

# EX Hardware: Unità di profondità inoltra pacchetti ACI.

## Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Scenari](#)

[2 EP nello stesso EPG/Stessa foglia - Frame commutato](#)

[Topologia](#)

[ELAM](#)

[2 EP in diversi EPG/Same Leaf - Routed Packet](#)

[Topologia](#)

[ELAM](#)

[2 EP in diversi EPG/Foglia diversa - Routed Packet](#)

[Topologia](#)

[ELAM](#)

[1 EP —> L3 in uscita - Flusso indirizzato](#)

[Topologia](#)

[ELAM](#)

[1 EP —> EP remoto o SVI - Verifica della colonna vertebrale](#)

[Topologia](#)

[Logica](#)

[IP sintetico](#)

[Fabric Module ELAM](#)

[Scenario aggiuntivo: Recupero di un ovetto non presente nell'output "hal internal-port pi"](#)

[Topologia](#)

[Logica](#)

## Introduzione

Questo documento descrive i diversi scenari di inoltra usando gli switch ACI "EX" basati su ACI in Application Centric Infrastructure (ACI). Verrà mostrato come verificare che l'hardware sia programmato correttamente e stiamo inoltrando i pacchetti agli endpoint di destinazione corretti (EP) nei Gruppi di endpoint appropriati (EPG).

## Prerequisiti

## Requisiti

Nessun requisito specifico previsto per questo documento.

## Componenti usati

Le informazioni di questo documento si basano sulle seguenti versioni hardware e software:

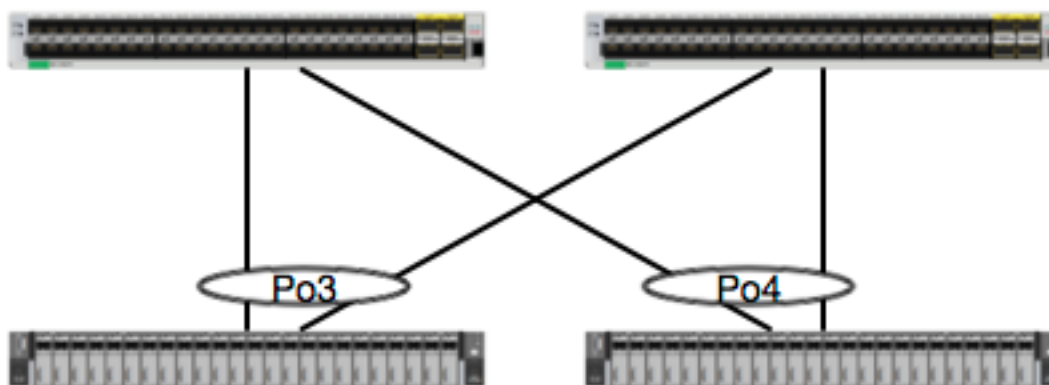
- Un fabric ACI costituito da due switch Spine e due switch Leaf che utilizzano hardware EX
- Un host ESXi con due uplink diretti a ciascuno degli switch foglia
- Nexus 5000 Dispositivo che opera come router.
- Controller APIC (Application Policy Infrastructure Controller) utilizzato per la configurazione iniziale

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

## Scenari

### 2 EP nello stesso EPG/Stessa foglia - Frame commutato

#### Topologia



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

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

Data questa topologia, il flusso da EP1 a EP2 è un flusso L2 e deve essere attivato localmente su qualsiasi foglia in cui si trovi il traffico di origine. La prima cosa da controllare con i flussi di layer 2 (L2) è la tabella degli indirizzi MAC per determinare se e dove lo switch ha ricevuto i frame:

```
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
```

Per visualizzare la vlan di incapsulamento, è possibile controllare anche il database 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/                               Encap           MAC Address       MAC Info/
Interface                           VLAN            IP Address        IP 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/                               Encap           MAC Address       MAC Info/
Interface                           VLAN            IP Address        IP 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
```

Sappiamo che la FD\_VLAN 30 corrisponde, ma possiamo sempre convalidare la mappatura nel software:

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

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

E naturalmente, possiamo controllare l'hardware per assicurarci che la VLAN 30 sia mappata alla VLAN 2268 come incapsulamento del pannello anteriore.

```
leaf4# vsh_lc
```

```
module-1# show system internal eltmc info vlan 30
```

```
      vlan_id:           30   :::      hw_vlan_id:           22
      vlan_type:         FD_VLAN  :::      bd_vlan:              28
      access_encap_type:  802.1q  :::      access_encap:         2268
      fabric_encap_type:  VXLAN   :::      fabric_encap:         11960
      sclass:            32778  :::      scope:                11
      untagged:          0
      access_encap_hex:  0x8dc  :::      fabric_enc_hex:       0x2eb8
      pd_vlan_ft_mask:   0x8
      fd_learn_disable:  0
      qos_class_id:      0   :::      qos_pap_id:           0
      qq_met_ptr:        25  :::      ipmc_index:           0
      ingressBdAclLabel: 0   :::      ingBdAclLblMask:     0
      egressBdAclLabel:  0   :::      egrBdAclLblMask:     0
```

```

qos_map_idx:          0   :::      qos_map_pri:          0
qos_map_dscp:        0   :::      qos_map_tc:          0
vlan_ft_mask:        0xe30
hw_bd_idx:           0   :::      hw_epg_idx:         11267
intf_count:          2   :::      glbl_scp_if_cnt:    2

```

<SNIPPED>

Dato che gli EP vengono appresi nel software, possiamo anche verificare che l'hardware abbia programmato le informazioni L2 di questi EP. Nel nuovo hardware è presente Hardware Abstraction Layer (HAL), che rappresenta lo stato software dell'hardware. Il compito di HAL è di prendere una richiesta di programmazione software e spingerla all'hardware.

Per visualizzare le informazioni sull'hardware L2 relative a un endpoint, è possibile esaminare la tabella L2 in HAL per gli indirizzi mac specificati:

```

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 1 0 0

```

Ora che abbiamo mappato l'hardware, facciamo un ELAM e vediamo dove dovrebbe andare il pacchetto.

## 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

```

Perfetto, Leaf4 ha ricevuto il frame su Asic 0 Slice 1. Con ELAM sul nuovo hardware, c'è un nuovo campo che è molto importante per la risoluzione dei problemi: **ovector\_idx**. Questo indice è l'indice delle porte fisiche da cui il frame/pacchetto deve essere inoltrato. Una volta ottenuto **ovector\_idx**, è possibile utilizzare questo comando per individuare la porta a cui è associato:

```

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

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
Vif Tid:  Vif Tid
RwV Tid:  RwVif Tid
Ing Lbl:  Ingress Acl Label
Egr Lbl:  Egress Acl Label

```

UM: UCPcMbr  
 PROF ID: Lport Profile Id  
 VS: VifStateTable  
 Install  
 RV: Rw VifTable  
 Num. of Sandboxes: 1

Reprogram:  
 HI: LportProfile Hw

Sandbox\_ID: 0, BMP: 0x0  
 Port Count: 8

```

=====
=====
| Rep |          Uc   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
- -      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   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
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 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 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
D-350 -    400 0     0 0   1 0
  
```

Lo switch ritiene che il pacchetto debba essere inoltrato dall'interfaccia Ethernet 1/32. L'indirizzo MAC è stato indicato nell'indirizzo IP4?

```

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
  
```

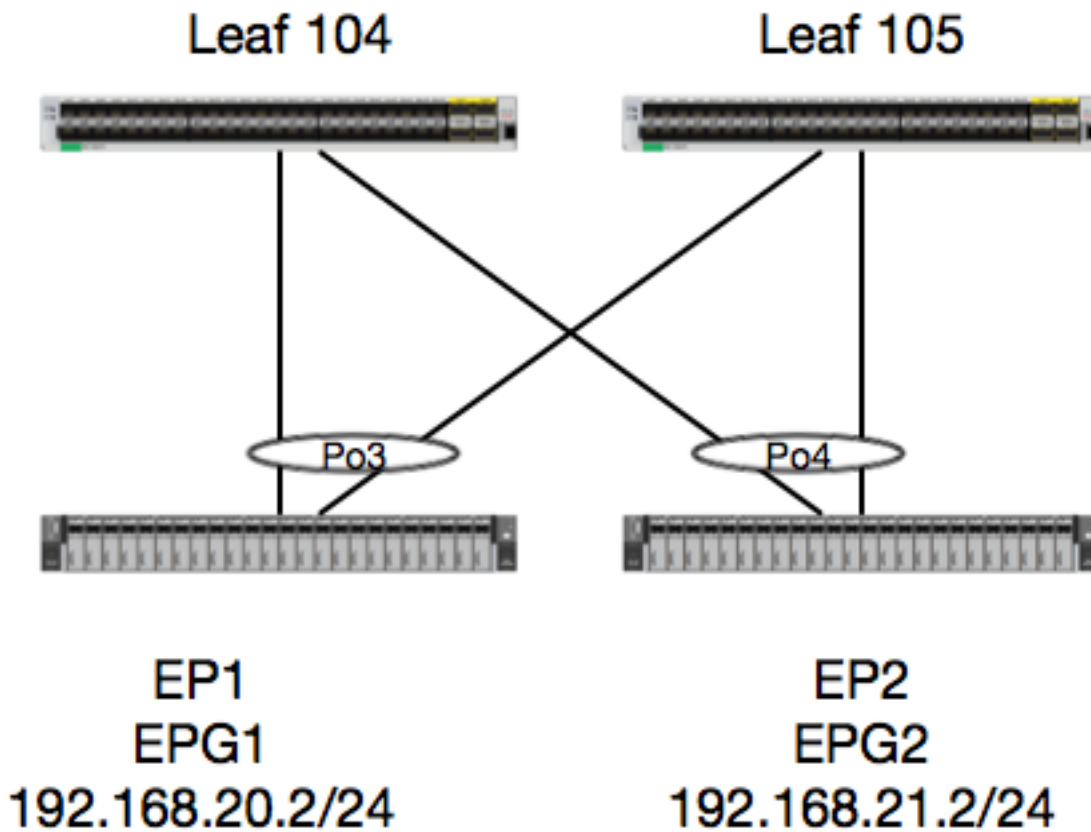
```

-----
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)
  
```

Sì, quindi il pacchetto verrà inoltrato dall'interfaccia 1/32 all'host di destinazione.

## 2 EP in diversi EPG/Same Leaf - Routed Packet

Topologia



In questo esempio, verrà tracciato il flusso di un pacchetto da EP1 a EP2, dove sono presenti sulla stessa coppia di foglia vPC. I due EP sono in differenti EPG utilizzando BD differenti.

La prima cosa da fare è controllare il database del Parlamento europeo per vedere se abbiamo appreso i dati del Parlamento:

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

```
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/
-------	-------	-------------	-----------





```

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 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 tabella HAL Layer3 (I3) è molto utile in quanto fornisce informazioni sulla VLAN/porta per gli EP I3 appresi. Poiché la destinazione è un Po4, il pacchetto deve essere inoltrato da qualsiasi porta del Po4.

Facciamo funzionare un ELAM e vediamo cosa otteniamo!

## 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

```

Bene, abbiamo attivato il pacchetto, e abbiamo scoperto che "ovector\_idx" è 0x9E. L'indice ovector è l'indice dell'interfaccia fisica in uscita da cui il pacchetto deve essere inoltrato. Vediamo quale porta ha quell'indice:

```

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
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 |
| Rep |      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 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
D-350 -        400 0      0 0    1    0

```

Dovremmo inviarlo alla porta 1/32, giusto?

```

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

```

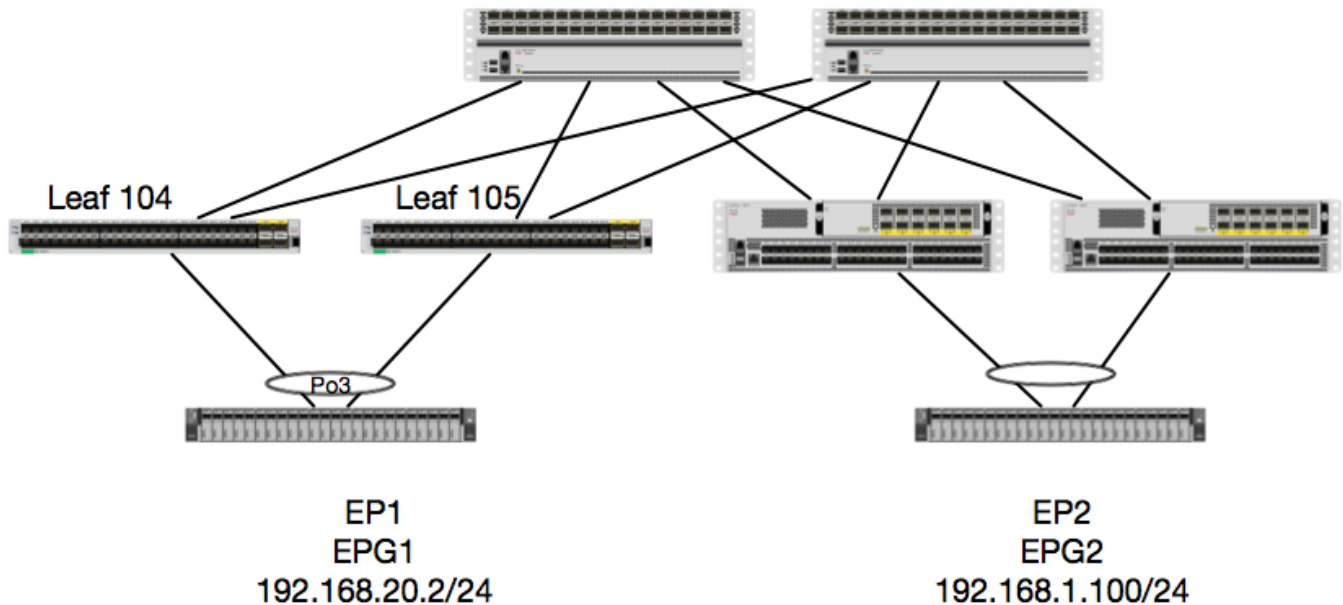
Channel

Channel	Port	Interface	Protocol	Peer Port
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	<b>Po4(SU)</b>	<b>Eth</b>	<b>LACP</b>	<b>Eth1/32(P)</b>

Sì, è corretto.

## 2 EP in diversi EPG/Foglia diversa - Routed Packet

### Topologia



In questo esempio, seguiremo il flusso di un pacchetto da EP1 a EP2 dove EP1 esiste su una coppia di vPC EX ed EP2 esiste su una coppia di Leaf vPC di generazione 1 remota. I due EP sono in differenti EPG utilizzando BD differenti.

Di nuovo, controlliamo dove vengono appresi gli 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 Domain	Encap VLAN	MAC Address IP Address	MAC Info/ IP Info
30 po3	vlan-2268	0050.56a5.fccc	LV
Joey-Tenant:Joey-Internal po3	vlan-2268	192.168.20.2	LV

```
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

```

Verifichiamo ora ciò che l'hardware ha programmato:

```

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 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 1 0 -
L3 - 00:0c:0c:0c:0c:0c Tunnel2    Tunnel2 - 0.0.0.0
Joey-T*ernal2 Pl 192.168.3.1        1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
Joey-T*ternal Pl 192.168.20.1       1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
Joey-T*ternal Pl 192.168.20.2      800a 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 -
L2 BD-28 00:50:56:a5:fc:cc - Po3 FD-30 -
Joey-T*ternal Pl 192.168.21.1        1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
Joey-T*ternal Pl 192.168.21.2       800c 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 -
L2 BD-7 00:50:56:a5:0c:11 - Po4 FD-8 -
Joey-T*ternal Pl 2001:0:0:100::1     1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0

```

L'hardware ritiene che l'EP esista sul tunnel 2. Qual è la destinazione del tunnel 2?

```

module-1# show system internal eltc info interface tunnel2

```

```

IfInfo:

```

```

interface:      Tunnel2  :::      ifindex:      402718722
iod:            66      :::      state:        up
Mod:            0      :::      Port:         0
Tunnel Index:  0      :::      Tunnel Dst ip: 0xc0a87843
Tunnel Encap:  ivxlan  :::      Tunnel VPC Peer: 0
Tunnel Dst ip str: 192.168.120.67  :::      Tunnel ept:    0x1

```

```

[SDK Info]:

```

```

tunnl_name:
vrf_id:         2      :::      if_index:     0x18010002
hwencapidx:    0      :::      encapsype:    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

```

Poiché la destinazione esiste al di fuori di un vPC, tale IP di destinazione deve essere l'IP virtuale vPC dei fogli di lavoro remoti. Controlliamo una foglia remota e vediamo:

```
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
```

Perfetto, così ha imparato l'EP di destinazione dalla coppia di vPC remoti. Vediamo cosa rileva ELAM e verifichiamo che stiamo inoltrando il pacchetto correttamente:

## 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
```

Ora, con le destinazioni remote su hardware EX, ci sono 2 valori ELAM che sono molto importanti quando si risolvono i problemi di flusso dei pacchetti. L'ovector\_idx come prima e 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
```

Sull'hardware EX, abbiamo la capacità di guidare la porta di destinazione dalla quale il pacchetto deve essere inoltrato. In precedenza, di solito abbiamo controllato l'Encrypt Ix e verificato che la destinazione Ix fosse il tunnel corretto. Qui possiamo verificare quale porta è mappata 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
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 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| 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 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 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 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 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 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 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 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 0 0 1 5 3 2 2
D-350 -      400 0      0 0    1 0

```

L'interruttore pensa di doverlo inoltrare al dorso sull'interfaccia Eth1/49. Ma come possiamo verificare che l'encap sia corretto?

Per prima cosa dobbiamo esaminare le informazioni hardware sul tunnel. A tale scopo, eseguire il comando HAL seguente:

```

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
IL:          Is Local
P6:          Proxy for V6
II:          Is Ingress Only
C OBD:       Copy Service Outer Bd
NBT:         Next Base Type E: ECMP N: Next-Hop
NH cnt:      Next Hop Count
Vrf Name:    Vrf Name
Mac:         Mac
L3IfName:    L3 If Name
L2IfName:    L2 If Name

IVP:         Is VPC Peer
P4:          Proxy for v4
PM:          Proxy for Mac
IC:          Is Copy Service
U D:         Use DF
NB Id:       Next Base Id
VrfId:       Vrf Id
IP:          IP Address
L3 IfId:     L3 IfId
L2 IfId:     L2 IfId

```

Num. of Sandboxes: 1

Sandbox\_ID: 0, BMP: 0x0  
Remote Tep Count: 15

```

=====
=====
=====

```

NH	Vrf	E	T Lid	VrfId	Name	IP	Enc	I P P P I I C				U B B		
								L3	L3	L2	L2	D	T	Id
Cnt	VrfId	Name	IP	Mac	IfId	IfName	IfId	IfName						

```

=====
=====
=====
18010002 Tunnel2 I 3005 2 overlay-1 192.168.120.670 0 0 0 0 0 0 0 1 0 E 2
2 2 overlay-1 0.0.0.0 0d:0d:0d:0d:0d:00 1a030001 Eth1/49.1 1a030000 Eth1/49
9
2 overlay-1 0.0.0.0 0d:0d:0d:0d:0d:00 1a031002 Eth1/50.2 1a031000 Eth1/50
0

```

Questo output ci dà alcuni valori che ci interessano:

**lfd:** ID interfaccia allocato al tunnel

**IP - L'IP della destinazione.** Deve corrispondere a ELTMC.

**L3 lfd:** le interfacce di layer 3 che lo switch può utilizzare per inoltrare alla destinazione appropriata.

Una volta conosciuto l'lfld, possiamo verificare che l'encap ricevuto nell'elam corrisponda alla destinazione del tunnel:

```

module-1(DBG-TAH-elam-insel9)# show platform internal hal tunnel rtep apd
Non-Sandbox Mode
LEGEND:
-----
ifId:      Interface Id
HwVrfId:   Hardware Vrf Id
BDXlate:   Egress BDXlate
RwEncapIdx: Rw Encap Index
Num:       Number of hops
L2 Index:  L2 Index

IP:        IP address
SrcTepIdx: Source Tep 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

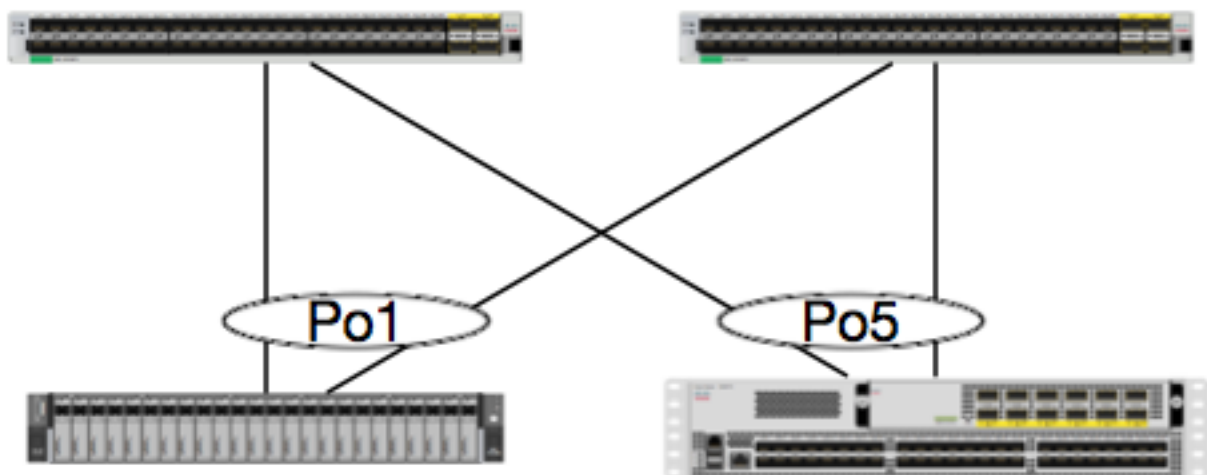
```
=====
=====
ifId      IP          HwVrfId BDXlate SrcTepIdx DstInfoIdx RwEncapIdx ECMPIdx  ECMPMbrIdx Num
L2Index  RwdmacIdx
=====
=====
18010002 192.168.120.67 2        1        3a9a      3005      6          0          0          2
1a030000 0              <---- RwEncapIdx is 6! Same as the "encap_idx"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

1a031000 1

Questo tunnel ha un RwEncapIdx (Re-Write Encap Index) di 6, che è quello che è stato visualizzato nel flusso.

## 1 EP → L3 in uscita - Flusso indirizzato

### Topologia



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

**N5K -OSPF**  
**100.100.100.100/32**

In questo esempio, verrà tracciato il flusso di un pacchetto da EP1 che invia un messaggio ICMP a un loopback su un N5K che esegue OSPF. N5K è connesso tramite un L3Out sulla stessa coppia di switch EX.

Poiché all'inizio di questo documento è stata verificata la programmazione EP locale, supponiamo che tale programmazione sia stata appresa correttamente nell'hardware e proseguiamo con la verifica del percorso.

Per prima cosa, controllare lo stato OSPF e la tabella di routing:

```
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
```

Sappiamo quindi che la tabella di routing mostra l'hop successivo come il 5K alle 10.10.27.3. Buon inizio, ma come possiamo verificare quale hardware ha?

Controlliamo prima la tabella di adiacenza nell'hardware per assicurarci che ARP sia risolto a 10.10.27.3 e che sia programmato con l'interfaccia corretta:

```
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
```

Gli indirizzi MAC corrispondono a quelli dei 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
```

Sulle piattaforme EX, esiste un "hw\_vrf\_idx" assegnato a un VRF. Questo indice verrà utilizzato come riferimento quando si verifica la programmazione hardware. Ecco l'indice:

```
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
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
```

```
::::
```

Dopo aver rilevato l'adiacenza, HAL dovrebbe programmare una rotta. Per verificarlo, usare il comando seguente:

```
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
|

```

```

-----
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
- Trie ----->|<Dleft Trie>|
| VRF |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|N| NB-ID | NB Hw | PID | FPID/| TBI |TRO|Ifindex|CLSS|CLP| AI |SH|DH| Flags |      |      |      |      |      |      |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|B|      | Idx  |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
- DLEFT ----->|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|N| NB-ID | NB Hw |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|B|      | Idx  |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
- TCAM ----->|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|N| NB-ID | NB Hw |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|B|      | Idx  |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
-----

```

```

|Sandbox_ID: 0 Asic Bitmap: 0x0
-----

```

```

module-1# show platform internal hal l3 routes | egrep 100.100.100.100
| 4612|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
6010|A|      | 7567| 802e| 186a| 1/ 2|      | 10| 0|      | f| 3| 0| 0| 0|spi,dpi

```

Questo output ci fornisce informazioni sul percorso dell'hop successivo. 4612 è l'hw\_vrf\_idx del jr:sb VRF. Per poter verificare l'hop successivo, il "NB Hw Idx" in TCAM verrà utilizzato in base alla tabella seguente:

```

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

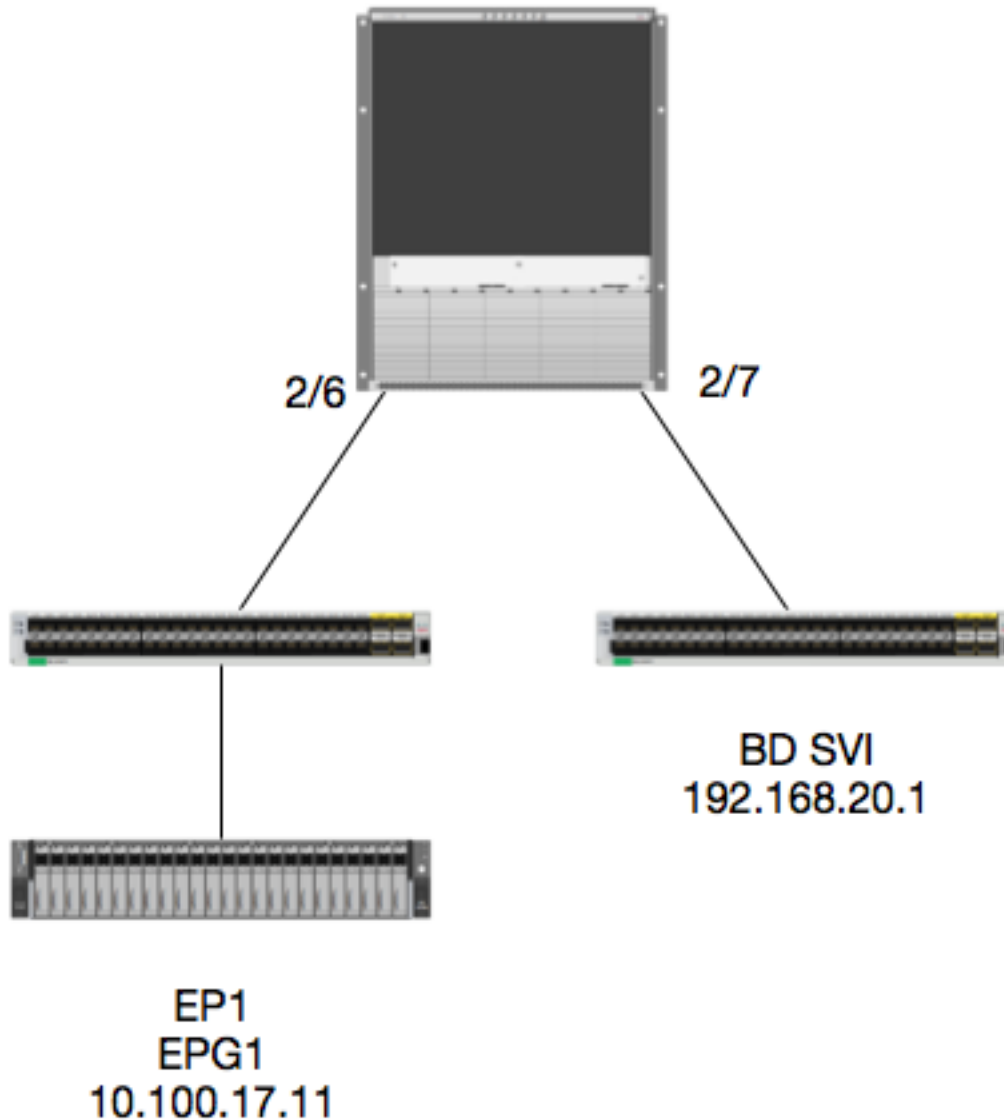
```



```
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 - Verifica della colonna vertebrale

### Topologia



### Logica

In questo esempio, verrà tenuta traccia del flusso di un pacchetto da EP1 destinato a un'interfaccia virtuale commutata BD remota (SVI). Lo scopo di questo esempio è verificare Spine Forwarding per assicurarsi che il pacchetto venga inviato alla foglia corretta. Supponiamo che il pacchetto sia stato inviato al Proxy della colonna vertebrale in entrata.

Sul dorso, verifichiamo prima il protocollo COOP (Council of Oracles Protocol) per l'IP di destinazione, poiché il pacchetto viene inviato al proxy dorso per una ricerca:

```
calol-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

```

### Verifica dell'indirizzo TEP della foglia:

```

spinel# acidiag fnvread | grep 10.0.224.88
  105      1      cal01-leaf5      FDO20160TPS      10.0.224.88/32      leaf
active    0

```

Poiché sappiamo che il pacchetto sta entrando nel dorso sul modulo 2, porta 6, possiamo collegarci al modulo 2 e vedere il layout della porta.

```

spinel# vsh
Cisco iNX-OS Debug Shell
This shell should only be used for internal commands and exists
for legacy reasons. User should use ibash infrastructure as this
will be deprecated.
cal01-spinel# attach module 2
Attaching to module 2 ...
To exit type 'exit', to abort type '$.'
No directory, logging in with HOME=/
Bad terminal type: "xterm-256color". Will assume vt100.
Cisco iNX-OS Debug Shell
This shell should only be used for internal commands and exists
for legacy reasons. User should use ibash infrastructure as this
will be deprecated.
Loading parse tree (LC). Please be patient...
module-2#

```

```

module-2# show platform internal hal 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
S1:	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
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	0	0	0	0
D-f3	D-61	100	0	0	0	1	0																		
1a081000	Eth2/2	0	9b	22	0	d	0	c	18	18	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D-1ee	D-30b	100	0	0	0	1	0																		
1a084000	Eth2/5	0	9e	1e	0	3d	1	14	28	a8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D-19a	D-2ee	100	0	0	0	1	0																		
1a085000	<b>Eth2/6</b>	0	9f	24	0	39	1	10	20	a0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D-87	D-184	100	0	0	0	1	0																		
1a086000	Eth2/7	0	a0	26	0	35	1	c	18	98	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
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 2/6 è l'interfaccia che si connette a Leaf 6 su ASIC 0 SLICE 1

Ora sappiamo su quale ASIC far funzionare il nostro 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

```

Guardando l'ELAM, possiamo trovare l'indice ovettoriale:

Front Panel ELAM drove **sug\_elam\_out\_sidebnd\_no\_spare\_vec.ovector\_idx: 0xB8**

Come si esegue il mapping di 0xb8 a una porta? Poiché sappiamo che il pacchetto deve essere inviato a un modulo fabric (FM) per una ricerca, possiamo esaminare la mappatura delle porte interne per trovare il FM di destinazione:



```
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
<b>96</b>	-	<b>0</b>	<b>49</b>	<b>1</b>	<b>20</b>	<b>38</b>	<b>b8</b>	<b>0</b>	<b>7</b>
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

```
=====
```

Utilizzando ASIC0 / Ovec B8, otteniamo MbrId 0x7, Slice non importa.

Questo MbrId è l'interfaccia dell'USD che viene mappata a un'interfaccia su un FM. Tenere presente che MbrId è in formato esadecimale e deve essere convertito in decimale.

Possiamo scoprire quale FM guardando le interfacce USD e ispezionando Port 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 base dello "slot" è 0, mentre la numerazione FM è 1, quindi è necessario aggiungere 1 al numero indicato. Ciò significa che il pacchetto deve essere inviato alla FM 23.

## IP sintetico

Come in Alpine, esiste un IP sintetico usato come indirizzo IP esterno per determinare l'hash per la ricerca COOP. Per risolvere questo problema, è necessario eseguire questo comando e utilizzare il comando grep per l'indirizzo IP DST 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
```

Questo ci mostra che 1.203.211.185 è il nostro IP sintetico. Sulla base di questo, possiamo anche impostare "Outer DST IP" sul nostro elam FM per essere questo. Dovremmo attivare la radio FM:

## Fabric Module ELAM

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

```
module-23(DBG-TAH-elam-insel13)# stat
ELAM STATUS
=====
Asic 0 Slice 0 Status Armed
Asic 0 Slice 1 Status Armed
Asic 0 Slice 2 Status Triggered <----- Triggered on SLICE 2
Asic 0 Slice 3 Status Armed
Asic 0 Slice 4 Status Armed
Asic 0 Slice 5 Status Armed
```

È ovvio che è necessario eseguire il dump del report completo, ma esaminiamo l'ovector\_idx per il pacchetto che è stato attivato:

lac\_elam\_out\_sidebnd\_no\_spare\_vec.ovector\_idx: **0x20** ← Indice del settore utilizzato nel comando seguente

Come facciamo a capire quale interfaccia ha questo vettore? In FM, eseguire quanto segue:

**\*\* A causa del bug [CSCvf42796](#) , aggiungere tutti i comandi FM con "| no-more". In caso contrario, alcune voci potrebbero non essere visualizzate nell'output finale.**



```

module-2(DBG-TAH-elam)# trigger init in-select 13 out-select 0
module-2(DBG-TAH-elam-insell13)# set inner ipv4 src_ip 10.100.17.11 dst_ip 192.168.20.1
module-2(DBG-TAH-elam-insell13)# start
stat
module-2(DBG-TAH-elam-insell13)# stat
ELAM STATUS
=====
Asic 0 Slice 0 Status Armed
Asic 0 Slice 1 Status Armed

```

```

module-2(DBG-TAH-elam-insell13)# stat
ELAM STATUS
=====
Asic 0 Slice 0 Status Triggered <---- Packet triggered from FM
Asic 0 Slice 1 Status Triggered <---- Packet triggered from Front Panel

```

L'ovettore su questo ELAM è sug\_elam\_out\_sidebnd\_no\_spare\_vec.ovector\_idx: **0x98**, che si conosce dal "hal l2 port gpd", è mappato all'interfaccia corretta sul LC:

```

=====
=====

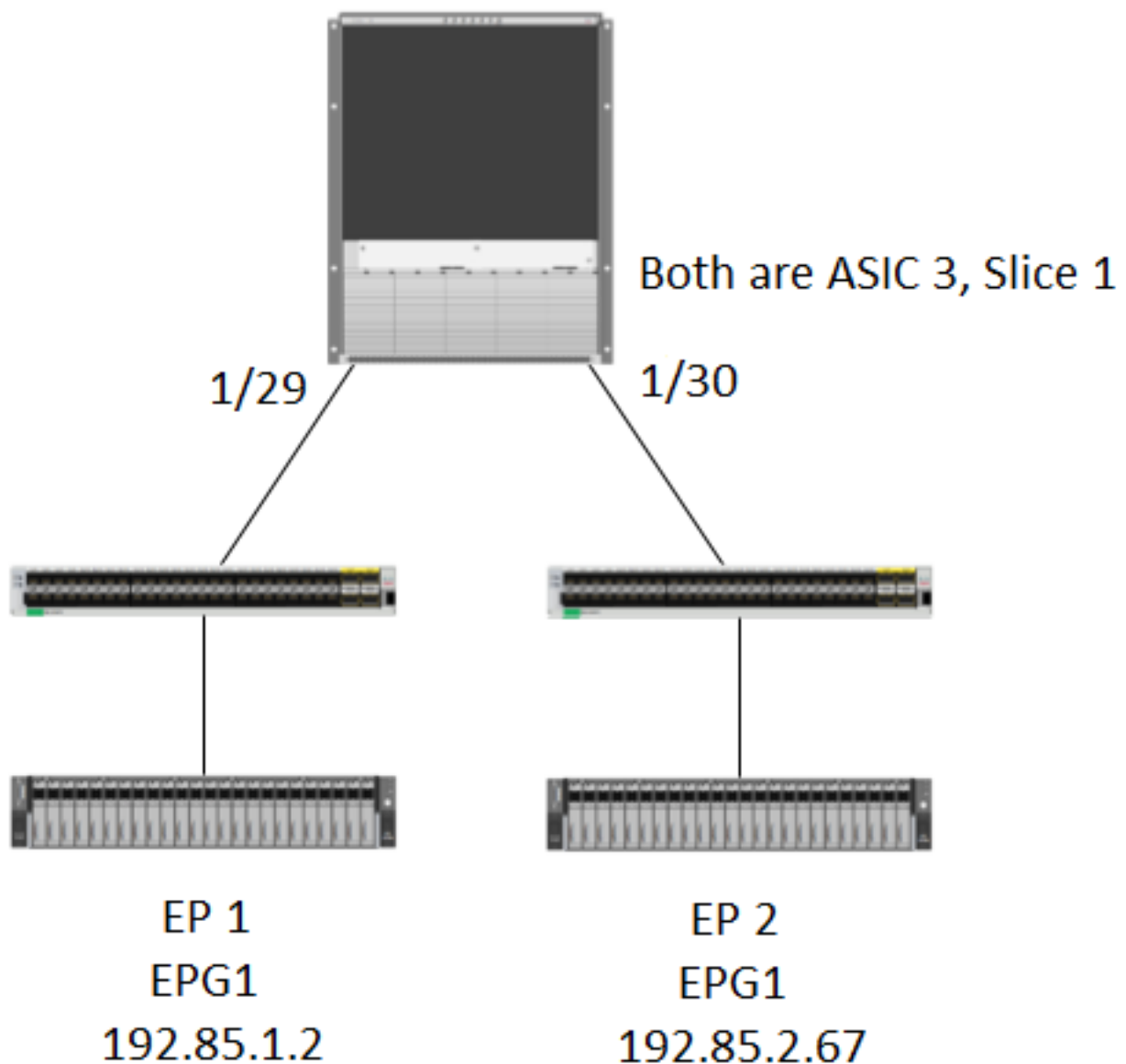
```

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	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
D-2d4	D-3e1	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
D-f3	D-61	100	0	0	1	0												
1a081000	Eth2/2	0	9b	22	0	d	0	c	18	18	1	0	0	0	0	0	0	0
D-1ee	D-30b	100	0	0	1	0												
1a084000	Eth2/5	0	9e	1e	0	3d	1	14	28	a8	1	0	0	0	0	0	0	0
D-19a	D-2ee	100	0	0	1	0												
1a085000	Eth2/6	0	9f	24	0	39	1	10	20	a0	1	0	0	0	0	0	0	0
D-87	D-184	100	0	0	1	0												
1a086000	<b>Eth2/7</b>	0	a0	26	0	35	1	c	18	<b>98</b>	1	0	0	0	0	0	0	0
1d0	D-357	100	0	0	1	0												
1a088000	Eth2/9	0	a2	20	1	d	0	c	18	18	1	0	0	0	0	0	0	0
D-3ea	D-1a9	100	0	0	1	0												

Ethernet 2/7 è l'interfaccia che si connette a Leaf 5.

## Scenario aggiuntivo: Recupero di un ovetto non presente nell'output "hal internal-port pi"

### Topologia



## Logica

In alcuni scenari viene intercettato un pacchetto che non contiene un ovector nella tabella "**show platform internal hal I2 internal-port pi**". Nello scenario seguente, stiamo effettivamente catturando il pacchetto che torna dalla FM, quindi dobbiamo esaminare una tabella diversa per vedere quale porta del pannello anteriore il pacchetto sta selezionando.

Si noti che la topologia riportata sopra è un ambiente completamente diverso in cui viene appreso il traffico di transito (senza routing proxy). Il modulo è 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
```

```
=====
                                         UcPc Lb
IfId   IfName                         As AP Sl SP Ss Ovec CfgId MbrId
=====
```

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

<<<<<<<< we cant find an  
entry that matches 0xA0

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

```
Legend:
-----
<snip>
```

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

```
=====
=====
               Uc   Uc               |   Reprogram
|               |           |   |
|               |           |   |
```

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	Sl	Sp	Ss	Ovec	S	P P P S P	Sp	Sp	C M L	3 Idx	Idx	L3
L3 Tid	Tid	Lbl	Lbl	S V	ID	I											
1f5	SpInBndMgmt	0 9de	1a	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0	0	0
0 D-2d4	D-3e1	0	0	0 0	1	0											
1a000000	Eth1/1	0 1b	1c	0 11	0	10 20 20	1	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	1 1	1	1	1
1 D-13b	D-33b	500 0	1 0	3	0												
1a01c000	Eth1/29	0 37	1e	3 3d	1	14 28 a8	1	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	1 8	8	1	1
1 D-3f2	D-7a	100 0	0 0	2	0												
1a01d000	<b>Eth1/30</b>	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	1 5	5	1	1
1 D-36e	D-362	100 0	0 0	2	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	1 9	9	1	1
1 D-273	D-8	100 0	0 0	2	0												
1a01f000	Eth1/32	0 3a	24	3 31	1	8 10 90	1	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	1 a	a	1	1
1 D-154	D-5d	100 0	0 0	2	0												

1/30 è l'interfaccia phys che si connette alla foglia 102, verificata dalla topologia, ASIC 3, Slice 1