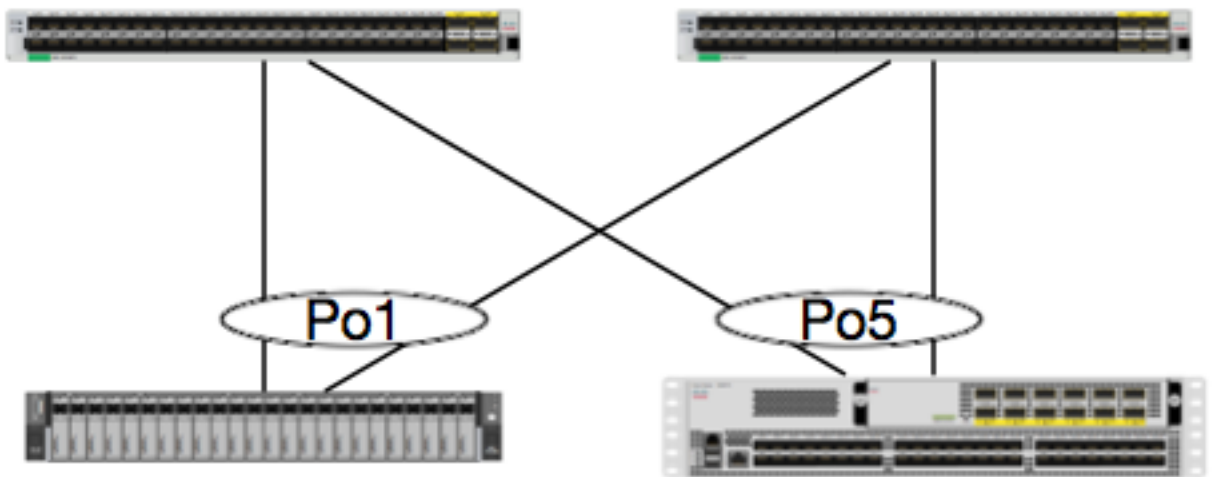
Uma vez que nós conhecemos o lflid, nós podemos verificar que o encap que nós obtivemos no elam combina o destino de túnel:

```
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
```

Este túnel tem um RwEncapIdx (deslocamento predeterminado do encap da reescrita) de 6, que é o que foi indicado no elam.

1 EP --> L3 para fora - Fluxo roteado

Topologia



EP1
EPG1
0050.56a5.50ab
192.168.20.10/24

N5K -OSPF
100.100.100.100/32

Neste exemplo, nós seguiremos o fluxo de pacote de informação de um pacote de EP1 que envia o ICMP a um laço de retorno em um N5K que executa o OSPF. N5K é conectado através de um L3Out nos mesmos pares de Switches EX.

Desde que nós verificamos o EP local que programa no início deste documento, nos deixamos supor que o EP está aprendido corretamente no hardware e continue sobre à verificação da rota.

Primeiramente, deixe-nos verificar o estado OSPF e a tabela de roteamento:

```
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
```

Assim nós sabemos que a tabela do routing mostra o salto seguinte como o 5K em 10.10.27.3. Bom começo, mas como podemos nós verificar o que o hardware tem?

Deixe-nos primeira verificação a tabela de adjacência no hardware certificar-se que nós temos o ARP resolvidos a 10.10.27.3, e isso está programado com a relação correta:

```
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
```

O MAC endereça os fósforos que no 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
```

Em Plataformas EX, há um “hw_vrf_idx” que seja atribuído a um VRF. Este deslocamento predeterminado será provido quando nós verificamos a programação do hardware. Deixe-nos encontrar o deslocamento predeterminado:

```
module-1# show system internal eltc info vrf jr:sb VRF-TABLE: jr:sb vrf_type: tenant :::
context_id: 6 overlay_index: 0 ::: vniid: 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 vniid_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_vniid: 2129921 ::: is_infra: 0 tornbinfracwbd: 0 :::
torsbinfracwbd: 0 ingressBdAclLabel: 0 ::: ingBdAclLlblMask: 0 egressBdAclLabel: 0 :::
egrBdAclLlblMask: 0 sg_label: 5 ::: sclass: 16386 sp_incomplete: 1 ::: sclassprio: 3 [SDB INFO]:
v4 table vrf type: 1 vrf id: 5 vniid: 2129921 internal infra vlan: 0 external router
mac:00:22:bd:f8:19:ff v6 table vrf type: 1 vrf id: 5 vniid: 2129921 internal infra vlan: 0
external router mac:00:22:bd:f8:19:ff :::
```

Depois que nós detectamos a adjacência, HAL deve programar uma rota. Nós podemos verificar este que usa o comando seguinte:

```
module-1# show platform internal hal 13 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 13 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 | -----

Trie -----><Dleft Trie>												VRF	Prefix/Len	RT	RID	LID	Type	PID	FPID/	HIT	NB-ID	NB Hw	PID	FPID/	TBI	TRO	Ifindex	CLSS	CLP	AI	SH	DH	Flags		TID	IDX	B	Idx		TID
-----><Dleft												-----><Dleft			-----><Dleft			-----><Dleft			-----><Dleft			-----><Dleft			-----><Dleft													
NB-ID	NB Hw																																							
-----><Dleft												-----><Dleft			-----><Dleft			-----><Dleft			-----><Dleft			-----><Dleft																

```

----- |Sandbox_ID: 0 Asic Bitmap: 0x0 -----
----- module-1# show platform internal hal 13 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

```

Esta saída dá-nos a informação em relação à rota do salto seguinte. 4612 são o hw_vrf_idx do júnior: sb VRF. Para que nós a verificação o salto seguinte, "o HW Idx NB" no TCAM será usada contra a tabela seguinte:

```

module-1# show platform internal hal 13 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 13
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 1 1 1
1214 8c:60:4f:02:88:fc 0 0 2c0d 0 0 0 0 0 0 10.10.27.3

```

Aqui, nós tomamos "o HW Idx NB" e traçamo-lo à "BATIDA IDX". Isto mostra-nos a entrada que corresponde ao salto seguinte MAC/IP. Este é o equivalente de olhar a mostra do defip "I3" e a mostra da saída "I3" em Broadcom no Switches da folha da geração 1 ACI.

Como nós podemos ver, a tabela tem a informação correta:

L2 INTF: 0x16000004 ---> o ifIndex do canal de porta 5

BATIDA IDX: O deslocamento predeterminado conduzido do HW Idx N.B. nas rotas I3 hal

MAC: 8c:60:4f:02:88:fc --> MAC do salto seguinte SVI em 5K

EPG: SCLASS de L3 EPG

Endereço IP: 10.10.27.3 ---> IP do salto seguinte do SVI em 5K

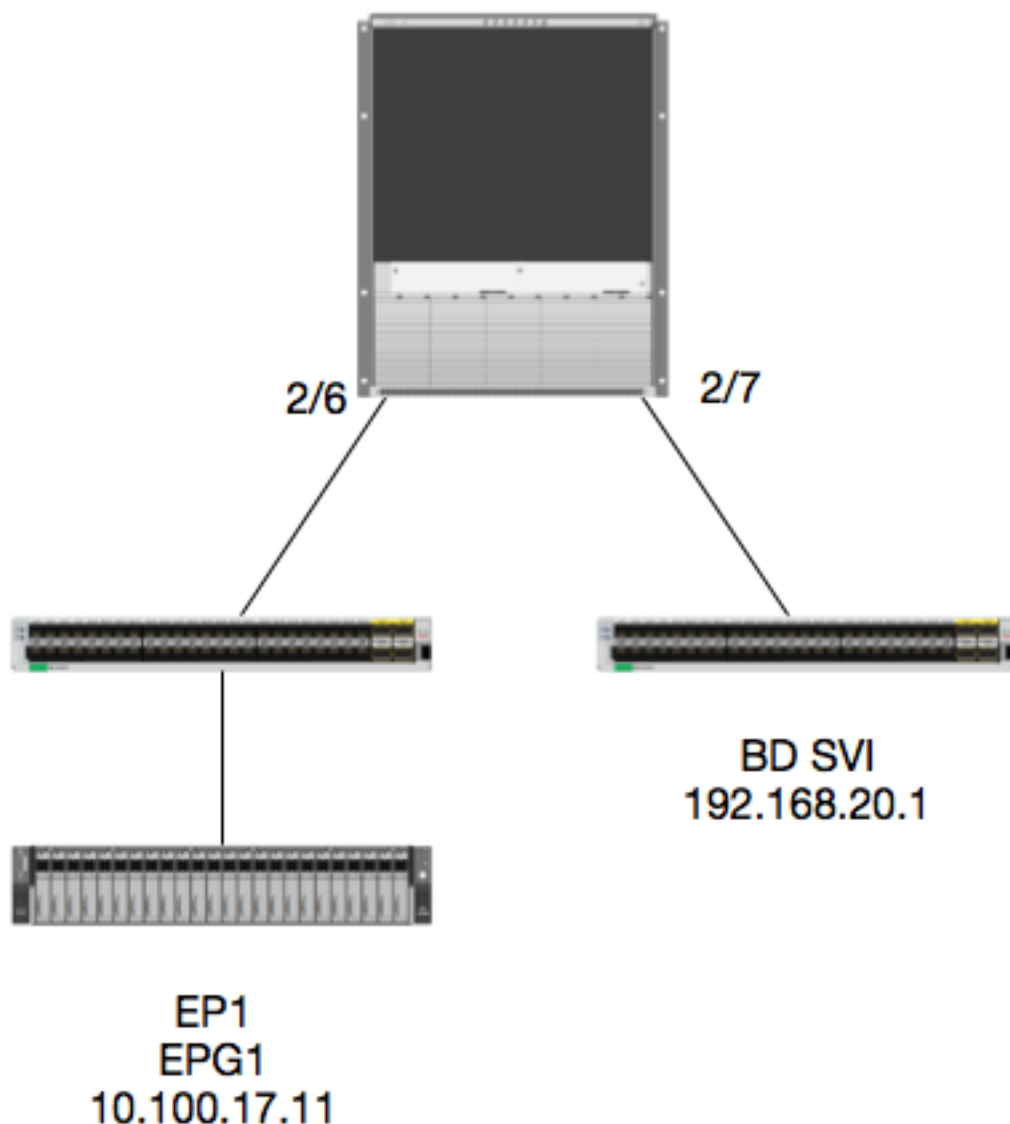
ELAM

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

```
leaf6# cat elam_report.txt | grep ip.da
  sug_pr_lu_vec_l3v.ip.da: 0x000000000000000064646464 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 --> EP ou SVI remoto - Verificação da espinha

Topologia



Lógica

Neste exemplo, nós seguiremos o fluxo de pacote de informação de um pacote de EP1 destinado a um Switched Virtual Interface remoto BD (SVI). A finalidade deste exemplo será verificar que transmissão da espinha para assegurar o pacote está enviada à folha correta. Deixe-nos supor o pacote foi enviado ao proxy da espinha na folha do ingresso.

Na espinha, deixe-nos verificarmos primeiramente o Conselho do protocolo dos oráculos (CAPOEIRA) para o IP de destino desde que o pacote é enviado ao proxy da espinha para uma consulta:

```
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
```

Deixe-nos verificar que folha tem esse endereço TEP:

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

Desde que nós sabemos que o pacote está entrando a espinha no módulo 2, mova 6, nós podemos anexar ao módulo 2 e olhar a disposição da porta.

```

spinel# 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-spinel# 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 12 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 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 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 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 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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 0 0 0 0 0 0 0 0 0 0
0 0 1 0

```

Agora nós sabemos que ASIC para executar nosso ELAM. 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

```

Olhando o ELAM, nós podemos encontrar o deslocamento predeterminado do ovector:

Front Panel ELAM drove sug_elam_out_sidebnd_no_spare_vec.ovector_idx: 0xB8

Agora, como nós traçamos 0xb8 a uma porta? Desde que nós sabemos o pacote deve obter enviado a um módulo de construção (FM) para uma consulta, nós pode olhar o mapeamento da porta interna para encontrar o dest FM:

```

module-2# show platform internal hal 12 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

```

Este Mbrld é a relação em USD esse mapas a uma relação em FM. Nós podemos encontrar que FM olhando USD conecta e inspecionando a porta 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
```

O “entalhe” é 0 baseado, e a numeração de FM é 1 baseado, assim que nós precisamos de adicionar 1 ao número alistado aqui. Isto significa que o pacote deve ser enviado a FM 23.

IP sintético

Apenas como em alpino, há um IP sintético usado como o endereço IP de Um ou Mais Servidores Cisco ICM NT exterior para determinar a mistura para a consulta da CAPOEIRA. A fim encontrar isto, você precisa de executar estes comando e grep para o DST IP interno:

```
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
```

Isto mostra-nos que 1.203.211.185 é nosso IP sintético. Baseado nisto, nós podemos igualmente ajustar “o DST IP exterior” em nosso elam de FM para ser este. Nós devemos provocar em FM:

Módulo de construção 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
```

Obviamente, despeje os relatórios completos, mas deixe-nos olhar o ovector_idx para este pacote que nós provocamos:

```
lac_elam_out_sidebnd_no_spare_vec.ovector_idx: 0x20 <----- Deslocamento predeterminado de
Ovector usado no comando abaixo
```

Como nós figuramos para fora que a relação tem esse ovector?. Em FM, execute isto:

```
module-23(DBG-TAH-elam-insel13)# show platform internal hal 12 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: UCPCfg 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 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 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 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
```

Esse ovector traça a LC1 (placa de linha no entalhe 2, desde que é 0 baseado), em ASIC 0/FATIA 0. Como nós sabemos do ELAM executado originalmente no LC, nós provocamos nesta fatia:

```
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
```

O ovector neste ELAM é sug_elam_out_sidebnd_no_spare_vec.ovector_idx: 0x98, que nós conhecemos “do gpd hal da porta l2”, traça à relação correta no LC:

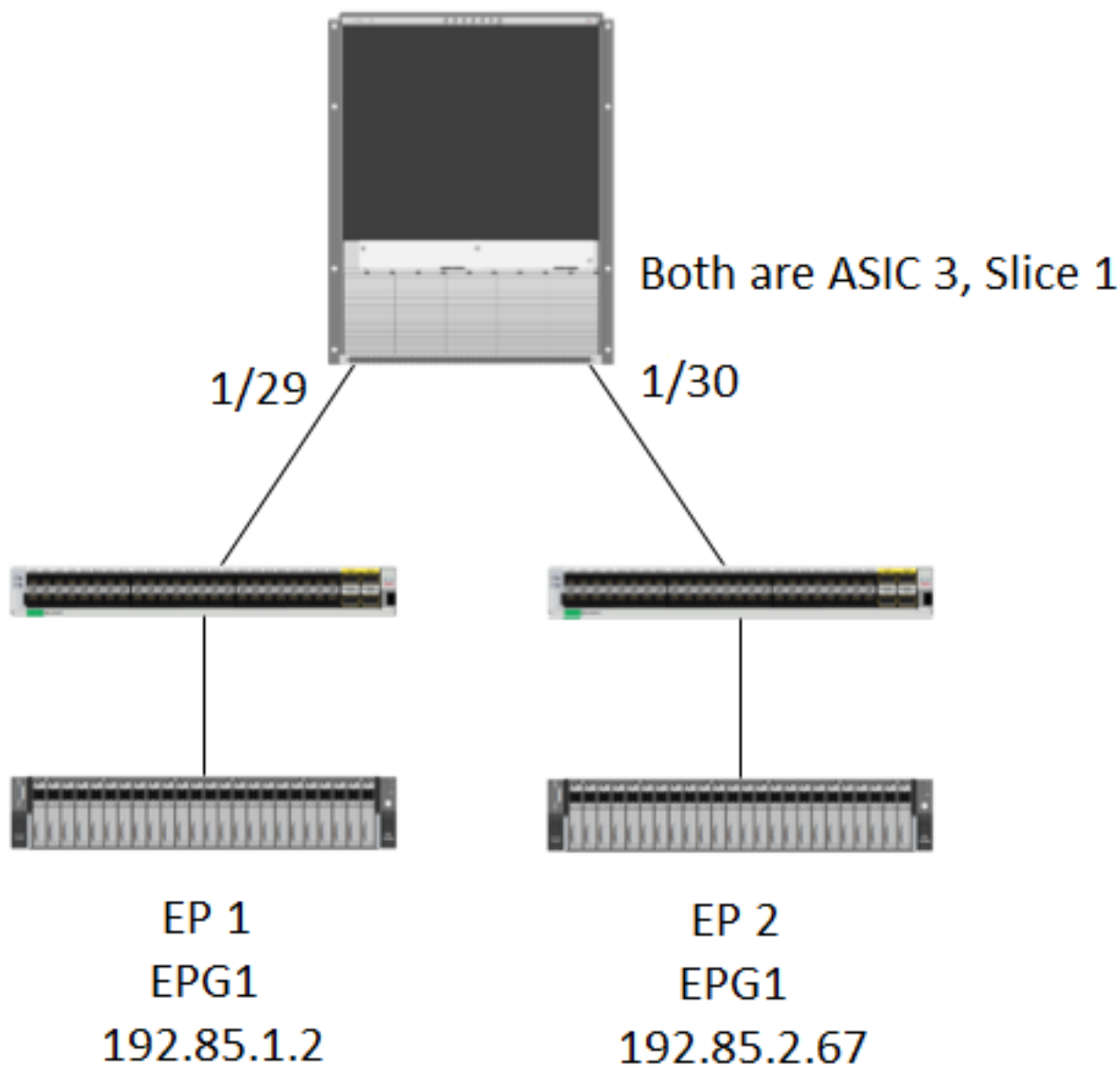
```
=====
```

=====																														
Rep	Uc							Reprogram																						
NI Vif	RwV	Ing	Egr	V	R	PROF	H	L		R	I	R	D	R	U	U	X	L	Xla	Ovx	N									
IfId	Ifname	P	Cfg	MbrID	As	AP	S1	Sp	Ss	Ovec	S	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					
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	0	0	0	0					
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	0	0	0	0					
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	0	0	0	0					
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	0	0	0	0					
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	0	0	0	0					
<--- Interface that connects to Leaf 5 1a088000 Eth2/9																														
0	0	0	0	0	0	0	0	0	0	0	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																							

```
=====
```

Encenação extra: Obtendo um Ovector que não esteja “na saída do pi hal da porta interna”

Topologia



Lógica

Há algumas encenações onde nós travamos um pacote que não tenha um Ovector da “do pi hal interno da porta interna I2 da plataforma mostra na tabela”. Na encenação abaixo, nós estamos travando realmente o pacote que volta de FM, assim que nós precisamos de olhar uma tabela diferente para ver que painel dianteiro mover o pacote está selecionando.

Note que a topologia acima é um ambiente completamente diferente onde o tráfego de trânsito seja instruído (nenhum proxy que distribui). O módulo é um 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
```