

EX hardware: ACI Packet Forwarding diep duiken.

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Scenarios](#)

[2 EP's in hetzelfde EPG/Dezelfde Leaf - Switched frame](#)

[Topologie](#)

[ELAM](#)

[2 EP's in verschillende EPG/Zelfde Leaf - Routed Packet](#)

[Topologie](#)

[ELAM](#)

[2 EP's in verschillende EPG/Different Leaf - Routed Packet](#)

[Topologie](#)

[ELAM](#)

[1 EP —> L3 uitloop - Routed Flow](#)

[Topologie](#)

[ELAM](#)

[1 EP —> Afstandsbediening EP of SVI - Spinetoetsing](#)

[Topologie](#)

[Logic](#)

[Synthetisch IP](#)

[Fabric-module ELAM](#)

[Extra scenario: Een Ovector verkrijgen die niet in de "hal interne haven pi"-uitgang zit](#)

[Topologie](#)

[Logic](#)

Inleiding

Dit document beschrijft verschillende doorsturen scenario's met behulp van de "EX"-gebaseerde ACI-switches in Application Centric Infrastructure (ACI). Het toont hoe de hardware correct geprogrammeerd is en we verzenden pakketten naar de juiste bestemming Endpoints (EP's) in de juiste Endpoint Group (EPG's).

Voorwaarden

Vereisten

Er zijn geen specifieke vereisten van toepassing op dit document.

Gebruikte componenten

De informatie in dit document is gebaseerd op deze hardware- en softwareversies:

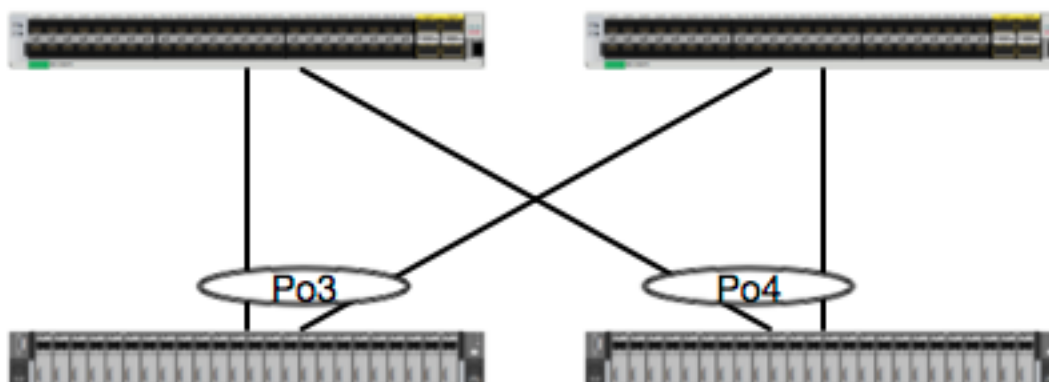
- Een ACI Fabric die uit twee spineswitches en twee Leaf-switches bestaat met EX Hardware
- Een ESXi host met twee uplinks die op elk van de Leaf-switches werken
- Nexus 5000 apparaat dat als router werkt.
- Een APIC-controller (Application Policy Infrastructure Controller) die voor eerste instellingen wordt gebruikt

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u de potentiële impact van elke opdracht begrijpen.

Scenarios

2 EP's in hetzelfde EPG/Dezelfde Leaf - Switched frame

Topologie



EP1
EPG1
0050.56a5.fccc
192.168.20.2/24

EP2
EPG1
0050.56a5.6794
192.168.20.3/24

Gezien deze topologie is de stroom van EP1 naar EP2 een L2-stroming en moet deze lokaal op welk blad dan ook worden geschakeld. Het eerste wat te controleren met Layer 2 (L2) stromen is de hoofdadrestabel om te bepalen of en waar de switch frames ontving:

```
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
```

Om het insluitingsVLAN te zien, kunnen we ook de EP-database controleren:

```
leaf4# show endpoint mac 0050.56a5.fccc
```

```
Legend:
```

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

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

```
calo2-leaf4# show endpoint mac 0050.56a5.6794
```

```
Legend:
```

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

```
+-----+-----+-----+-----+
----+
VLAN/                               Encap           MAC Address       MAC Info/
Interface
Domain                             VLAN            IP Address        IP Info
+-----+-----+-----+-----+
----+
30                                  vlan-2268       0050.56a5.6794  LV
po4
Joey-Tenant:Joey-Internal         vlan-2268       192.168.20.3    LV
po4
```

We kennen de FD_VLAN 30-overeenkomsten, maar we kunnen altijd de mapping in software valideren:

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

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

En natuurlijk kunnen we de hardware controleren om VLAN 30 kaarten aan VLAN 2268 als de insluiting van het voorpaneel te verzekeren.

```
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
      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>

Aangezien het Europees Parlement in software wordt geleerd, kunnen we ook de hardware die de L2-informatie van deze EP's heeft geprogrammeerd, valideren. In de nieuwe hardware is er de Hardware Abstraction Layer (HAL), de softwarefase van de hardware. HAL heeft als taak een softwareprogrammaaanvraag in te dienen en op hardware te duwen.

Om L2 hardwareinformatie over een eindpunt te bekijken, kunnen we de L2 tabel in HAL voor bepaalde mac adressen bekijken:

```

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

```

Nu we de hardware in kaart hebben gebracht, doen we een ELAM en zien waar het pakje moet gaan.

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

```

Geweldig, dus Leaf4 ontving het kader op Asic 0 Slice 1. Met ELAM op de nieuwe hardware is er een nieuw veld dat erg belangrijk is bij het oplossen van problemen: **ovector_idx**. Deze index is de fysieke poortindex waarvan het frame/pakket moet worden doorgestuurd. Zodra je de ovector_idx hebt, kunnen we deze opdracht gebruiken om te zoeken naar welke poort het in kaart brengt:

```

module-1(DBG-TAH-elam-insel6)# show platform internal hal l2 port gpd
Legend:
-----
IfId:           Interface Id           IfName:         Interface Name
I P:           Is PC Mbr           IfId:           Interface Id
Uc PC Cfg:     UcPcCfg Idx           Uc PC MbrId:   Uc Pc Mbr Id
As:           Asic           AP:           Asic Port
Sl:           Slice           Sp:           Slice Port
Ss:           Slice SrcId       Ovec:         Ovector (slice |
srcid)
L S:           Local Slot           Reprogram:
L3:           Is L3
P:           PifTable           Xla Idx:       Xlate Idx
RP:          Rw PifTable         Ovx Idx:       OXlate Idx
IP:          If Profile Table   N L3:         Num. of L3 Ifs
RS:          Rw SrcId Table     NI L3:        Num. of Infra L3 Ifs
DP:          DPort Table       Vif Tid:      Vif Tid
SP:          SrcPortState Table  RwV Tid:     RwVif Tid
RSP:         RwSrcPortstate Table  Ing Lbl:     Ingress Acl Label
UC:          UCPCfg           Egr Lbl:     Egress Acl Label

```

UM: UCPcMbr
 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 | | V | R | | PROF | H | | L | | R | I | R | D | | R | U | U | X | | L | X | l | a | O | v | x | 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 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
-      -      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
D-256 -      800 0    0 1    e    0
1a007000 Eth1/8      0 2e   7     0 10 0 f 1e 1e  1  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
D-84  -      800 0    0 1    30   0
1a01e000 Eth1/31     1 0    2d    0 37 1 e 1c 9c  1  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-      -      0 0    0 1    0    0
1a01f000 Eth1/32  1 0    3d    0 38 1 f 1e 9e  1  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-      -      0 0    0 1    0    0
1a030000 Eth1/49     0 2    1     0 49 1 20 38 b8  1  0 0 0 0 0 0 0 0 0 0 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 0
D-350 -      400 0    0 0    1    0
  
```

De switch denkt dat het pakket moet worden doorgestuurd vanuit interface Ethernet 1/32. Is dat PO4 waar we dat mac-adres hebben geleerd?

```

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
  
```

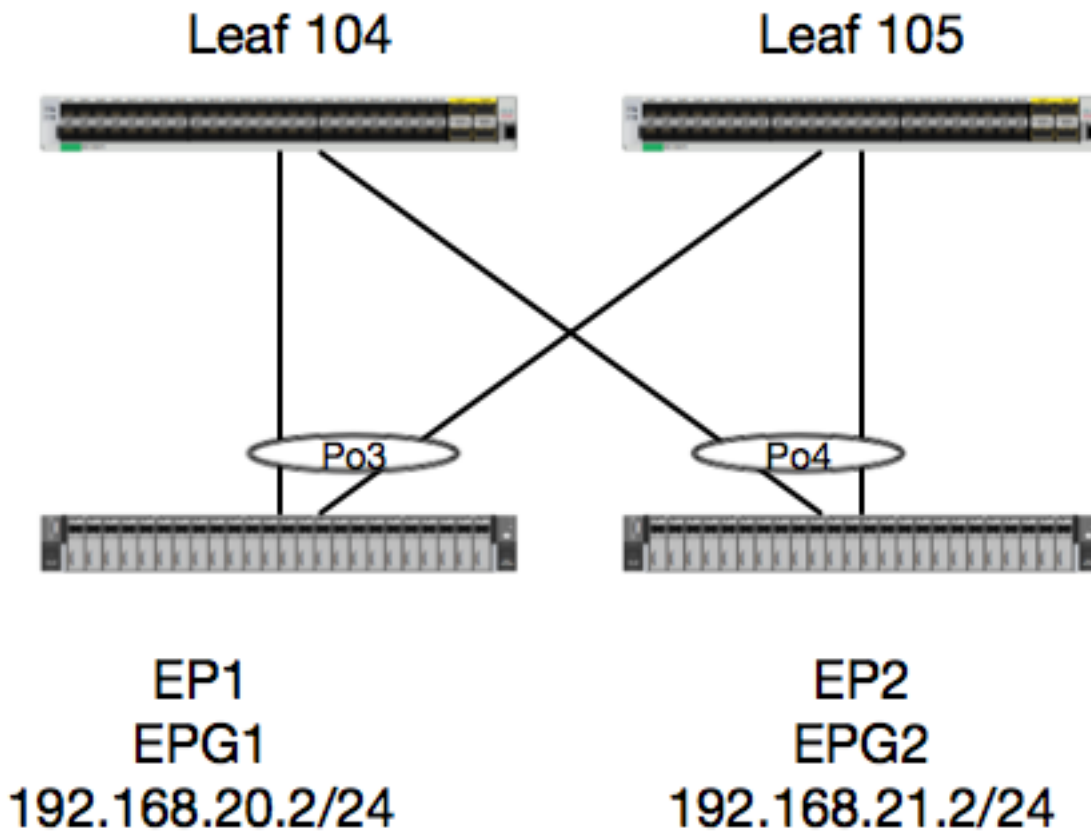
```

-----
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)
  
```

Ja, dus wordt het pakket vanuit interface 1/32 naar de Destination Host verzonden.

2 EP's in verschillende EPG/Zelfde Leaf - Routed Packet

Topologie



In dit voorbeeld volgen we de pakketstroom van een pakket van EP1 naar EP2 waar ze op hetzelfde vPC-bladpaar bestaan. De twee EP's zijn in verschillende EPG's met verschillende BD's.

Het eerste wat we altijd moeten doen is de EP-database controleren om te zien of we de EP's hebben geleerd:

```
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 192.168.21.2
```

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

Aangezien wij het EP hebben geleerd en de informatie over de intellectuele eigendomsverhoudingen kennen, moeten wij de informatie van het EP over de leerstof in hardware kunnen bekijken:

```
leaf4# vsh_lc
module-1# show platform internal hal ep l3 all
LEGEND:
-----
VrfName:          Vrf Name                               T:                Type
(Pl: Physical, Vl: Virtual, Xr: Remote)
EP IP:            Endpoint IP
S Class:          S Class                               Age Intvl:        Age
Interval
S T:              Static Ep                             S E:
Secure EP
L D:              Learn Disable                         B N D:            Bind
Notify Disable
E N D:            Epg Notify Disable                       B E:
Bounce Enable
I D L:            IVxlan Dont Learn                       SPI:
Source Policy Incomplete
DPI:              Dest Policy Incomplete                 SPA:
Source Policy Applied
DPA:              Dest Policy Applied                     DSS:              Dest
Shared Service
IL:              Is Local                               VUB:              Vnid
Use Bd
SO:              SA Only                               EP NH L3IfName:  EP
Next Hop L3 If Name
NHT:              Next Hop Type (L2: L2 Entry L3: L3 Next Hop)   BD Name:          L2 NH
BD Name
EP Mac:           EP Mac                                               L3 IfName:        L3 NH
If Name
L2 IfName:        L2 If Name                                           FD Name:          L2
Entry FD Name
IP:              L3 NH IP

L3 EP Count: 12

=====
=====
B E I S D S D D V EP-NH
N |
Vrf          EP          S      Age  S S L N N B D P P P P S I U S L3
H | BD      EP          L3      L2      FD
Name         T IP          Class Intvl T E D D D E L I I A A S L B O
IfName       T | Name      Mac      IfName  Ifname  Name    IP
=====
=====
common*rewall Pl 10.6.112.1          1      0      1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3      -      00:00:00:00:00:00 -      -      -      0.0.0.0
```



```

I P:          Is PC Mbr                      IfId:        Interface Id
Uc PC Cfg:    UcPcCfg Idx                  Uc PC MbrId: Uc Pc Mbr Id
As:           Asic                          AP:          Asic Port
Sl:           Slice                         Sp:          Slice Port
Ss:           Slice SrcId                   Ovec:        Ovector (slice |
srcid)
L S:          Local Slot                    Reprogram:
L3:          Is L3
      P:      PifTable                      Xla Idx:     Xlate Idx
      RP:     Rw PifTable                   OvX Idx:     OXlate Idx
      IP:     If Profile Table              N L3:        Num. of L3 Ifs
      RS:     Rw SrcId Table               NI L3:       Num. of Infra L3 Ifs
      DP:     DPort Table                  Vif Tid:     Vif Tid
      SP:     SrcPortState Table           RwV Tid:     RwVif Tid
      RSP:    RwSrcPortstate Table         Ing Lbl:     Ingress Acl Label
      UC:     UCPCfg                       Egr Lbl:     Egress Acl Label
      UM:     UCPcMbr                       Reprogram:
PROF ID:      Lport Profile Id              HI:          LportProfile Hw
      VS:     VifStateTable
Install
      RV:     Rw VifTable
Num. of Sandboxes: 1

```

```

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

```

```

=====
| Rep |      Uc   Uc          |      Reprogram      |
|-----|-----|-----|
NI Vif  RwV  Ing  Egr  | V R | PROF H      L |  R I R D   R  U U X | L Xla OvX N
IfId    Ifname  P Cfg  MbrID As AP 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
-      -      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   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
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 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

```

Het lijkt erop dat we de haven 1/32 moeten uitzenden, klopt dat?

```

leaf4# show port-channel summary
Flags:  D - Down          P - Up in port-channel (members)
        I - Individual    H - Hot-standby (LACP only)
        s - Suspended     r - Module-removed
        S - Switched      R - Routed
        U - Up (port-channel)
        M - Not in use. Min-links not met
        F - Configuration failed

```

```
-----
```

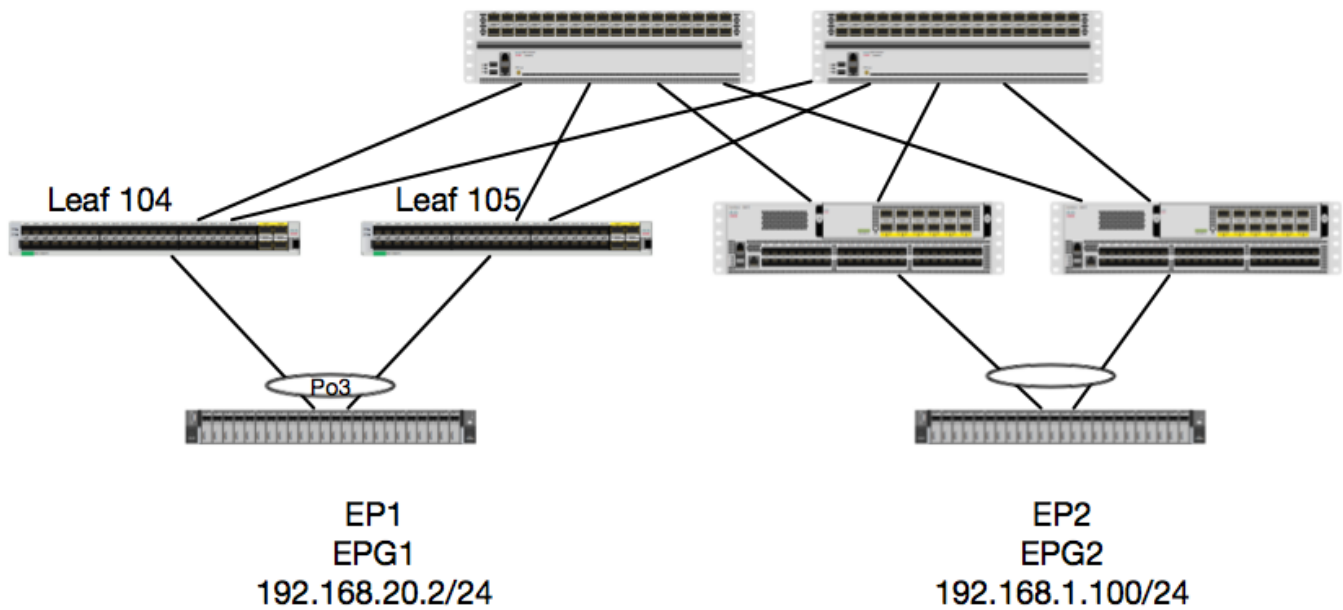
Group	Port-Channel	Type	Protocol	Member Ports
1	Po1 (SU)	Eth	LACP	Eth1/5 (P)
2	Po2 (SU)	Eth	LACP	Eth1/6 (P)
3	Po3 (SU)	Eth	LACP	Eth1/31 (P)
4	Po4 (SU)	Eth	LACP	Eth1/32 (P)

```
-----
```

Ja, dat klopt.

2 EP's in verschillende EPG/Different Leaf - Routed Packet

Topologie



In dit voorbeeld volgen we de pakketstroom van een pakket van EP1 naar EP2 waar EP1 op een EX vPC paar bestaat en EP2 op een ver Generation 1 vPC Leaf paar bestaat. De twee EP's zijn in verschillende EPG's met verschillende BD's.

Laten we eens kijken waar het EP leert:

```
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 192.168.1.100

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

Table with 4 columns: VLAN/Interface, Encap, MAC Address, MAC Info/ IP Info. Includes separator lines and headers.

Joey-Tenant:Joey-Internal 192.168.1.100
tunnel2

Laten we nu controleren wat de hardware heeft geprogrammeerd:

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

Table with columns: N, Vrf, H, Name, IfName, EP, L3, L2, S, Age, S S L N N B D P P P S I U S L3, Class Intvl T E D D D E L I I A A S L B O, Mac, IfName, Ifname, Name, IP. Includes separator lines.

```

common*rewall Pl 10.6.112.1          1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
common*rewall Pl 10.6.114.1          1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
common*rewall Pl 10.6.114.129        1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
common*efault Pl 100.100.101.1        1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
Joey-T*ternal Pl 192.168.1.1          1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
Joey-T*ternal Xr 192.168.1.100      8013 128 0 0 0 1 0 0 0 0 0 0 0 0 1 0 -
L3 - 00:0c:0c:0c:0c:0c Tunnel2    Tunnel2 - 0.0.0.0
Joey-T*ternal2 Pl 192.168.3.1          1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
Joey-T*ternal Pl 192.168.20.1          1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
Joey-T*ternal Pl 192.168.20.2      800a 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 -
L2 BD-28 00:50:56:a5:fc:cc -      Po3 FD-30 -
Joey-T*ternal Pl 192.168.21.1          1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
Joey-T*ternal Pl 192.168.21.2          800c 0 0 0 0 0 0 0 0 0 0 0 0 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

```

Hardware denkt dat het EP bestaat op Tunnel 2. Wat is de bestemming voor Tunnel 2?

```

module-1# show system internal eltmc info interface tunnel2

```

```

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

```

```

[SDK Info]:
tunnl_name:
vrf_id:        2      :::      if_index:     0x18010002
hwencapidx:    0      :::      encapsytype:  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
```

Aangezien de bestemming van een vPC bestaat, zou die Bestemming IP de vPC Virtual IP van de afstandsbediening moeten zijn. Laten we een afgelegen blad bekijken:

```
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
```

Perfect, dus het heeft de Destination EP geleerd van het afgelegen vPC paar. Laten we kijken wat ELAM ziet en verifiëren dat we het pakket correct verzenden:

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
```

Nu, met verre bestemmingen op EX Hardware, zijn er 2 ELAM waarden die zeer belangrijk zijn wanneer u een oplossing vindt voor pakketstroom. Zoals eerder, en de 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
```

Op EX Hardware, hebben wij de mogelijkheid om de deelpoort te besturen waar het pakket moet worden doorgestuurd. Voordien controleerden we meestal de encap idx en controleerden we of de bestemming idx de juiste tunnel was. Hier kunnen we verifiëren welke havenkaarten tot 8B:

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

Legend:

IfId:	Interface Id	IfName:	Interface Name
I P:	Is PC Mbr	IfId:	Interface Id
Uc PC Cfg:	UcPcCfg Idx	Uc PC MbrId:	Uc Pc Mbr Id
As:	Asic	AP:	Asic Port


```
Vxlan I: IVxlan N: NVGRE
VrfId: Vrf Id Vrf Name: Vrf Name
IP: Tunnel's IP
Hw Enc: Hw Encap Idx IVP: Is VPC Peer
IL: Is Local P4: Proxy for v4
P6: Proxy for V6 PM: Proxy for Mac
II: Is Ingress Only IC: Is Copy Service
C OBD: Copy Service Outer Bd U D: Use DF
NBT: Next Base Type E: ECMP N: Next-Hop NB Id: Next Base Id
NH cnt: Next Hop Count VrfId: Vrf Id
Vrf Name: Vrf Name IP: IP Address
Mac: Mac L3 IfId: L3 IfId
L3IfName: L3 If Name L2 IfId: L2 IfId
L2IfName: L2 If Name
```

Num. of Sandboxes: 1

Sandbox_ID: 0, BMP: 0x0

Remote Tep Count: 15

```
=====
=====
=====
```

NH	Vrf	E	Vrf	Hw	V	I	P	P	P	I	I	C	U	B	B
IfId	Ifname	T Lid	VrfId	Name	IP	Enc	P	L	4	6	M	I	C	OBD	D T Id
Cnt	VrfId	Name	IP	Mac	IfId	IfName	IfId	IfName	IfId	IfName	IfId	IfName	IfId	IfName	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 Eth1/4
9
2 overlay-1 0.0.0.0 0d:0d:0d:0d:0d:00 1a031002 Eth1/50.2 1a031000 Eth1/5
0
```

Deze output geeft ons een paar waarden waar we om geven:

AsID - De interface-ID die aan de tunnel is toegewezen

IP - Het IP van de bestemming. Dit moet overeenkomen met ELTMC.

L3 AsID - De Layer 3 interface(s) kan de switch gebruiken om naar de juiste bestemming te sturen.

Zodra we de AsID kennen, kunnen we controleren of het plafond dat we in de elam hebben gekregen overeenkomt met de tunnelbestemming:

```
module-1(DBG-TAH-elam-inse19)# show platform internal hal tunnel rtep apd
Non-Sandbox Mode
LEGEND:
-----
ifId: Interface Id IP: IP address
HwVrfId: Hardware Vrf Id SrcTepIdx: Source Tep Index
BDXlate: Egress BDXlate DstInfoIdx: Destination info index
```


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

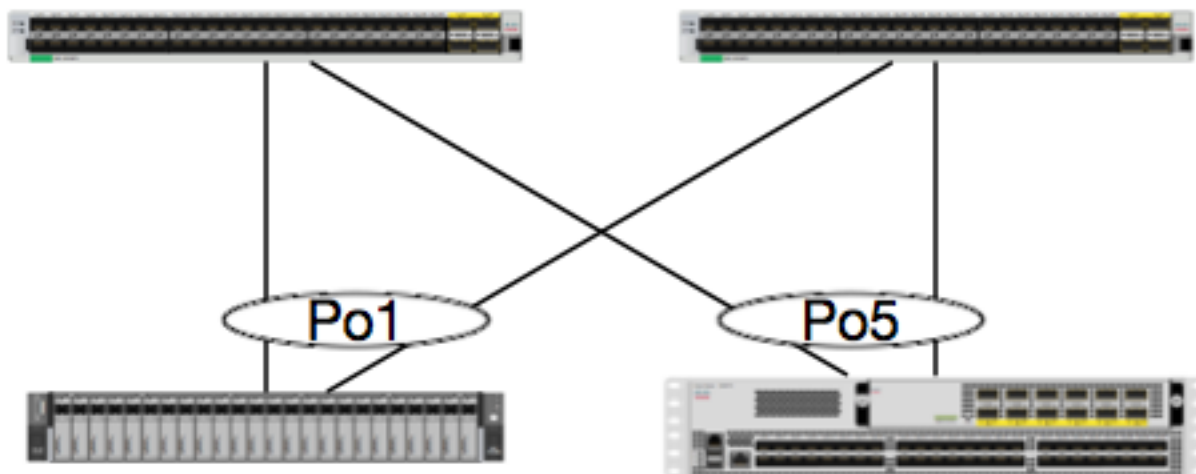
```
=====
=====
ifId      IP                HwVrfId BDXlate SrcTepIdx DstInfoIdx RwEncapIdx ECMPIdx  ECMPMbrIdx Num
L2Index   RwDmacIdx
=====
=====
18010002 192.168.120.67 2        1        3a9a     3005      6          0        0          2
1a030000 0                <---- RwEncapIdx is 6! Same as the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

1a031000 1

Deze tunnel heeft een RwEncapIdx (Re-Write Encap Index) van 6, wat in het elam weergegeven werd.

1 EP → L3 uitloop - Routed Flow

Topologie



EP1
EPG1
0050.56a5.50ab
192.168.20.10/24

N5K -OSPF
100.100.100.100/32

In dit voorbeeld volgen we de pakketstroom van een pakket van EP1 het verzenden van ICMP naar een loopback op een N5K lopende OSPF. N5K is aangesloten via een L3Out op hetzelfde paar EX-switches.

Aangezien we aan het begin van dit document de lokale programmering van het EP hebben

gecontroleerd, mogen we ervan uitgaan dat het EP correct wordt geleerd in hardware en verder gaat met de routeverificatie.

Eerst, laten we OSPF staat en de routingtabel controleren:

```
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
```

We weten dus dat de routingtabel de volgende hop toont als de 5K om 10.10.27.3. Goede start, maar hoe kunnen we controleren wat hardware heeft?

Laten we eerst de nabijheidstafel in hardware controleren om zeker te zijn dat we ARP opgelost hebben tot 10.10.27.3, en dat het geprogrammeerd is met de juiste interface:

```
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

MAC adressen die op 5K overeenkomen:

```
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
```

Op EX Platforms, is er een "hw_vrf_idx" die aan een VRF wordt toegewezen. Deze index zal worden vermeld wanneer we de hardwareprogrammering controleren. Laten we de index vinden:

```
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

```

::::

Nadat we de nabijheid hebben ontdekt, moet HAL een route programmeren. We kunnen dit controleren met de volgende opdracht:

```

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:

```

-----
|
-----

```

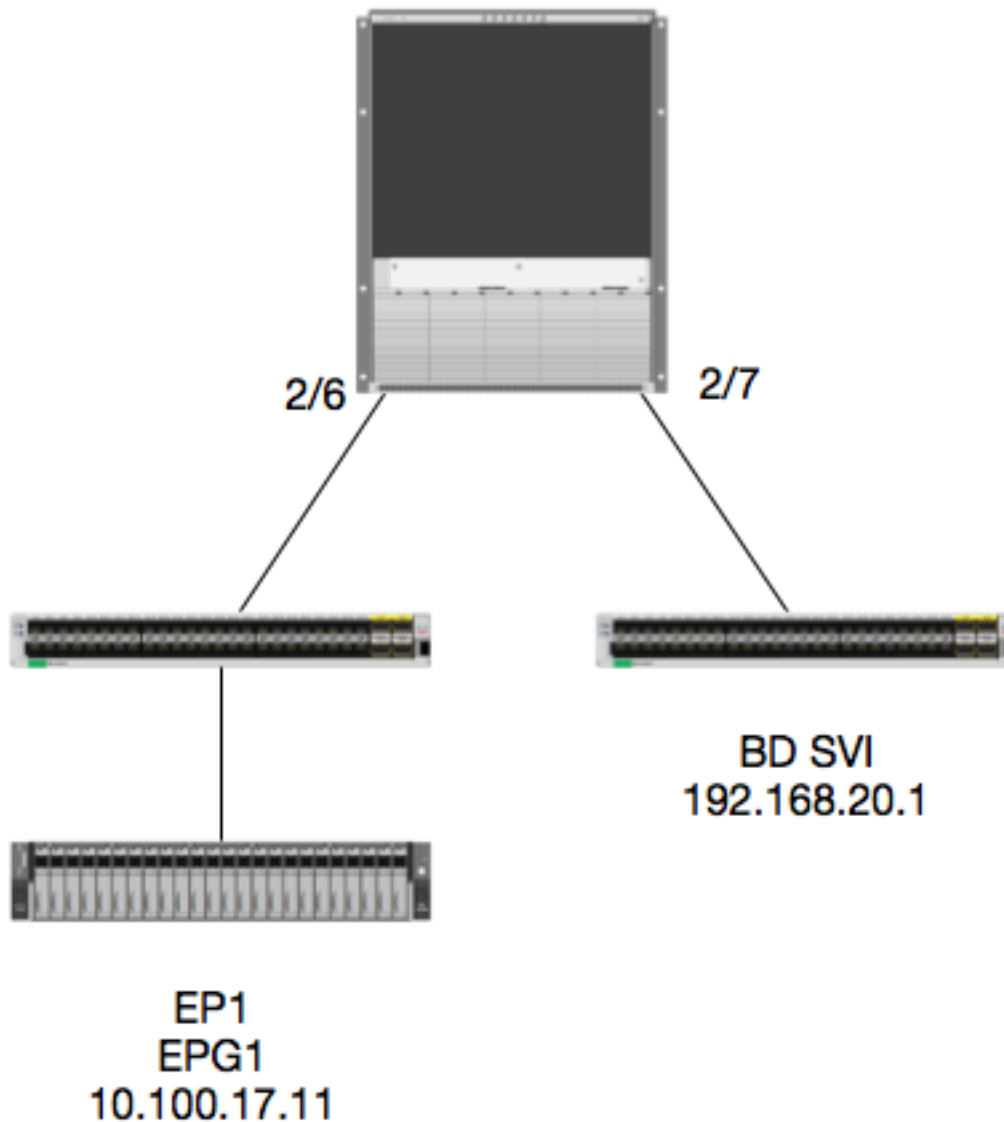


```
sug_lurw_vec.dst_addr.adj.padfield: 0x04F0288FC
sug_lurw_vec.dst_addr.adj.idx: 0x2318
sug_lurw_vec.adj_vld: 0x0
```

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

1 EP → Afstandsbediening EP of SVI - Spinetoetsing

Topologie



Logic

In dit voorbeeld volgen we de pakketstroom van een pakket van EP1 dat bestemd is voor een externe BD Switched Virtual Interface (SVI). Het doel van dit voorbeeld is om doorsturen van de wervelkolom te controleren om er zeker van te zijn dat het pakket naar de juiste Leaf wordt verzonden. Laten we aannemen dat het pakje naar de ruggengraat werd gestuurd op de innerlijke lef.

Op de ruggengraat controleren we eerst het Protocol van de Raad van Rakelen (COOP) voor de

bestemming IP omdat het pakket voor een raadpleging naar de volmacht van de Spine wordt verzonden:

```
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
```

Controleer welk blad dat TEP-adres heeft:

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

Aangezien we weten dat het pakket op module 2, poort 6 in de centrifuge komt, kunnen we aan module 2 toevoegen en de Port Layout bekijken.

```
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	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          HI:             LportProfile Hw
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	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	0	0					
D-2d4	D-3e1	0	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	1	b	b	1	1		
D-f3	D-61	100	0	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	1	c	c	1	1		
D-1ee	D-30b	100	0	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	1	1	1	1	1		
D-19a	D-2ee	100	0	0	0	0	1	0																								
1a085000	Eth2/6	0	9f	24	0	39	1	10	20	a0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	e	e	1	1		
D-87	D-184	100	0	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	1	d	d	1	1		
D-1d0	D-357	100	0	0	0	0	1	0																								
1a088000	Eth2/9	0	a2	20	1	d	0	c	18	18	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
D-3ea	D-1a9	100	0	0	0	0	1	0																								

Ethernet 2/6 is de interface die op Leaf 6 is aangesloten op ASIC 0 SLICE 1

Nu weten we welke ASIC onze ELAM aankan. 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

```

Als we naar de ELAM kijken, zien we de ovector Index:

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

Hoe brengen we 0xb8 in een haven? Aangezien we weten dat het pakket naar een Fabric Module (FM) moet worden verzonden voor een raadpleging, kunnen we naar de interne-port mapping kijken om de beste FM te vinden:

```
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

```
=====
```

Met behulp van ASIC0 / Ovec B8 krijgen we MbrID 0x7, Slice maakt niet uit.

Deze MbrID is de interface op de USD die op een FM-interface wijst. Houd in gedachten dat deze MbrID in hex is en in decimale vorm moet worden geconverteerd.

We kunnen uitzoeken welke FM door de USD-interfaces en de inspectie van haven 7 te bekijken:

```
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
```

De "sleuf" is op 0 gebaseerd en de FM-nummering is 1 gebaseerd, dus we moeten 1 toevoegen aan het nummer dat hier wordt genoemd. Dit betekent dat het pakket naar FM 23 moet worden verzonden.

Synthetisch IP

Net als in de Alpen is er een synthetisch IP, dat wordt gebruikt als het IP-adres voor het bepalen van de hash voor het COOP-lookup. Om dit te vinden, moet u deze opdracht en hulp voor de binnenste DST IP uitvoeren:

```
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
```

Dit laat zien dat 1.203.211.185 onze synthetische IP is. Op basis hiervan kunnen we ook de "Outer DST IP" op ons FM-elam instellen als dit. We moeten de FM op gang brengen:

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
```

Dump het volledige rapport natuurlijk, maar laten we kijken naar de ovector_idx voor dit pakje dat we geactiveerd hebben:

- - 0 0 0 0 0 0

Die overlijnkaarten naar LC1 (Lijnkaart in sleuf 2, omdat hij op 0 is gebaseerd), op ASIC 0 / SLICE 0. Zoals we weten van de ELAM run oorspronkelijk op de LC, werd dit segment geactiveerd:

```

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

```

De selector op deze ELAM is sug_elam_out_sideband_no_reserve_vec.ovector_idx: **0x98**, die we kennen van de "hal I2 poort gpd", kaarten naar de juiste interface op de LC:

```

=====
=====

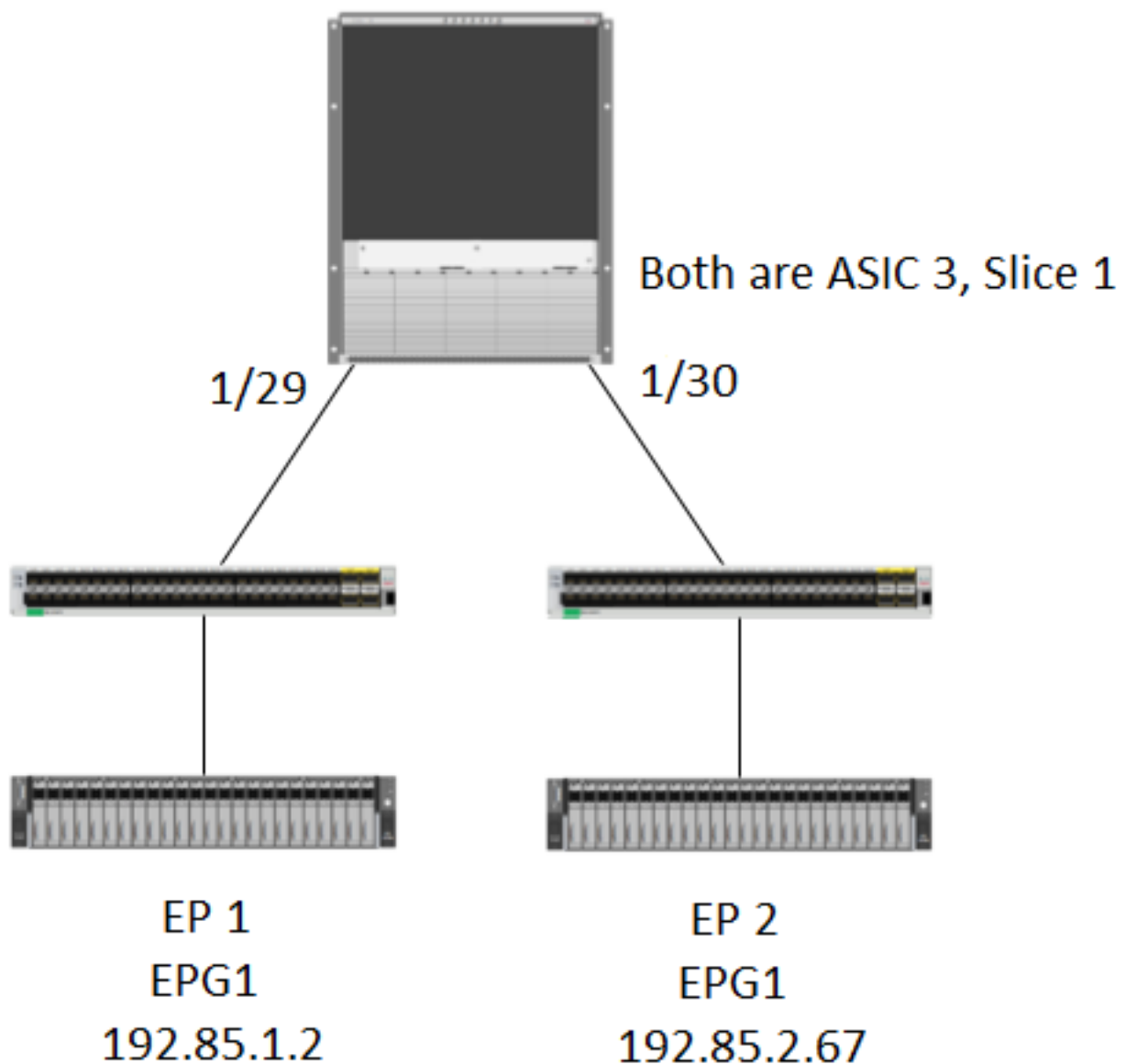
```

Rep		Uc Uc		Reprogram																							
NI Vif	RwV	Ing	Egr	I PC	Pc	L	R	I	R	D	R	U	U	X	L	X	a	O	v	x	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	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 2/7 is de interface die op Leaf 5 verbindt.

Extra scenario: Een Ovector verkrijgen die niet in de "hal interne haven pi"-uitgang zit

Topologie



Logic

Er zijn een aantal scenario's waar we een pakket vangen dat geen Ovector in de "**show platform interne hal I2 interne-poort pi**" tabel heeft. In het onderstaande scenario vangen we het pakket dat uit de FM komt, dus moeten we een andere tabel bekijken om te zien welke voorpaneel poort het pakket selecteert.

Merk op dat de topologie hierboven een volledig ander milieu is waar het transitverkeer wordt geleerd (geen volmacht routing). De module is een 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
```


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 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
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 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
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 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
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 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
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 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
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 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

1/30 is de phys-interface die zich verbindt met bladzijde 102, geverifieerd door topologie, ASIC 3, Slice 1