

# Hardware EX: Buceo de profundidad del reenvío de paquete ACI.

## Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Escenarios](#)

[2 EP en la misma hoja EPG/Same - Trama conmutada](#)

[Topología](#)

[ELAM](#)

[2 EP en diversa hoja EPG/Same - Paquete ruteado](#)

[Topología](#)

[ELAM](#)

[2 EP en diversa hoja EPG/Different - Paquete ruteado](#)

[Topología](#)

[ELAM](#)

[1 EP --> L3 hacia fuera - Flujo ruteado](#)

[Topología](#)

[ELAM](#)

[1 EP --> telecontrol EP o SVI - Verificación de la espina dorsal](#)

[Topología](#)

[Lógica](#)

[IP sintetizado](#)

[Módulo de recursos físicos ELAM](#)

[Escenario adicional: Consiguiendo un Ovector que no está en del "la salida puerto interno pi hal"](#)

[Topología](#)

[Lógica](#)

## Introducción

Este documento describe diversos escenarios de la expedición usando el hardware ASIC de la generación nueva en la infraestructura céntrica de la aplicación (ACI). Mostrará cómo verificar el hardware se programa correctamente y estamos remitiendo los paquetes a los puntos finales del destino correcto (EP) en los grupos apropiados del punto final (EPGs).

## Prerrequisitos

## Requisitos

No hay requisitos específicos para este documento.

## Componentes Utilizados

La información que contiene este documento se basa en estas versiones de software y hardware.

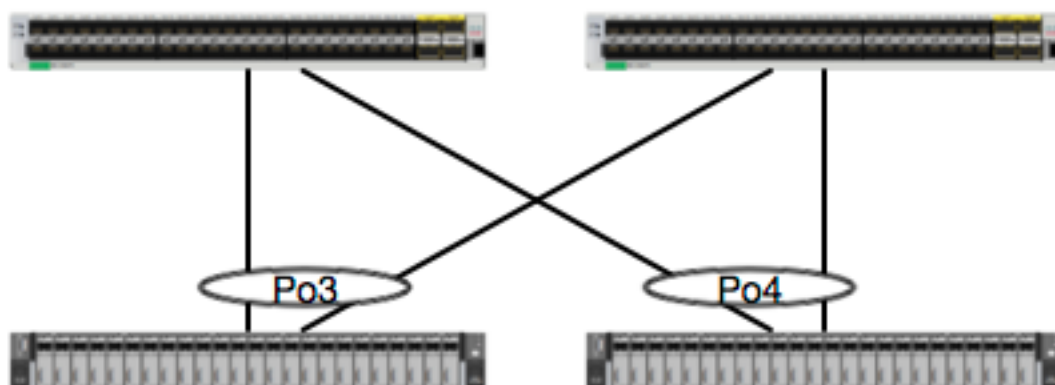
- Una tela ACI que consiste en dos Switches de la espina dorsal y dos Switches de la hoja usando el hardware EX
- Un host de ESXi con dos uplinks que va a cada uno de la hoja conmuta
- Dispositivo del nexu 5000 que actúa como router.
- Un regulador de la infraestructura de la directiva de la aplicación (APIC) que se utiliza para la configuración inicial

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

## Escenarios

### 2 EP en la misma hoja EPG/Same - Trama conmutada

#### Topología



**EP1**  
**EPG1**  
**0050.56a5.fccc**  
**192.168.20.2/24**

**EP2**  
**EPG1**  
**0050.56a5.6794**  
**192.168.20.3/24**

Dado esta topología, el flujo de EP1 a EP2 es un flujo L2 y se debe conmutar localmente en cualquier hoja el tráfico de origen viene adentro. La primera cosa a marcar con los flujos de la capa 2 (L2) es la tabla de direcciones del mac a determinar si y donde las tramas recibidas del Switch:

```
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
```

Para ver la encapsulación vlan, podemos marcar la base de datos EP también:



## 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
```

Grande, así que Leaf4 recibió la trama en Asic 0 rebanadas 1. Con ELAM en el nuevo hardware, hay un nuevo campo que es muy importante al resolver problemas: **ovector\_idx**. Este índice es el índice del puerto físico que la trama/el paquete se debe remitir de. Una vez que usted tiene el **ovector\_idx**, podemos utilizar este comando de encontrar qué puerto asocia:

```
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
```

El Switch piensa que el paquete se debe remitir de las interfaces Ethernet 1/32. ¿Es ese PO4 donde hemos aprendido que los address del 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)
```

Sí, así que el paquete Br remitido de la interfaz 1/32 a la computadora principal de destino.

## 2 EP en diversa hoja EPG/Same - Paquete ruteado



```

===== 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*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

```

La tabla HAL Layer3 (I3) es mismo uself puesto que nos da la información VLAN/Port para los EP aprendidos I3. Sabemos que el destino existe de un Po4, así que el paquete se debe remitir fuera de cualquier puerto en Po4.

¡Ejecutemos un ELAM y veamos lo que conseguimos!

## 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, así que accionamos el paquete, y nosotros encontró que el “ovector\_idx” es 0x9E. El índice del ovector es el índice phycial saliente de la interfaz que el paquete se debe remitir de. Veamos qué puerto tiene ese índice:

```

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 - - 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 - -
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 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

```







```

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

```

Ahora, con los destinos remotos en el hardware EX, hay 2 valores ELAM que son muy importantes al resolver problemas el flujo de paquetes. El ovector\_idx tiene gusto antes, y el 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

```

En el hardware EX, tenemos la capacidad de conducir el puerto destino que el paquete se debe remitir de. Antes, generalmente marcamos el idx del encap y acabamos de verificamoslo que el idx del destino era el túnel correcto. Aquí podemos verificar qué mapas del puerto 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: 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 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 0 1 5 3 2 2 D-350 - 400 0 0 0 1 0

```

El Switch piensa que debe remitirlo a la espina dorsal en la interfaz Eth1/49. ¿Pero cómo podemos verificar el encap estamos correctos?

Primero necesitamos mirar la información de hardware sobre el túnel. Podemos hacer esto funcionando con 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 salida nos da algunos valores que cuidamos alrededor:

lflid - La interfaz ID afectada un aparato al túnel

IP - El IP del destino. Esto debe hacer juego ELTMC.

L3 lflid - Las interfaces de la capa 3 que el Switch puede utilizar para remitir al destino apropiado.

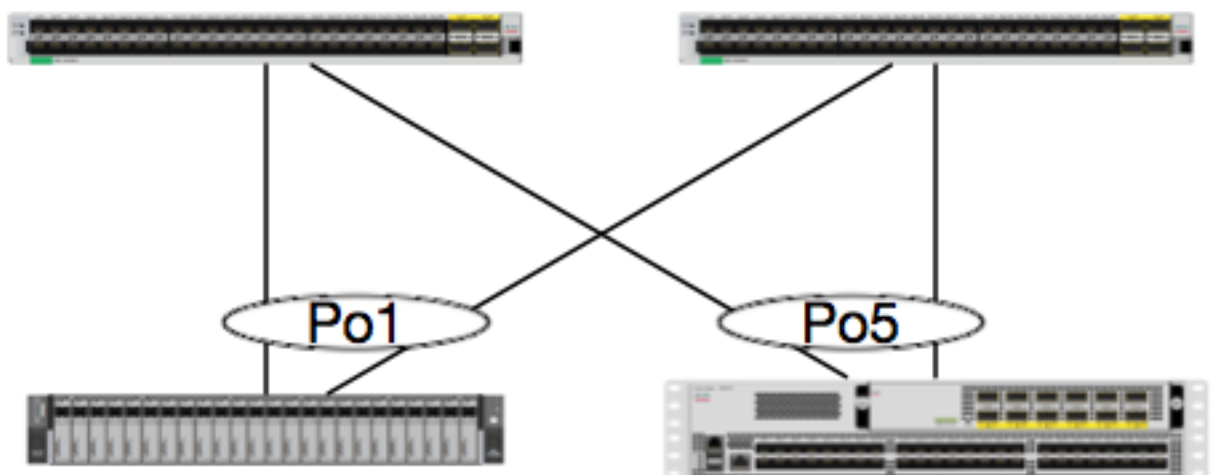
Una vez que conocemos el lflid, podemos verificar que el encap que conseguimos en el elam hacemos juego el destino del túnel:

```
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 tiene un RwEncapIdx (índice del encap de la reescritura) de 6, que es qué fue visualizada en el elam.

## 1 EP --> L3 hacia fuera - Flujo ruteado

### Topología



EP1  
EPG1  
0050.56a5.50ab  
192.168.20.10/24

N5K -OSPF  
100.100.100.100/32

En este ejemplo, seguiremos el flujo de paquetes de un paquete de EP1 que envía el ICMP a un

loopback en un N5K que ejecuta el OSPF. N5K está conectado vía un L3Out en los mismos pares de Switches EX.

Puesto que hemos verificado el EP local que programaba al principio de este documento, nos dejamos asumir que el EP está aprendido correctamente en hardware y continúe encendido a la verificación de la ruta.

Primero, marquemos el estado OSPF y la tabla de ruteo:

```
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
```

Sabemos tanto como la tabla del routing muestra el salto siguiente como el 5K en 10.10.27.3.  
¿Buen comienzo, pero cómo podemos verificar lo que tiene el hardware?

En primer lugar controle la tabla de adyacencia en hardware aseguremosnos nos tienen ARP resueltos a 10.10.27.3, y eso se programa con la interfaz correcta:

```
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
```

Coincidencias de las direcciones MAC que en el 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
```

En las Plataformas EX, hay un “hw\_vrf\_idx” que se asigna a un VRF. Este índice será referido cuando verificamos la programación del hardware. Encontremos el índice:

```
module-1# show system internal eltc 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 tornbinfracrhwbd: 0 :::
torsbinfracrhwbd: 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 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 :::
```

Después de que detectemos la adyacencia, el HAL debe programar una ruta. Podemos marcar esto usando el siguiente comando:

```
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
```



Como podemos ver, la tabla tiene la información correcta:

L2 INTF: 0x16000004 ---> el ifIndex del canal del puerto 5

GOLPE IDX: El índice conducido del hw Idx NOTA en las rutas hal l3

MAC: 8c:60:4f:02:88:fc --> MAC del salto siguiente SVI en 5K

EPG: SCLASS de L3 EPG

Dirección IP: 10.10.27.3 ---> IP del salto siguiente del SVI en 5K

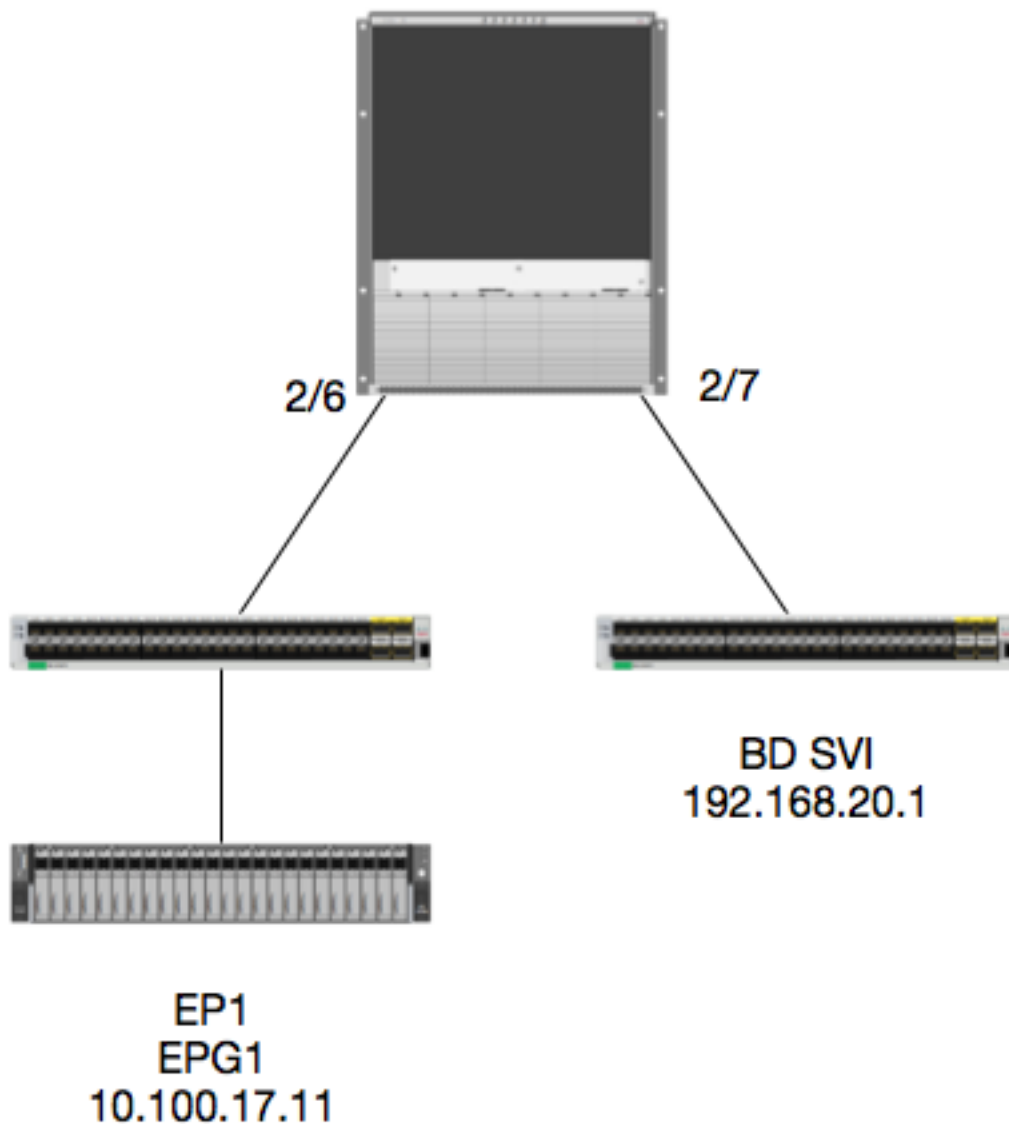
## ELAM

```
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 --> telecontrol EP o SVI - Verificación de la espina dorsal**

## Topología



## Lógica

En este ejemplo, seguiremos el flujo de paquetes de un paquete de EP1 destinado a un Switched Virtual Interface del BD del telecontrol (SVI). El propósito de este ejemplo será verificar la expedición de la espina dorsal para asegurarse que el paquete esté enviado a la hoja correcta. Asumamos el paquete fue enviado al proxy de la espina dorsal en la hoja del ingreso.

En la espina dorsal, déjenos primero verificar al consejo del protocolo de los oráculos (GALLINERO) para el IP de destino puesto que el paquete se envía al proxy de la espina dorsal para las operaciones de búsqueda:

```
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
```

Verifiquemos qué hoja tiene ese direccionamiento TEP:

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

Puesto que sabemos que el paquete está entrando en la espina dorsal en el módulo 2, el puerto 6, podemos asociar al módulo 2 y mirar la disposición del puerto.



Este Mbrld es la interfaz en USD ese las correspondencias a una interfaz en FM. Podemos descubrir que FM mirando USD interconecta y examinando el puerto 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
```

El "slot" es 0 basado, y la numeración de FM es 1 basado, así que necesitamos agregar 1 al número enumerado aquí. Esto significa que el paquete se debe enviar a FM 23.

## IP sintetizado

Apenas como en alpino, hay un IP sintetizado usado como el IP Address externo para determinar el hash para las operaciones de búsqueda del GALLINERO. Para encontrar esto, usted necesita ejecutar este comando y grep para el 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
```

Esto nos muestra que 1.203.211.185 es nuestro IP sintetizado. De acuerdo con esto, podemos también fijar el "DST IP externo" en nuestro elam de FM para ser éste. Debemos accionar en FM:

## Módulo de recursos físicos 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, vacie el informe completo, pero miremos el ovector\_idx para este paquete que accionamos:

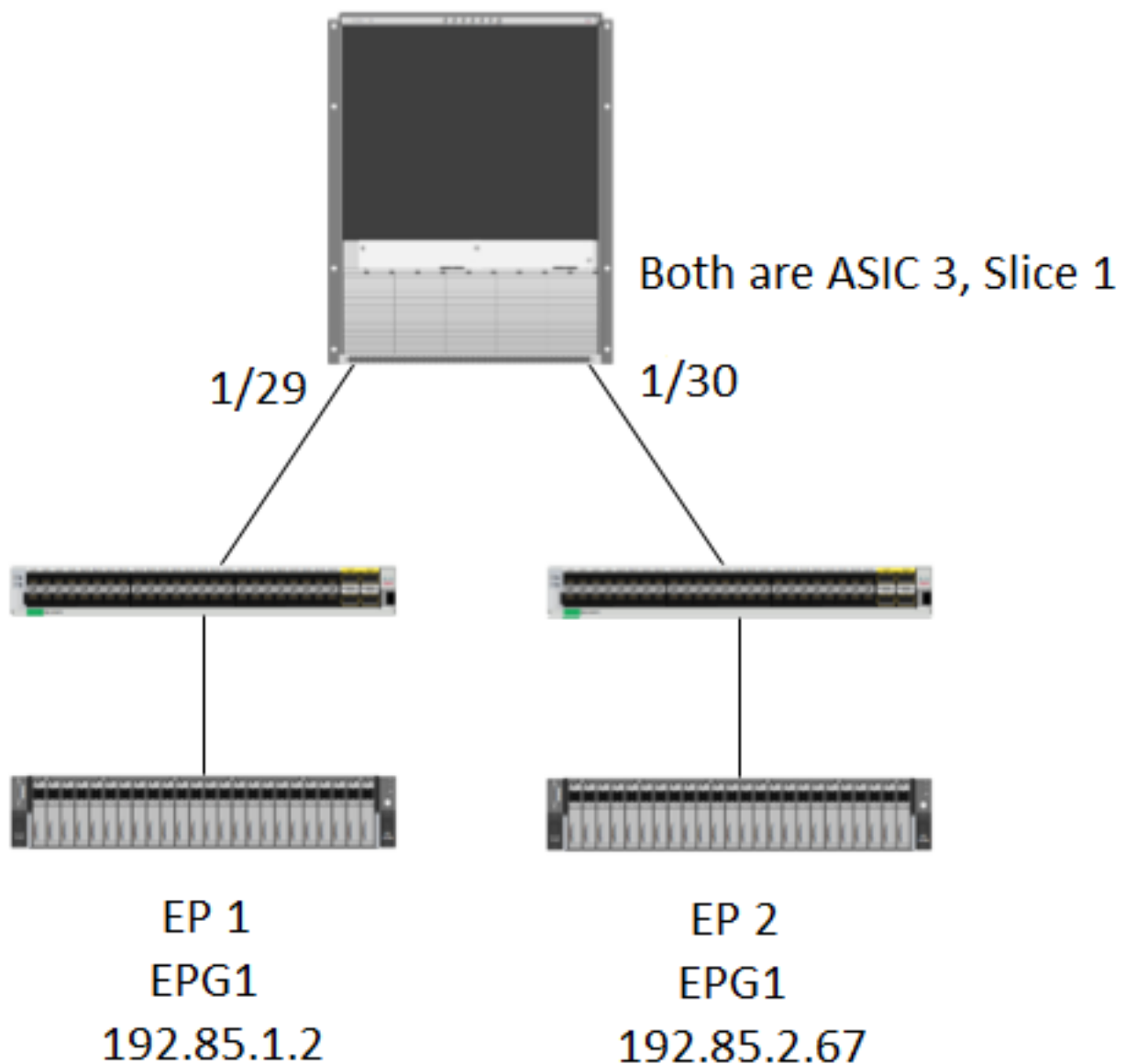
```
lac_elam_out_sidebnd_no_spare_vec.ovector_idx: 0x20 <----- Índice de Ovector usado en el
comando abajo
```

Cómo imaginamos que la interfaz tiene ese ovector?. En FM, ejecute esto:

```
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: 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 b1 fc0-lc1:1-1 1 0 14 0 39 2 8
```







## Lógica

Hay algunos escenarios donde cogemos un paquete que no tenga un Ovector en “**demostración la tabla del puerto interno I2 plataforma pi hal interno**”. En el escenario abajo, estamos cogiendo realmente el paquete que se vuelve de FM, así que necesitamos mirar una diversa tabla para ver qué panel frontal está seleccionando virar el paquete hacia el lado de babor.

Observe que la topología antedicha es un entorno totalmente diverso donde está docto el tráfico de tránsito (ningún proxy que rutea). El módulo es 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
```



