

Hardware EX: Análisis en profundidad de reenvío de paquetes de ACI.

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Escenarios](#)

[2 EP en el mismo EPG/Misma hoja - Trama conmutada](#)

[Topología](#)

[ELAM](#)

[2 EP en diferentes EPG/Misma hoja - Paquete enrutado](#)

[Topología](#)

[ELAM](#)

[2 EP en diferentes EPG/hojas diferentes - paquetes enrutados](#)

[Topología](#)

[ELAM](#)

[1 EP —> L3 out - Flujo ruteado](#)

[Topología](#)

[ELAM](#)

[1 EP —> EP remoto o SVI - Verificación de columna](#)

[Topología](#)

[Lógica](#)

[IP sintética](#)

[Módulo de fabric ELAM](#)

[Escenario adicional: Obtención de un vector que no está en la salida "hal internal-port pi"](#)

[Topología](#)

[Lógica](#)

Introducción

Este documento describe diferentes escenarios de reenvío mediante switches ACI basados en "EX" en Application Centric Infrastructure (ACI). Mostrará cómo verificar que el hardware está programado correctamente y estamos reenviando paquetes a los terminales de destino (EP) correctos en los grupos de terminales adecuados (EPG).

Prerequisites

Requirements

No hay requisitos específicos para este documento.

Componentes Utilizados

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

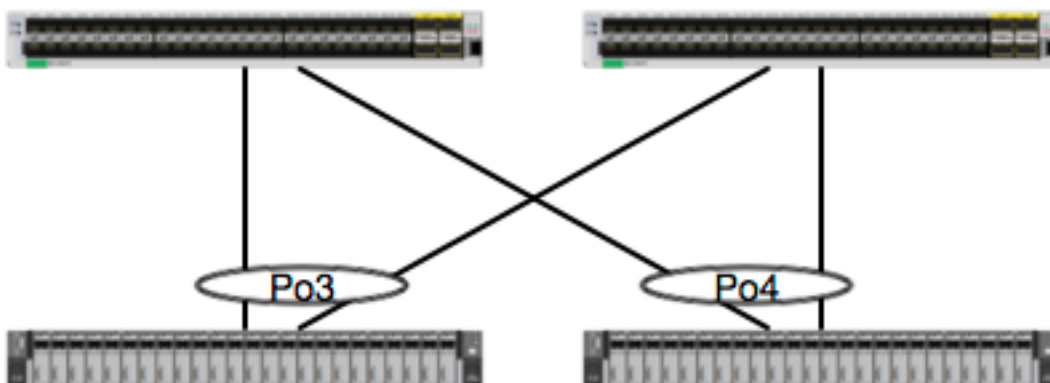
- Fabric ACI que consta de dos switches de columna y dos switches de hoja que utilizan hardware EX
- Host ESXi con dos enlaces ascendentes que van a cada uno de los switches hoja
- Dispositivo Nexus 5000 que actúa como router.
- Controlador de infraestructura de políticas de aplicaciones (APIC) que se utiliza para la configuración inicial

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Escenarios

2 EP en el mismo EPG/Misma hoja - Trama conmutada

Topología



EP1
EPG1
0050.56a5.fccc
192.168.20.2/24

EP2
EPG1
0050.56a5.6794
192.168.20.3/24

Dada esta topología, el flujo de EP1 a EP2 es un flujo de L2 y se debe conmutar localmente en cualquier hoja en la que se conecte el tráfico de origen. Lo primero que se debe verificar con los flujos de Capa 2 (L2) es la tabla de direcciones mac para determinar si el switch recibió tramas y dónde las recibió:

```
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 vlan de encapsulación, también podemos verificar la base de datos EP:

```
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/
Interface          Encap          MAC Address        MAC Info/
      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/
Interface          Encap          MAC Address        MAC Info/
      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
```

Sabemos que FD_VLAN 30 coincide, pero siempre podemos validar la asignación en el software:

```
leaf4# show vlan extended | grep 2268
```

```
30 enet CE          vlan-2268
```

Y, por supuesto, podemos verificar el hardware para asegurarnos de que VLAN 30 se mapee a VLAN 2268 como la encapsulación del panel frontal.

```
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
      aces_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 los EP se aprenden en software, también podemos validar que el hardware programó la información L2 de estos EP. En el nuevo hardware, está la capa de abstracción de hardware (HAL), que es el estado de software del hardware. El trabajo de HAL es aceptar solicitudes de programación de software y enviarlas al hardware.

Para ver la información de hardware L2 sobre un terminal, podemos ver la tabla L2 en HAL para las direcciones mac dadas:

```

leaf4# vsh_lc
module-1# show platform internal hal ep l2 mac 0050.56a5.fccc
LEGEND:
-----
BDId:          BD Id                               BD Name:      BD
Name
T:             EP Type (Pl: Physical Vl: Virtual Xr: Remote)  EP Mac:       Mac
L2 IfId:       L2 Interface                          L2 IfName:    L2
IfName
FDId:          FD Id                               FD Name:      FD
Name
S Class:       S Class                             Age Intvl:    Age
Interval
P A:           Packet Action (F: Forward, T: Trap to CPU,
                L: Log & Forward, D: Drop, N: None)
S T:           Static Ep                             S E:
Secure EP
L D:           Learn Disable                          B N D:        Bind
Notify Disable
E N D:         Epg Notify Disable                    B E:
Bounce Enable
I D L:         IVxlan Dont Learn                      SPI:
Source Policy Incomplete
DPI:           Dest Policy Incomplete                SPA:
Source Policy Applied
DPA:           Dest Policy Applied                    DSS:          Dest
Shared Service
IL:           Is Local                               VUB:          Vnid
Use Bd
SO:           SA Only

L2 EP Count: 1

=====
=====
                                                    B E
I S D S D D   V
   BD         EP           L2      L2      FD      S      Age      P S S L N N
B D P P P P S I U S
BdId Name     T  Mac           IfId     Ifname     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 0 1 0 0
```

```
module-1# show platform internal hal ep l2 mac 0050.56a5.6794
```

```

=====
=====
                                                                 B E
I S D S D D   V
      BD           EP           L2           L2           FD           S           Age           P S S L N N
B D P P P P S I U S
BdId Name      T Mac           IfId           Ifname           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 0 1 0 0

```

Ahora que hemos mapeado el hardware, hagamos un ELAM y veamos adónde debe ir el paquete.

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

```

Genial, así que Leaf4 recibió la trama en Asic 0 Slice 1. Con ELAM en el nuevo hardware, hay un nuevo campo que es muy importante para la resolución de problemas: **ovector_idx**. Este índice es el índice de puerto físico del que se debe reenviar la trama/paquete. Una vez que tenga **ovector_idx**, podemos utilizar este comando para encontrar el puerto al que mapea:

```
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
 UC: UCPcCfg
 UM: UCPcMbr
 PROF ID: Lport Profile Id
 VS: VifStateTable
 Install
 RV: Rv VifTable
 Num. of Sandboxes: 1

Ing Lbl: Ingress Acl Label
 Egr Lbl: Egress Acl Label
 Reprogram:
 HI: LportProfile Hw

Sandbox_ID: 0, BMP: 0x0
 Port Count: 8

```

=====
=====
| Rep |                Uc   Uc                |                Reprogram                | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| NI Vif | RvV | 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 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
- -      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
- -      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
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
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 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 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
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
D-350 -   400 0     0 0   1 0
  
```

El switch piensa que el paquete debe reenviarse fuera de la interfaz Ethernet 1/32. ¿Es ese PO4 donde hemos aprendido esa dirección 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
  
```

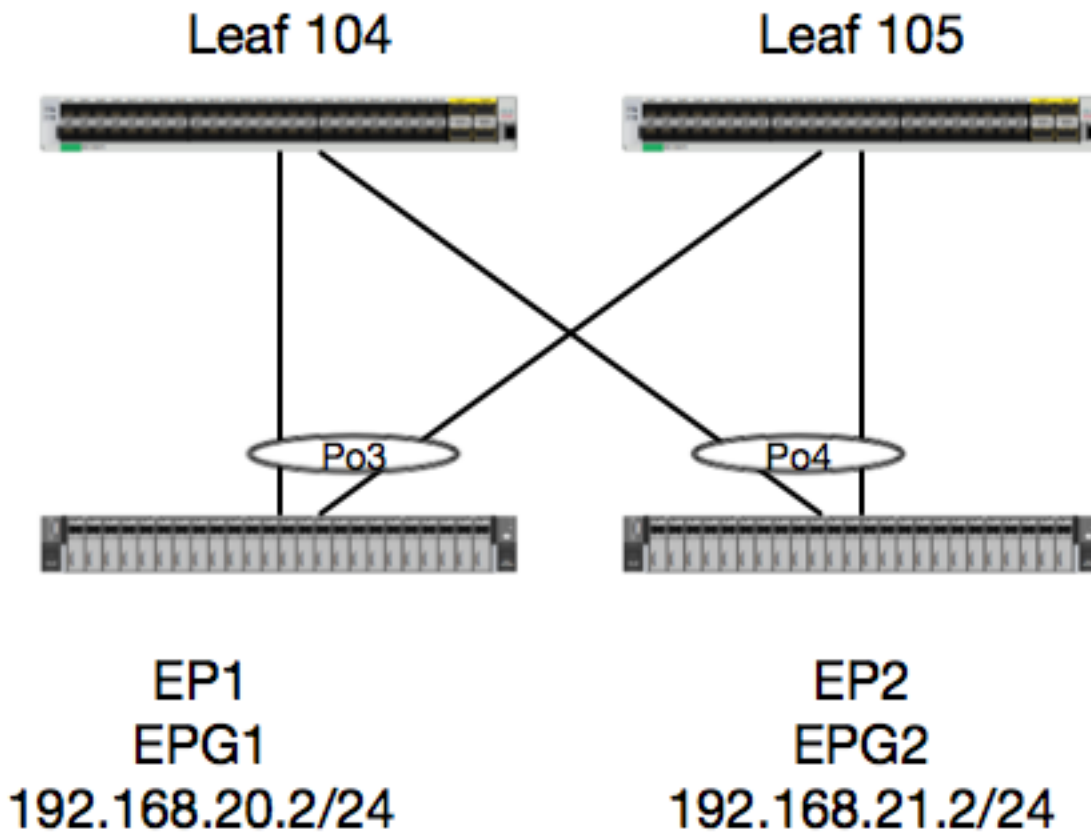
```

-----
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, de modo que el paquete se reenviará desde la interfaz 1/32 al host de destino.

2 EP en diferentes EPG/Misma hoja - Paquete enrutado

Topología



En este ejemplo, rastreamos el flujo de paquetes de un paquete de EP1 a EP2 donde existen en el mismo par de hojas de vPC. Los dos EP están en diferentes EPG usando diferentes BD.

Lo primero que siempre hay que hacer es revisar la base de datos de EP para ver si hemos aprendido los del 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

VLAN/ Interface	Encap	MAC Address	MAC Info/
Domain	VLAN	IP Address	IP Info
30	vlan-2268	0050.56a5.fccc	LV
po3			
Joey-Tenant:Joey-Internal	vlan-2268	192.168.20.2	LV
po3			

```
calo2-leaf4# show endpoint ip 192.168.21.2
```

Legend:

O - peer-attached H - vtep a - locally-aged S - static
V - vpc-attached p - peer-aged L - local M - span
s - static-arp B - bounce

```

+-----+-----+-----+-----+
----+
      VLAN/          Encap          MAC Address          MAC Info/
Interface
      Domain          VLAN          IP Address          IP Info
+-----+-----+-----+-----+
----+
8                vlan-2200    0050.56a5.0c11 LV
po4
Joey-Tenant:Joey-Internal    vlan-2200    192.168.21.2 LV
po4

```

Dado que hemos aprendido los EP y conocemos la información de IP, deberíamos poder ver la información de aprendizaje del EP en hardware:

```

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

```

```

LEGEND:
-----
VrfName:          Vrf Name                      T:              Type
(P1: 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 P1 10.6.112.1          1      0      1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -

```



```

L3 - 00:00:00:00:00:00 - - - 0.0.0.0
common*rewall Pl 10.6.114.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
common*rewall Pl 10.6.114.129 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
common*efault Pl 100.100.101.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
Joey-T*ternal Pl 192.168.1.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
Joey-T*ternal Xr 192.168.1.100 8013 128 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 -
L3 - 00:0c:0c:0c:0c:0c Tunnel12 Tunnel12 - 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 tabla HAL Layer3 (I3) es muy útil ya que nos proporciona información de VLAN/Puerto para los EP aprendidos de I3. Sabemos que el destino existe de un Po4, por lo que el paquete debe ser reenviado fuera de cualquier puerto en Po4.

¡Hagamos 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

```

Estupendo, entonces activamos el paquete y encontramos que "ovector_idx" es 0x9E. El índice de vectores es el índice de interfaz física saliente del que se debe reenviar el paquete. Veamos qué puerto tiene ese índice:

```

module-1(DBG-TAH-elam-insel6)# show platform internal hal l2 port gpd
Legend:
-----

```

```

IfId:      Interface Id
I P:       Is PC Mbr
Uc PC Cfg: UcPcCfg Idx
As:        Asic
Sl:        Slice
Ss:        Slice SrcId
srcid)
L S:       Local Slot
L3:       Is L3
P:         PifTable
RP:        Rw PifTable
IP:        If Profile Table
RS:        Rw SrcId Table
DP:        DPort Table
SP:        SrcPortState Table
RSP:       RWSrcPortstate Table
UC:        UCPcCfg
UM:        UCPcMbr
PROF ID:   Lport Profile Id
VS:        VifStateTable
Install
RV:        Rw VifTable
Num. of Sandboxes: 1

```

```

Sandbox_ID: 0, BMP: 0x0
Port Count: 8

```

```

=====
=====

```

Rep		Uc		Reprogram																													
NI Vif	RwV	Ing	Egr	I	PC	Pc	V	R	PROF	H	L	R	I	R	D	R	U	U	X	L	Xla	Ovx	N										
IfId	Ifname	P	Cfg	MbrID	As	AP	Sl	Sp	Ss	Ovec	S	P	P	P	S	P	Sp	Sp	C	M	L	3	Idx	Idx	L3								
L3 Tid	Tid	Lbl	Lbl	S	V	ID	I																										
1a004000	Eth1/5	1	0	1d	0	d	0	c	18	18	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
-	-	800	0	0	1	0	0																										
1a005000	Eth1/6	1	0	b	0	e	0	d	1a	1a	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
-	-	800	0	0	1	0	0																										
1a006000	Eth1/7	0	26	5	0	f	0	e	1c	1c	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
D-256	-	800	0	0	1	c	0																										
1a007000	Eth1/8	0	2f	7	0	10	0	f	1e	1e	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
D-199	-	800	0	0	1	2e	0																										
1a01e000	Eth1/31	1	0	2d	0	37	1	e	1c	9c	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
-	-	0	0	0	1	0	0																										
1a01f000	Eth1/32	1	0	3d	0	38	1	f	1e	9e	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
-	-	0	0	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	0	0	0							
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	0	0							
D-350	-	400	0	0	0	1	0																										

Parece que deberíamos enviarlo al puerto 1/32, ¿es correcto?

```

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

```

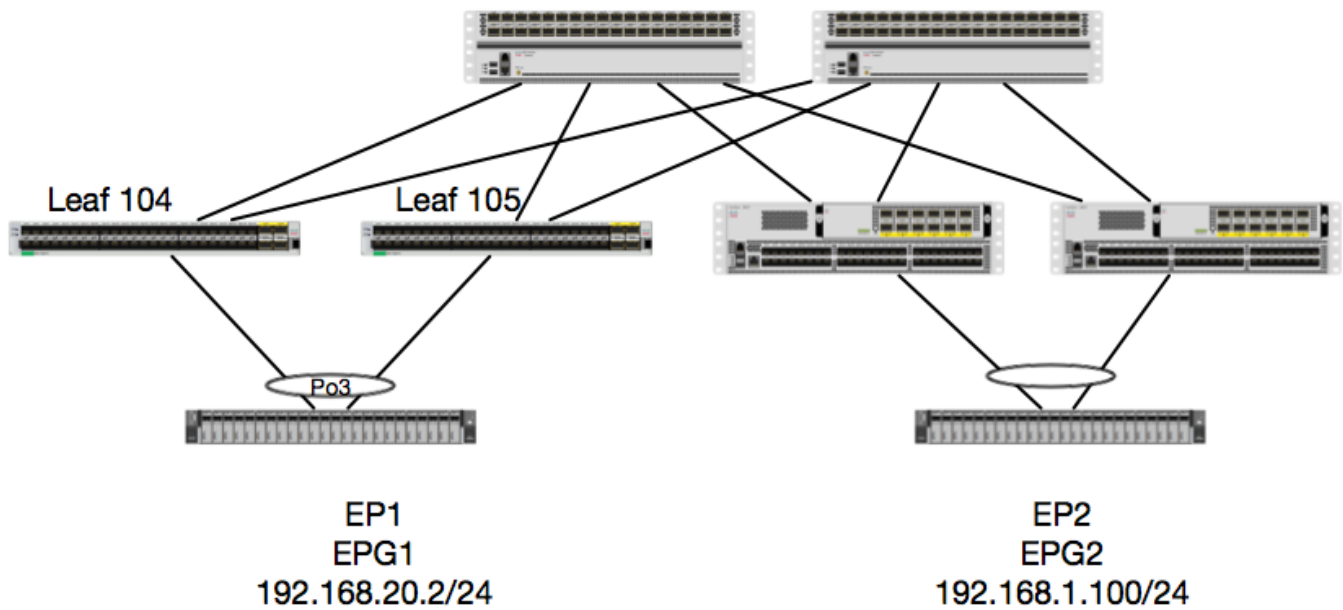
F - Configuration failed

```
-----  
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, esto es correcto.

2 EP en diferentes EPG/hojas diferentes - paquetes enrutados

Topología



En este ejemplo, rastreamos el flujo de paquetes de un paquete de EP1 a EP2 donde el EP1 existe en un par EX vPC y el EP2 existe en un par remoto de hoja vPC de Generación 1. Los dos EP están en diferentes EPG usando diferentes BD.

De nuevo, veamos dónde se aprenden los 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
```

```
-----+-----+-----+-----+-----  
----+  
      VLAN/          Encap      MAC Address      MAC Info/  
Interface  
      Domain        VLAN      IP Address      IP Info  
-----+-----+-----+-----+-----  
----+  
30          vlan-2268    0050.56a5.fccc LV  
po3  
Joey-Tenant:Joey-Internal  vlan-2268    192.168.20.2 LV  
po3
```

calo2-leaf4# show endpoint ip 192.168.1.100

Legend:

O - peer-attached H - vtep a - locally-aged S - static
 V - vpc-attached p - peer-aged L - local M - span
 s - static-arp B - bounce

```

+-----+-----+-----+-----+
----+
      VLAN/          Encap          MAC Address          MAC Info/
Interface
      Domain          VLAN          IP Address          IP Info
+-----+-----+-----+-----+
----+
Joey-Tenant:Joey-Internal          192.168.1.100
tunnel2
  
```

Ahora, verifiquemos qué ha programado el hardware:

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

LEGEND:

```

-----
VrfName:          Vrf Name          T:          Type
(P1: Physical, V1: 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 Tunnel12 Tunnel12 -          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 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

```

El hardware cree que el EP existe en el túnel 2. ¿Cuál es el destino del túnel 2?

```

module-1# show system internal eltmc info interface tunnel2
IfInfo:
interface:          Tunnel12  :::          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
```

Dado que el destino existe fuera de un vPC, esa IP de destino debe ser la IP virtual vPC de los folletos remotos. Comprobemos una hoja remota y veamos:

```
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
```

Perfecto, por lo que aprendió el EP de destino del par vPC remoto. Veamos qué ve ELAM y verifiquemos que estamos reenviando el paquete correctamente:

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
```

Ahora, con los destinos remotos en el hardware EX, hay 2 valores ELAM que son muy importantes a la hora de resolver problemas de flujo de paquetes. El ovector_idx como 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 de destino del que se debe reenviar el paquete. Antes, solíamos comprobar el idx de encapsulado y verificar que el idx de destino era el túnel correcto. Aquí podemos verificar qué puerto se asigna 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          HI:           LportProfile Hw
VS:           VifStateTable
Install
RV:           Rw VifTable
Num. of Sandboxes: 1

Sandbox_ID: 0, BMP: 0x0
Port Count: 8

```

```

=====
=====
| Rep |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|  NI  | Vif  |  RwV  | Ing  | Egr  |  Uc  | Pc  |  L  | R I R D  | R  | U  | U  | X  | L  | Xla  | Ov  | Ox  | N  |
| IfId | Ifname |  Tid  | Lbl  | Lbl  | 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 0
- - - - - 800 0    0 1    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  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 0 0
- - - - - 800 0    0 1    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  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 0 0
D-256 - - - - - 800 0    0 1    c   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  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 0 0
D-199 - - - - - 800 0    0 1    2e  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  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 0    0 1    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  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    0 1    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  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 0 0 0 0 0
D-24d - - - - - 400 0    0 0    1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  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 0 0 0 0
D-350 - - - - - 400 0    0 0    1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

```

El switch piensa que debe reenviarlo al conmutador central en la interfaz Eth1/49. Pero, ¿cómo podemos verificar que la encapsulación es correcta?

En primer lugar, necesitamos ver la información de hardware sobre el túnel. Podemos hacerlo ejecutando este comando HAL:

```

module-1(DBG-TAH-elam-insel6)# show platform internal hal tunnel rtep pi
Non-Sandbox Mode
LEGEND:
-----

```


BDXlate: Egress BDXlate
RwEncapIdx: Rw Encap Index
Num: Number of hops
L2 Index: L2 Index

DstInfoIdx: Destination info index
ECMPIdx: ECMP Index
ECMPMbrIdx: ECMP member Index
RwDmacIdx: Rw Dmax Index

Num. of Sandboxes: 1

Sandbox_ID: 0, BMP: 0x0
Remote Tep Count: 15

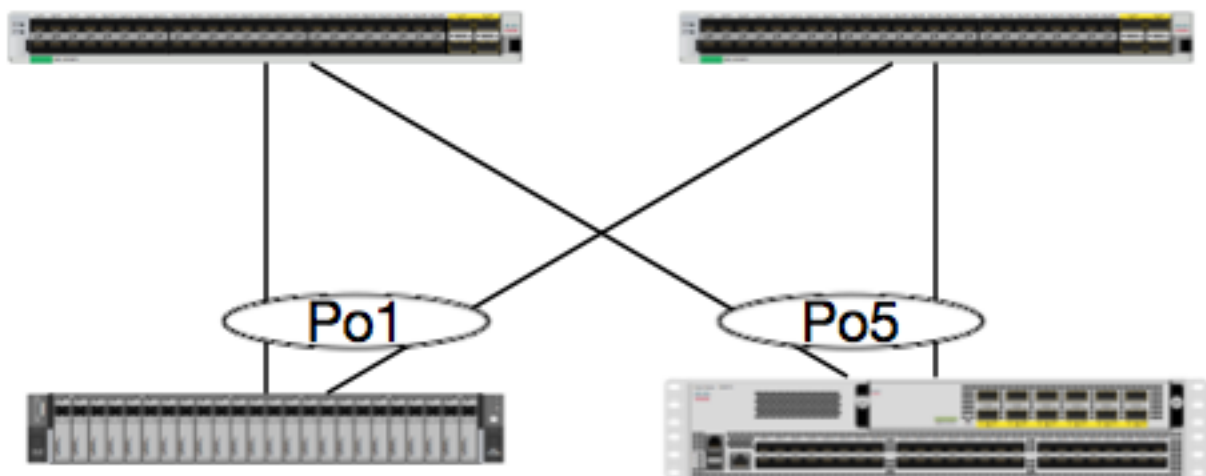
```
=====
=====
ifId      IP          HwVrfId BDXlate SrcTepIdx DstInfoIdx RwEncapIdx ECMPIdx  ECMPMbrIdx Num
L2Index  RwDmacIdx
=====
=====
18010002 192.168.120.67 2        1        3a9a     3005     6         0         0         2
1a030000 0                <---- RwEncapIdx is 6! Same as the "encap_idx" in the ELAM Report.
```

1a031000 1

Este túnel tiene un RwEncapIdx (Re-Write Encap Index) de 6, que es lo que se visualizó en el intervalo.

1 EP → L3 out - Flujo ruteado

Topología



EP1
EPG1
0050.56a5.50ab
192.168.20.10/24

N5K -OSPF
100.100.100.100/32

En este ejemplo, rastreamos el flujo de paquetes de un paquete desde el EP1 que envía ICMP a un loopback en un N5K que ejecuta OSPF. N5K se conecta a través de un L3Out en el mismo par de switches EX.

Dado que hemos verificado la programación de EP local al principio de este documento, supongamos que el EP se aprende correctamente en el hardware y continúe con la verificación de la ruta.

Primero, verifiquemos 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 que la tabla de ruteo muestra el salto siguiente como el 5K a 10.10.27.3. Buen comienzo, pero ¿cómo podemos verificar qué hardware tiene?

Primero verifiquemos la tabla de adyacencia en el hardware para asegurarnos de que ARP haya resuelto a 10.10.27.3 y que esté programado 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
```

Las direcciones MAC coinciden con las del 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" asignado a un VRF. Se hará referencia a este índice cuando verifiquemos la programación de hardware. Vamos a encontrar 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
```

```

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
ingressBdAcclLabel:  0   :::   ingBdAcclLb1Mask:  0
egressBdAcclLabel:  0   :::   egrBdAcclLb1Mask:  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

```

::::

Después de detectar la adyacencia, HAL debe programar una ruta. Podemos verificar esto usando el siguiente comando:

```
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 :Ivxlans
DoNotLearn DL : Dest Local  SA : Src Only          AI : Age Interval
|
SF : Static Flag        SH : Src Hit           DH: Dest Hit
|
-----
|      |      |      |      |      | LID |<-----
- Trie ----->|<Dleft Trie>|      |      |      |      |      |      |
| VRF |      |      |      |      |      |      |      |      |      |      |      |
|N| NB-ID | NB Hw | PID | FPID/ | TBI | |TRO|Ifindex|CLSS|CLP| AI |SH|DH| Flags  |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|B|      | Idx |      | TID |-----|---|-----|---|---|---|---|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|-----|<-----
- DLEFT ----->|-----|-----|-----|---|-----|---|---|---|---|---|-----|
|      |      |      |      |      |      |      |      |      |      |      | |
|N| NB-ID | NB Hw |      |      |      |      |      |      |      |      |      |
|      |      | Idx |      |      |      |      |      |      |      |      |      |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|<-----
- TCAM ----->|      |      |      |      |      |      |      |      |      |      |
|      |      |      |      |      |      |      |      |      |      |      | |
|N| NB-ID | NB Hw |      |      |      |      |      |      |      |      |      |
|      |      | Idx |      |      |      |      |      |      |      |      |      |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|<-----
-----
|Sandbox_ID: 0 Asic Bitmap: 0x0
-----
-----
```

```
module-1# show platform internal hal l3 routes | egrep 100.100.100.100
| 4612|          100.100.100.100/ 32| UC| e4| 4a04| TRIE| 10| 5/ 0|
6010|A| 7567| 802e| 186a| 1/ 2| 10| 0| 0| f| 3| 0| 0| 0|spi,dpi
```

Esta salida nos proporciona información sobre la ruta del siguiente salto. 4612 es el hw_vrf_idx del VRF jr:sb. Para que podamos verificar el próximo salto, el "NB Hw Idx" en TCAM se usará en la siguiente tabla:

```
module-1# show platform internal hal l3 nexthops
Non-Sandbox Mode
LEGEND:
-----
NHOP ID       : Nhop Identifier (Hex)          CONS       : H/W S/W info Consistency
TYPE         : Nexthop Type                    ACTN       : Nexthop Action
Vrf          : L3 Vrf of the Nhop      L3 INTF    : L3 interface index (Hex)
L2 INTF      : L2 interface index (Hex) BDID Or RvVRF : 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-----								-----Mac Entry-----																				
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	Rw	VRF	F	R	R	F	FP	I	IDX	Y	INTF	R	D	L	N	F	L	F	F	FID			
																L	N	N	N	D	P	P						
(Hex)	S	E	N	Vrf	(H)	(H)	(H)	R	F	N	R	PID	ID	D	(H)		P	(H)	N	L	D	B	L	D		T	V	(H)
Mac	CLSS P EPG E E E L I I				IP Address																							

```
module-1# show platform internal hal l3 nexthops | grep 802e
7567 N I F 5 901001c 16000004 1c 0 0 0 0 2e 9 0 802e 0 22 0 0 0 0 0 1 1 1
1214 8c:60:4f:02:88:fc 0 0 2c0d 0 0 0 0 0 0 10.10.27.3
```

Aquí, tomamos el "NB Hw ldx" y lo mapeamos al "HIT IDX". Esto nos muestra la entrada correspondiente a Next Hop MAC/IP. Esto equivale a ver "l3 defp show" y "l3 egress show" en Broadcom en switches de hoja ACI de primera generación.

Como podemos ver, la tabla tiene la información correcta:

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

IDX DE HIT: El índice impulsado desde Nb Hw ldx en rutas hal l3

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

EPG: CLASE DE EPG L3

IP Address: 10.10.27.3 —> IP de próximo salto de 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: 0x0000000000000000064646464
```

```
leaf6# cat elam_report.txt | grep ip.sa
sug_pr_lu_vec_l3v.ip.sa: 0x0000000000000000C0A8140A
```

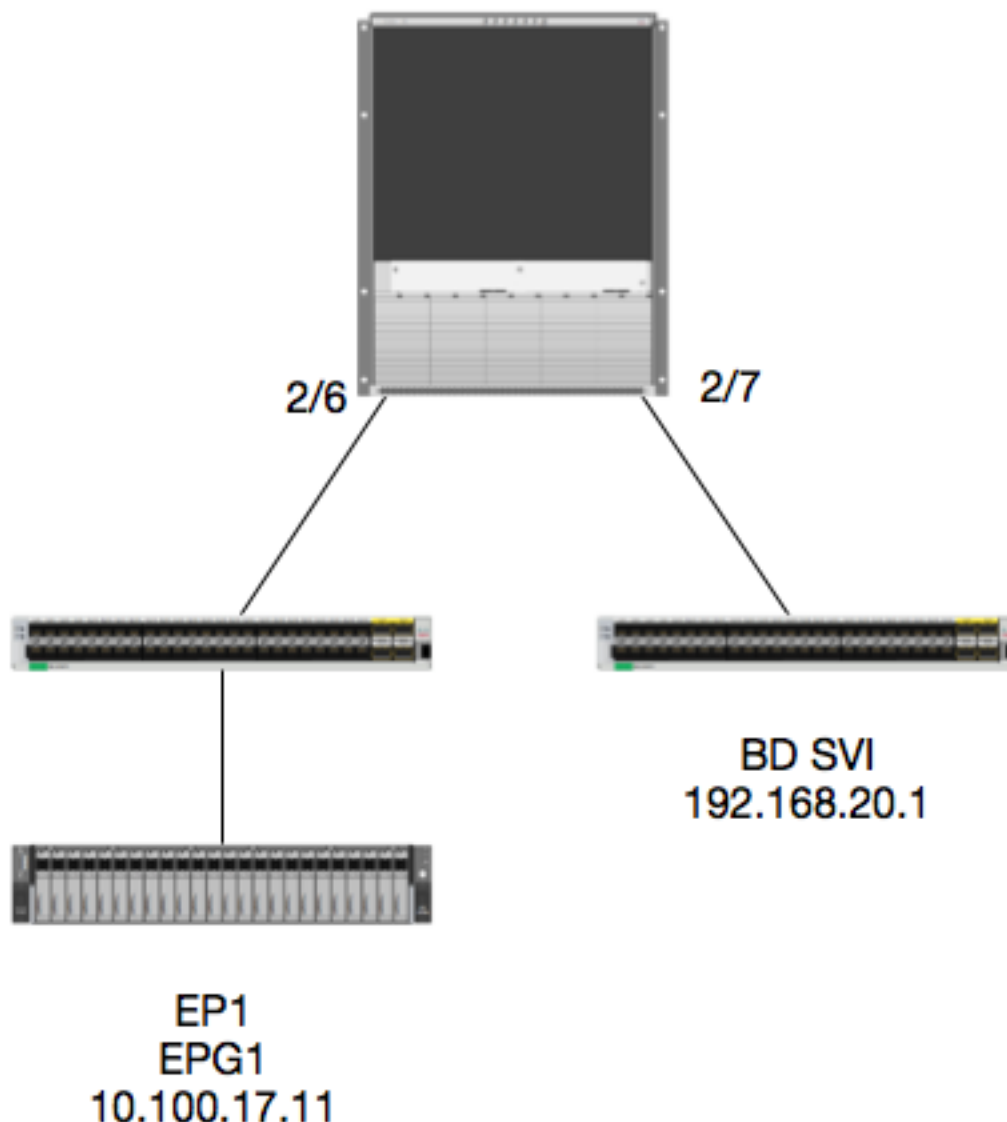
```
leaf6# cat elam_report.txt | grep adj
```

```
sug_lurw_vec.dst_addr.adj: 0x8C604F0288FC
sug_lurw_vec.dst_addr.adj.padfield: 0x04F0288FC
sug_lurw_vec.dst_addr.adj.idx: 0x2318
sug_lurw_vec.adj_vld: 0x0
```

```
leaf6# cat elam_report.txt | grep macdarslt.hit_idx
sug_fpc_lookup_vec.fplu_vec.rslt.macdarslt.hit_idx: 0x802E
```

1 EP → EP remoto o SVI - Verificación de columna

Topología



Lógica

En este ejemplo, rastreamos el flujo de paquetes de un paquete desde EP1 destinado a una interfaz virtual conmutada por BD remota (SVI). El propósito de este ejemplo será verificar el reenvío de columna para asegurarse de que el paquete se envíe a la hoja correcta. Supongamos que el paquete fue enviado al proxy de columna en la hoja de ingreso.

En la columna vertebral, primero verifiquemos el protocolo del Consejo de Oráculos (COOP) para

la IP de destino, ya que el paquete se envía al proxy de columna para una 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 esa dirección TEP:

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

Dado que sabemos que el paquete ingresa en la columna en el Módulo 2, Puerto 6, podemos conectarlo al Módulo 2 y observar la disposición del puerto.

```
spine1# vsh
Cisco iNX-OS Debug Shell
This shell should only be used for internal commands and exists
for legacy reasons. User should use ibash infrastructure as this
will be deprecated.
calo1-spine1# attach module 2
Attaching to module 2 ...
To exit type 'exit', to abort type '$.'
No directory, logging in with HOME=/
Bad terminal type: "xterm-256color". Will assume vt100.
Cisco iNX-OS Debug Shell
This shell should only be used for internal commands and exists
for legacy reasons. User should use ibash infrastructure as this
will be deprecated.
Loading parse tree (LC). Please be patient...
module-2#

module-2# show platform internal hal l2 port gpd
Legend:
-----
IfId:      Interface Id
I P:       Is PC Mbr
Uc PC Cfg: UcPcCfg Idx
As:        Asic
Sl:        Slice
Ss:        Slice SrcId
srcid)
L S:       Local Slot
L3:        Is L3
P:         PifTable
RP:        Rw PifTable
IP:        If Profile Table
RS:        Rw SrcId Table

IfName:    Interface Name
IfId:      Interface Id
Uc PC MbrId: Uc Pc Mbr Id
AP:        Asic Port
Sp:        Slice Port
Ovec:      Ovector (slice |

Reprogram:
Xla Idx:   Xlate Idx
Ovx Idx:   OXlate Idx
N L3:      Num. of L3 Ifs
NI L3:     Num. of Infra L3 Ifs
```

```

DP:   DPort Table
SP:   SrcPortState Table
RSP:  RWSrcPortstate Table
UC:   UCPcCfg
UM:   UCPcMbr
PROF ID:      Lport Profile Id
VS:   VifStateTable
Install
RV:   Rw VifTable
Num. of Sandboxes: 1

```

```

Sandbox_ID: 0, BMP: 0x0
Port Count: 7

```

```

=====
=====

```

Rep		Uc		Uc		Reprogram																				
NI Vif	RwV	Ing	Egr	I	PC	Pc	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	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	1	b	b	1	1
D-f3	D-61	100	0	0	0	1	0																			
1a081000	Eth2/2	0	9b	22	0	d	0	c	18	18	1	0	0	0	0	0	0	0	0	0	0	1	c	c	1	1
D-1ee	D-30b	100	0	0	0	1	0																			
1a084000	Eth2/5	0	9e	1e	0	3d	1	14	28	a8	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
D-19a	D-2ee	100	0	0	0	1	0																			
1a085000	Eth2/6	0	9f	24	0	39	1	10	20	a0	1	0	0	0	0	0	0	0	0	0	0	1	e	e	1	1
D-87	D-184	100	0	0	0	1	0																			
1a086000	Eth2/7	0	a0	26	0	35	1	c	18	98	1	0	0	0	0	0	0	0	0	0	0	1	d	d	1	1
D-1d0	D-357	100	0	0	0	1	0																			
1a088000	Eth2/9	0	a2	20	1	d	0	c	18	18	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D-3ea	D-1a9	100	0	0	0	1	0																			

Ethernet 2/6 es la interfaz que se conecta a la hoja 6 en ASIC 0 SLICE 1

Ahora sabemos en qué ASIC se ejecutará 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

```

Si miramos el ELAM, podemos encontrar el Índice de Vectores:

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

Ahora, ¿cómo mapeamos 0xb8 a un puerto? Como sabemos que el paquete debe enviarse a un módulo de fabric (FM) para una búsqueda, podemos observar la asignación de puertos internos para encontrar el FM más sencillo:

```
module-2# show platform internal hal l2 internal-port pi
```

```
Num. of Sandboxes: 1
```

```
Legend:
```

```
-----
```

IfId:	Interface Id	IfName:	Interface Name
As:	Asic	AP:	Asic Port
Sl:	Slice	SP:	Slice Port
Ss:	Slice SrcId	Ovec:	Ovector
UcPcCfgId:	Uc Pc CfgId	Lb MbrId:	LB MbrId

```
Sandbox_ID: 0, BMP: 0x0
```

```
Internal Port Count: 32
```

```
=====
```

IfId	IfName	As	AP	Sl	SP	Ss	Ovec	UcPc CfgId	Lb 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
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

```
=====
```

Usando ASIC0 / Ovec B8, obtenemos MbrId 0x7, Slice no importa.

Este MbrId es la interfaz en USD que se asigna a una interfaz en un FM. Tenga en cuenta que este IdMbr está en hexadecimal y debe convertirse a decimal.

Podemos averiguar qué FM mirando las interfaces USD e inspeccionando el Puerto 7:

```
module-2# show platform internal usd port info | grep -A 3 "Int 7"(if the interface has multiple digits, will be "Int##" with no space)
```

```
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
```

La "ranura" se basa en 0 y la numeración FM se basa en 1, por lo que necesitamos agregar 1 al número que se enumera aquí. Esto significa que el paquete debe enviarse a FM 23.

IP sintética

Al igual que en Alpine, hay una IP sintética utilizada como dirección IP externa para determinar el hash para la búsqueda COOP. Para encontrar esto, necesita ejecutar este comando y grep para la IP DST interna:

```
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 nuestra IP sintética. Basándonos en esto, también podemos configurar la "IP de DST externa" en nuestro equipo FM para que sea esto.

Deberíamos activar en el FM:

Módulo de fabric 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, volteemos el informe completo, pero echemos un vistazo a ovector_idx para este paquete que activamos:

lac_elam_out_sidebnd_no_repuesto_vec.ovector_idx: 0x20 ← Índice de vector utilizado en el comando siguiente

¿Cómo averiguamos qué interfaz tiene ese vector? En el FM, ejecute esto:

**** Debido al error [CSCvf42796](#) , añada todos los comandos FM con "| no más". De lo contrario, es posible que algunas entradas de la tabla no se muestren en el resultado final.**

```
module-23 (DBG-TAH-elam-insel13)# show platform internal hal l2 port gpd | no-more
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           HI:          LportProfile Hw
VS:         VifStateTable
Install
    RV:     Rw VifTable
Num. of Sandboxes: 1

Sandbox_ID: 1, BMP: 0x1
  Port Count: 8

=====
| Rep |                               |   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 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      fc0-1c1:0-0 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
af      fc0-1c1:0-1 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
b0      fc0-1c1:1-0 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
b1      fc0-1c1:1-1 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
b2      fc0-1c1:2-0 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
b3      fc0-1c1:2-1 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
b4      fc0-1c1:3-0 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

Ese ovector se mapea a LC1 (tarjeta de línea en la ranura 2, ya que está basada en 0), en ASIC 0 / SLICE 0. Como sabemos por el ELAM ejecutado originalmente en la LC, activamos en esta porción:

```
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
```

El ovector en este ELAM es sug_elam_out_sidebnd_no_repuesto_vec.ovector_idx: **0x98**, que sabemos por el "hal l2 port gpd", se asigna a la interfaz correcta en la LC:

```
=====
```

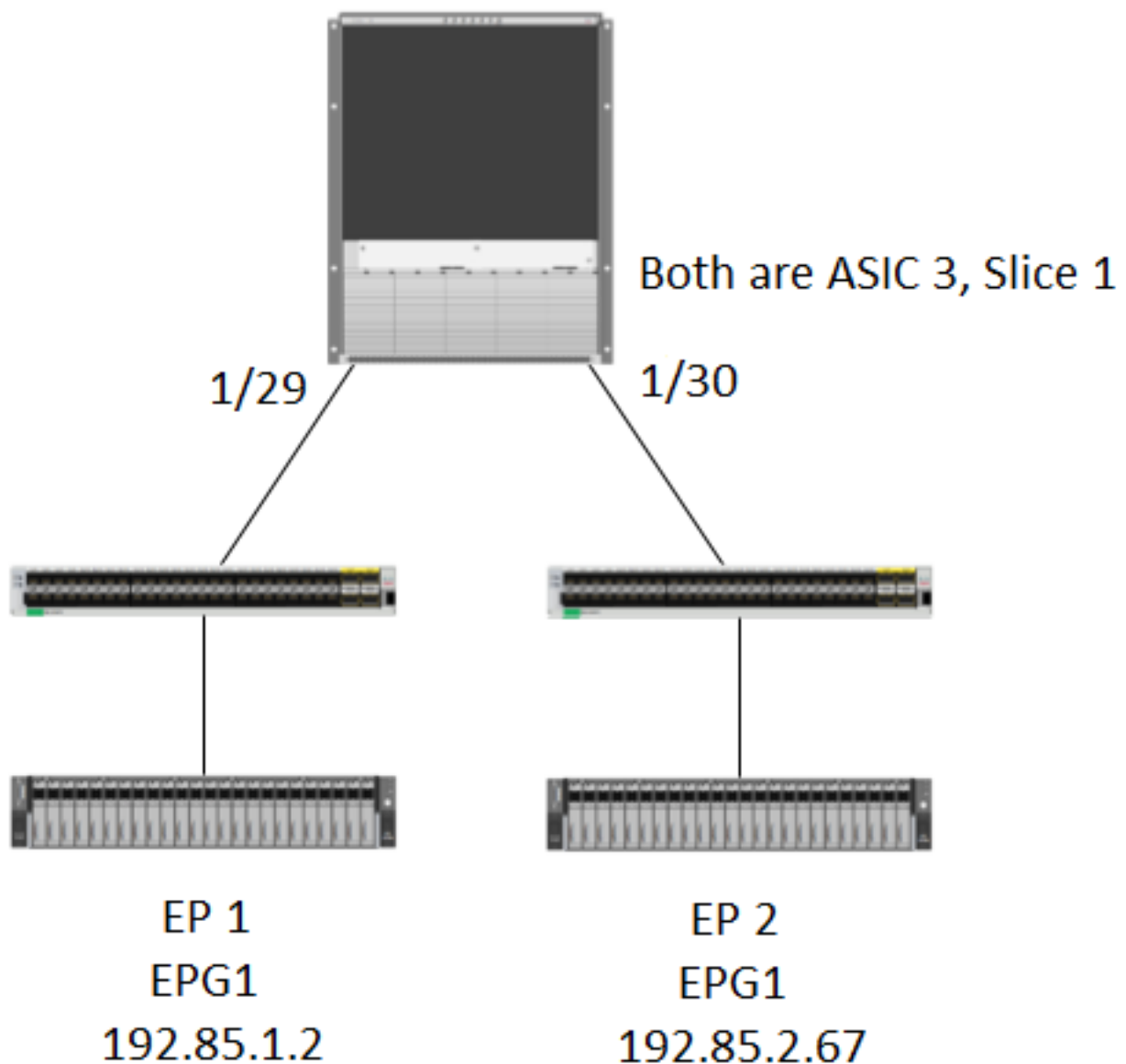
										Reprogram																
Rep	Uc				Uc																					
NI Vif	RwV	Ing	Egr	I PC	Pc	V R	PROF	H	L	R	I	R	D	R	U	U	X	L	Xla	Ovx	N					
IfId	Ifname	P	Cfg	MbrID	As	AP	Sl	Sp	Ss	Ovec	S	P	P	S	P	Sp	Sp	C	M	L	3	Idx	Idx	L3		
L3 Tid	Tid	Lbl	Lbl	S	V	ID	I																			
1f5	SpInBndMgmt	0	9de	1a	0	0	0	0	0	0	0	0	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	1	b	b	1	1	
D-f3	D-61	100	0	0	0	1	0																			
1a081000	Eth2/2	0	9b	22	0	d	0	c	18	18	1	0	0	0	0	0	0	0	0	0	1	c	c	1	1	
D-1ee	D-30b	100	0	0	0	1	0																			
1a084000	Eth2/5	0	9e	1e	0	3d	1	14	28	a8	1	0	0	0	0	0	0	0	0	0	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	1	e	e	1	1	
D-87	D-184	100	0	0	0	1	0																			
1a086000	Eth2/7	0	a0	26	0	35	1	c	18	98	1	0	0	0	0	0	0	0	0	0	1	d	d	1	1	D-
1d0	D-357	100	0	0	0	1	0																			
1a088000	Eth2/9	0	a2	20	1	d	0	c	18	18	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D-3ea	D-1a9	100	0	0	0	1	0																			

```
=====
```

Ethernet 2/7 es la interfaz que se conecta a la hoja 5.

Escenario adicional: Obtención de un vector que no está en la salida "hal internal-port pi"

Topología



Lógica

Hay algunos escenarios donde capturamos un paquete que no tiene un Ovector en la tabla "**show platform internal hal I2 internal-port pi**". En el siguiente escenario, realmente estamos capturando el paquete que regresa de FM, así que necesitamos mirar una tabla diferente para ver qué puerto del panel frontal está seleccionando el paquete.

Tenga en cuenta que la topología anterior es un entorno completamente diferente donde se aprende el tráfico de tránsito (sin ruteo de proxy). 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

@module-1(DBG-elam-insel13)# report | grep ovector

sug_elam_out_sidebnd_no_spare_vec.ovector_idx: 0xA0 <<<<<<<<<<<<<<<<<<<<< now we look for this in the "hal internal-port pi" command

@module-1# show platform internal hal l2 internal-port pi

No sandboxes exist

Num. of Sandboxes: 1

Legend:

IfId:	Interface Id	IfName:	Interface Name
As:	Asic	AP:	Asic Port
Sl:	Slice	SP:	Slice Port
Ss:	Slice SrcId	Ovec:	Ovector
UcPcCfgId:	Uc Pc CfgId	Lb Mbrid:	LB MbrId

Sandbox_ID: 0, BMP: 0x0

Internal Port Count: 24

=====

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

entry that matches 0xA0

@module-1# show platform internal hal l2 port gpd

Legend:

<snip>

Sandbox_ID: 0, BMP: 0x0

Port Count: 6

=====

=====

Uc	Uc		Reprogram
		Rep	

Vif	RwV	Ing	Egr	I PC	Pc	V R	PROF	H	L	R I R D	R U U X	L Xla	Ovx	N	NI
IfId	Ifname	P Cfg	MbrID	As	AP	S1	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 0 0
0 D-2d4	D-3e1	0 0	0 0	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	0 0 0	0 0 0
1a000000	Eth1/1	0 1b	1c	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	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
1 D-13b	D-33b	500 0	1 0	3 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
1a01c000	Eth1/29	0 37	1e	3 3d 1	14 28 a8	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 0 0 0 0 0	0 0 0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
1 D-3f2	D-7a	100 0	0 0	2 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
1a01d000	Eth1/30	0 38	20	3 39 1	10 20 a0	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 0 0 0 0 0	0 0 0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
1 D-36e	D-362	100 0	0 0	2 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
1a01e000	Eth1/31	0 39	22	3 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	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 D-273	D-8	100 0	0 0	2 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
1a01f000	Eth1/32	0 3a	24	3 31 1	8 10 90	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 0 0 0 0 0	0 0 0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
1 D-154	D-5d	100 0	0 0	2 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

1/30 es la interfaz phys que se conecta a la hoja 102, verificada por topología, ASIC 3, Slice 1