

Matériel EX : Plongée en eau profonde de transfert de paquet interception commandée en vol.

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Scénarios](#)

[2 PE dans la même feuille EPG/Same - Trame commutée](#)

[Topologie](#)

[ELAM](#)

[2 PE dans la feuille différente EPG/Same - Paquet routé](#)

[Topologie](#)

[ELAM](#)

[2 PE dans la feuille différente EPG/Different - Paquet routé](#)

[Topologie](#)

[ELAM](#)

[1 PE --> L3 - Écoulement conduit](#)

[Topologie](#)

[ELAM](#)

[1 PE --> PE ou SVI de distant - Vérification d'épine](#)

[Topologie](#)

[Logique](#)

[IP synthétique](#)

[Module ELAM de matrice](#)

[Scénario supplémentaire : Obtenant un Ovector qui n'est pas dans « la sortie du l'interne-port pi hal »](#)

[Topologie](#)

[Logique](#)

Introduction

Ce document décrit différents scénarios d'expédition utilisant le matériel de la nouvelle génération ASIC en infrastructure centrale d'application (interception commandée en vol). Il affichera comment vérifier le matériel est programmé correctement et nous sommes des transferts des paquets aux points finaux de destination appropriée (PE) dans les groupes appropriés de point final (EPGs).

Conditions préalables

Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

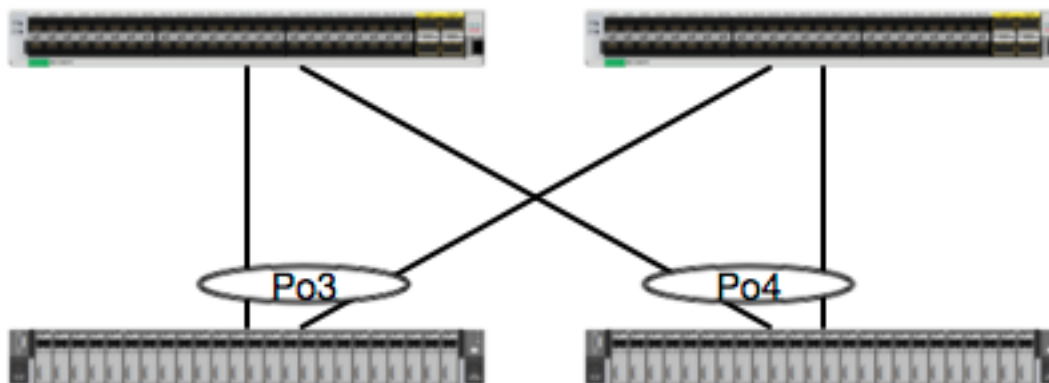
- Une matrice interception commandée en vol qui se compose de deux Commutateurs d'épine et de deux Commutateurs de feuille utilisant le matériel EX
- Un hôte d'ESXi avec deux liaisons ascendantes qui vont à chacune de la feuille commute
- Périphérique de Nexus 5000 agissant en tant que routeur.
- Un contrôleur d'infrastructure de stratégie d'application (APIC) qui est utilisé pour la première installation

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

Scénarios

2 PE dans la même feuille EPG/Same - Trame commutée

Topologie



EP1
EPG1
0050.56a5.fccc
192.168.20.2/24

EP2
EPG1
0050.56a5.6794
192.168.20.3/24

Donné cette topologie, l'écoulement d'EP1 à EP2 est un L2 circulent et devraient être commutés localement sur sur Qu'est ce que feuille le trafic source entre. La première chose à vérifier avec des écoulements de la couche 2 (L2) est l'adresse-table de MAC pour déterminer si et où les


```

=====
===== 1c BD-28 P1 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

```

Maintenant que nous avons tracé le matériel, nous avons permis de faire un ELAM et de voir où le paquet devrait disparaître.

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

```

Grand, ainsi Leaf4 a reçu la trame sur ASIC 0 parts 1. Avec ELAM sur le nouveau matériel, il y a un nouveau champ qui est pour le dépannage très important : **ovector_idx**. Cet index est l'index de port physique que la trame/paquet devrait être expédiée hors de. Une fois que vous avez l'ovector_idx, nous pouvons utiliser cette commande de trouver à quel port il trace :

```

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 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

```

Le commutateur pense que le paquet devrait être expédié hors des Ethernets 1/32 d'interface. Est-il ce PO4 où nous avons appris que des address de 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-Channel	Type	Protocol	Member Ports
1	Po1(SU)	Eth	LACP	Eth1/5(P)
2	Po2(SU)	Eth	LACP	Eth1/6(P)
3	Po3(SU)	Eth	LACP	Eth1/31(P)


```

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 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

```

La table HAL Layer3 (I3) est très uself puisqu'elle nous fournit les informations VLAN/Port pour les PE appris par I3. Nous savons que la destination existe d'un Po4, ainsi le paquet devrait être expédié hors de n'importe quel port dans Po4.

Exécutons un ELAM et voyons ce que nous obtenons !

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

```

Grand, ainsi nous avons déclenché le paquet, et nous fondrez que le « ovector_idx » est 0x9E. L'index d'ovector est l'index phycial sortant d'interface que le paquet devrait être expédié hors de. Voyons quel port a cet 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 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 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 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 0 0 0 1 5 3 2 2 D-350 - 400 0 0 0 1 0

```

Est-ce que nous ressemble à-lui lui envoyer lui le port 1/32, cela devrait est 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

```

```

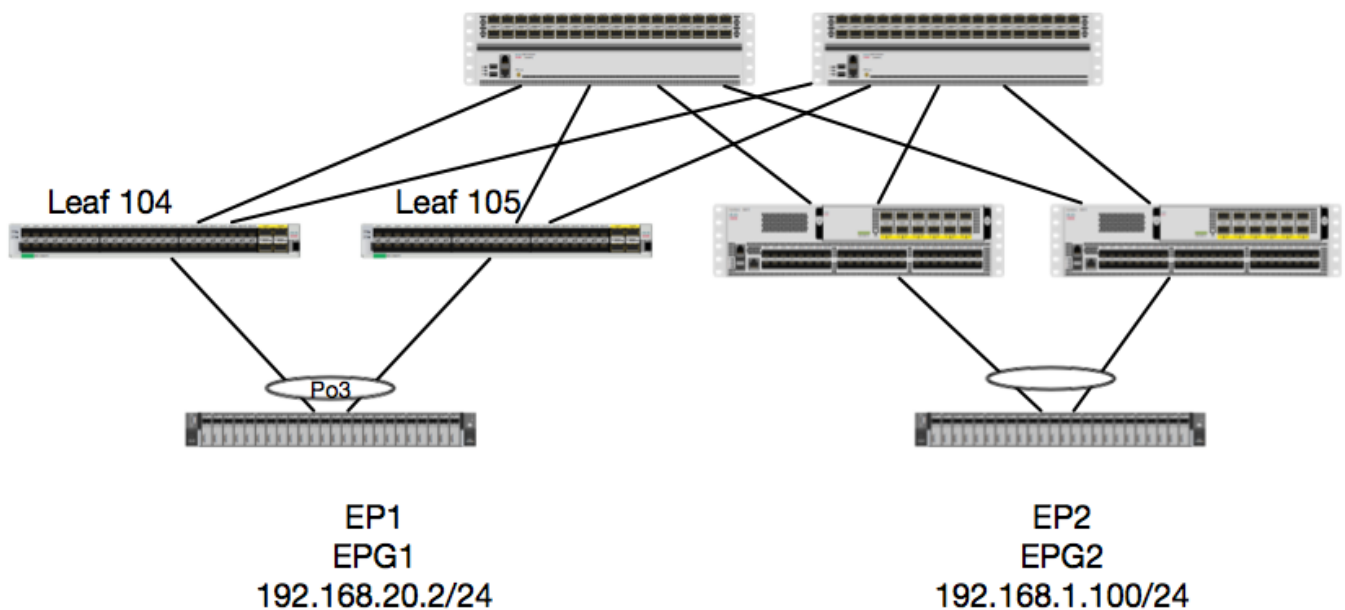
-----
Group Port-      Type      Protocol  Member Ports
Channel
-----
1      Po1(SU)     Eth       LACP      Eth1/5(P)
2      Po2(SU)     Eth       LACP      Eth1/6(P)
3      Po3(SU)     Eth       LACP      Eth1/31(P)
4 Po4(SU) Eth LACP Eth1/32(P)

```

Oui, c'est correct.

2 PE dans la feuille différente EPG/Different - Paquet routé

Topologie



Dans cet exemple, nous dépisterons l'écoulement de paquet d'un paquet d'EP1 à EP2 où EP1 existe sur une paire EX de vpc et EP2 existe sur une paire distante de feuille de vpc de la génération 1. Les deux PE sont dans différents EPG utilisant les différents BD.

De nouveau, vérifions où les PE sont appris :

voit et vérifiez-nous expédient le paquet 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
```

Maintenant, avec les destinations distantes sur le matériel EX, il y a 2 valeurs ELAM qui sont écoulement très important de paquet de pour le dépannage. L'ovector_idx aiment avant, et l'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
```

Sur le matériel EX, nous avons la capacité de piloter la destination port que le paquet devrait être expédié hors de. Avant, nous avons habituellement juste vérifié l'idx d'encap et l'avons vérifié que l'idx de destination était le tunnel correct. Ici nous pouvons vérifier quel port trace à 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 - - 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 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 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 0 1 5 3 2 2 D-350 - 400 0 0 0 1 0
```

Le commutateur pense qu'il devrait l'expédier à l'épine sur l'interface Eth1/49. Mais comment pouvons-nous vérifier l'encap est-nous correct ?

Nous le premier besoin de regarder des informations de matériel sur le tunnel. Nous pouvons faire ceci en exécutant cet ordre 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 O B d 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 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

```

Cette sortie nous donne quelques valeurs que nous nous inquiétons environ :

lflid - L'ID d'interface alloué au tunnel

IP - L'IP de la destination. Ceci devrait apparier ELTMC.

L3 lflid - Les interfaces de la couche 3 que le commutateur peut employer pour expédier à la destination appropriée.

Une fois que nous connaissons l'**lflid**, nous pouvons vérifier que l'encap que nous avons obtenu dans l'**elam** apparie la destination de tunnel :

```

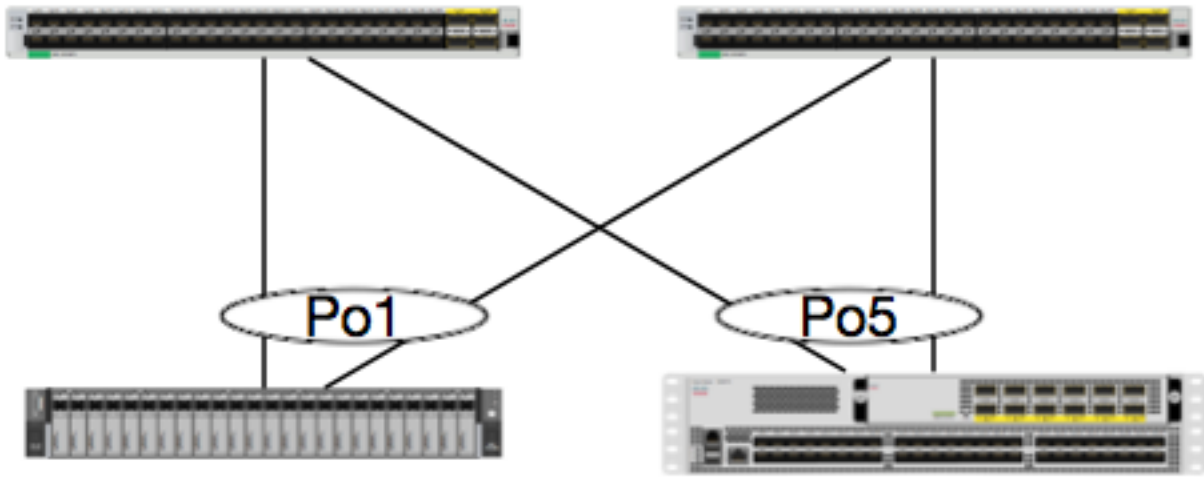
module-1(DBG-TAH-elam-insel9)# show platform internal hal tunnel rtep apd Non-Sandbox Mode
LEGEND: ----- ifId: Interface Id IP: IP address HwVrfId: Hardware Vrf Id SrcTepIdx: Source Tep
Index BDxlate: Egress BDxlate DstInfoIdx: Destination info index RwEncapIdx: Rw Encap Index
ECMPIdx: ECMP Index Num: 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

```

Ce tunnel a un **RwEncapIdx** (index d'encap de réécriture) de 6, qui est ce qui a été affiché dans l'**elam**.

1 PE --> L3 - Écoulement conduit

Topologie



**EP1
EPG1
0050.56a5.50ab
192.168.20.10/24**

**N5K -OSPF
100.100.100.100/32**

Dans cet exemple, nous dépisterons l'écoulement de paquet d'un paquet d'EP1 envoyant l'ICMP à un bouclage sur un N5K exécutant l'OSPF. N5K est connecté par l'intermédiaire d'un L3Out sur les mêmes paires de Commutateurs EX.

Puisque nous avons vérifié le PE de gens du pays programmant au début de ce document, nous permettons supposez que le PE est appris correctement dans le matériel et continuez en fonction à la vérification d'artère.

D'abord, vérifions l'état OSPF et la table de routage :

```
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
```

Ainsi nous savons que la table de routng affiche le prochain saut comme 5K chez 10.10.27.3.

Bon début, mais comment pouvons-nous vérifier ce que le matériel a ?

D'abord vérifions la table de juxtaposition dans le matériel pour s'assurer que nous avons l'ARP résolu à 10.10.27.3, et cela il est programmé avec l'interface appropriée :

```
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
```

Correspondances d'adresses MAC qui sur le 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
```

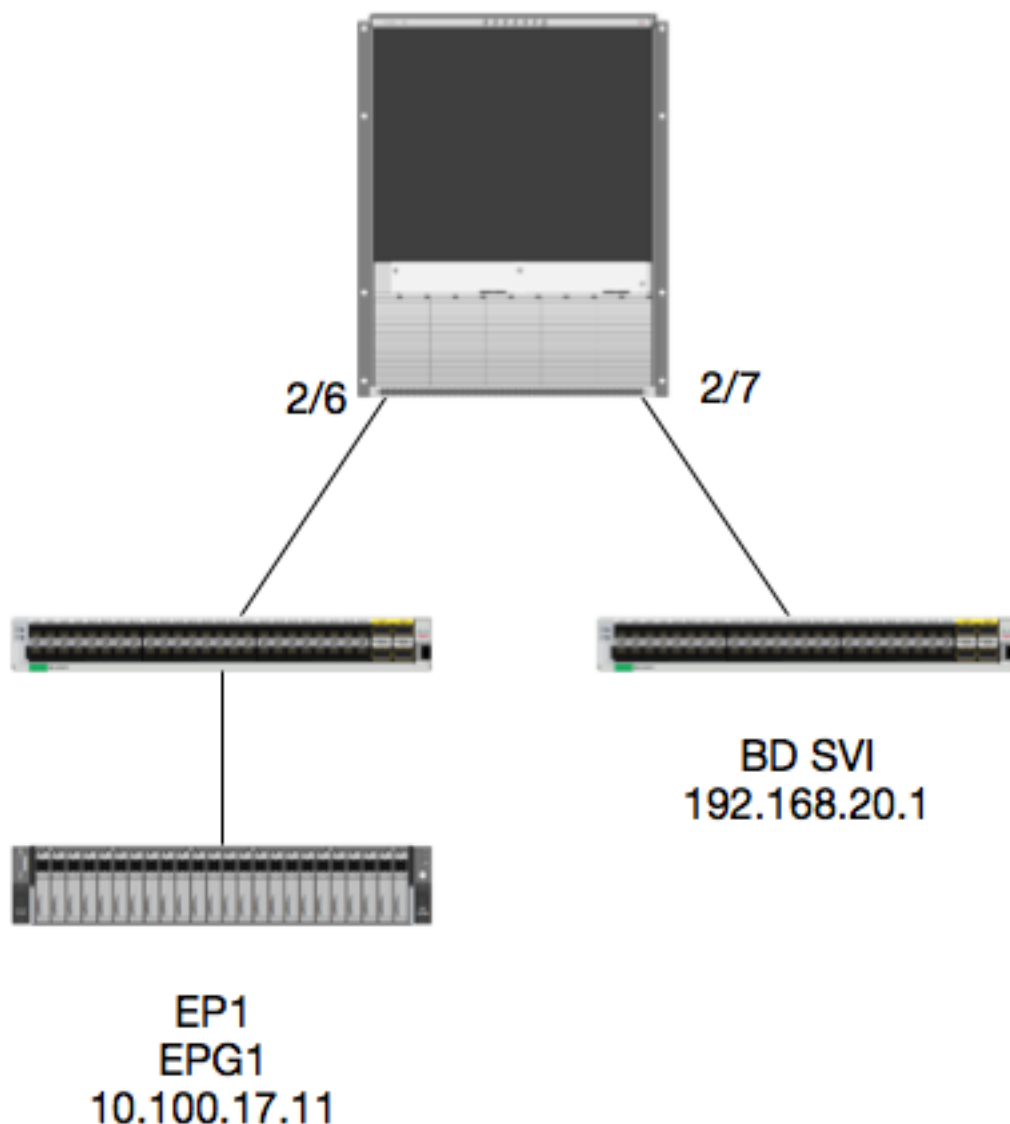
Sur les Plateformes EX, il y a un « hw_vrf_idx » qui est assigné à un VRF. Cet index sera mis en référence quand nous vérifions la programmation de matériel. Trouvons l'index :

```
module-1# show system internal eltcmc 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 tornbinfracwbd: 0 :::
torsbinfracwbd: 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 :::
```

Après que nous détectons la contiguïté, HAL devrait programmer une artère. Nous pouvons vérifier ceci utilisant la commande suivante :

```
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 ----->|<left 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TCAM | HIT |N| NB-ID | NB Hw | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
----- |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
```

Cette sortie nous fournit les informations concernant la prochaine artère de saut. 4612 est le



Logique

Dans cet exemple, nous dépisterons l'écoulement de paquet d'un paquet d'EP1 destiné à une interface virtuelle commutée par BD de distant (SVI). Le but de cet exemple sera de vérifier l'expédition d'épine pour s'assurer que le paquet est envoyé à la feuille correcte. Assumons le paquet a été envoyé au proxy d'épine sur la feuille d'entrée.

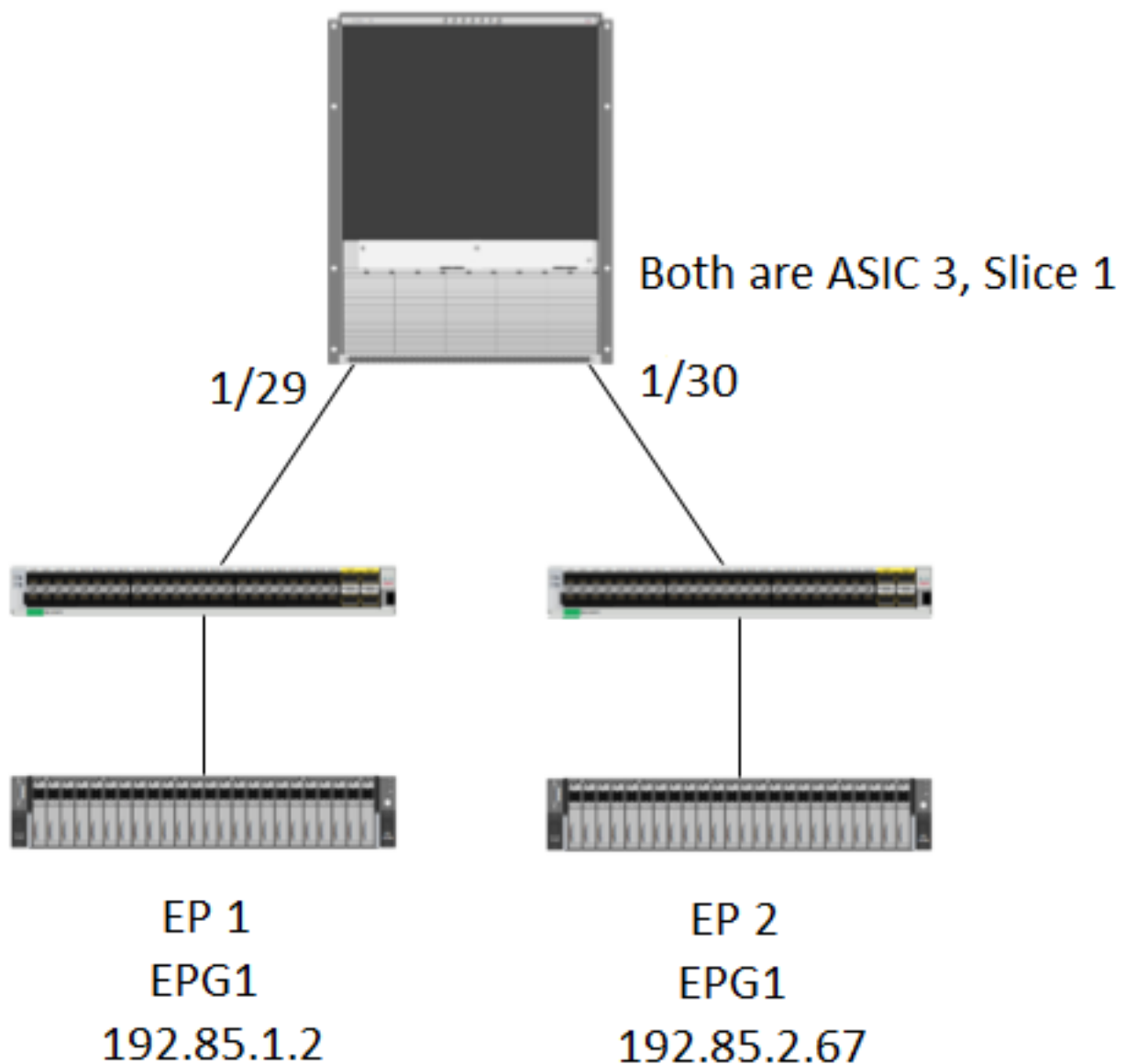
Sur l'épine, permettez-nous vérifient d'abord le Conseil des oracles Protocol (CAGE) pour l'IP de destination puisque le paquet est envoyé au proxy d'épine pour une consultation :

```
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
```

Vérifions quelle feuille a cette adresse TEP :

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

Puisque nous savons que le paquet entre dans l'épine sur le module 2, le port 6, nous pouvons nous relier au module 2 et regarder l'affichage de port.



Logique

Il y a quelques scénarios où nous attrapons un paquet qui n'a pas un Ovector dans le « **show platform la table du l'interne-port I2 pi hal interne** ». Dans le scénario ci-dessous, nous attrapons réellement le paquet revenant de FM, ainsi nous devons regarder une table différente pour voir quel panneau avant mettre en communication le paquet sélectionne.

Notez que la topologie ci-dessus est un environnement complètement différent où le trafic de transit est appris (aucun routage de proxy). Le module est un 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
```