

EX ハードウェア : ACI パケット転送の詳細

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[シナリオ](#)

[同じ EPG/同じリーフでの 2 つの EP - スイッチド フレーム](#)

[トポロジ](#)

[ELAM](#)

[別の EPG/同じリーフでの 2 つの EP - ルーテッド パケット](#)

[トポロジ](#)

[ELAM](#)

[別の EPG/別のリーフでの 2 つの EP - ルーテッド パケット](#)

[トポロジ](#)

[ELAM](#)

[1 EP → L3アウト - ルーテッドフロー](#)

[トポロジ](#)

[ELAM](#)

[1 EP → リモート EP または SVI - スパイン 検証](#)

[トポロジ](#)

[ロジック](#)

[シンセティック IP](#)

[ファブリック モジュール ELAM](#)

[追加シナリオ : 「hal internal-port pi」出力にない Ovector の取得](#)

[トポロジ](#)

[ロジック](#)

概要

このドキュメントでは、アプリケーションセントリックインフラストラクチャ(ACI)の「EX」ベースのACIスイッチを使用する異なる転送シナリオについて説明します。ハードウェアが正しくプログラムされ、適切なエンドポイントグループ(EPG)の正しい宛先エンドポイント(EP)にパケットを転送します。

前提条件

要件

このドキュメントに特有の要件はありません。

使用するコンポーネント

このドキュメントの情報は、次のハードウェアとソフトウェアのバージョンに基づいています。

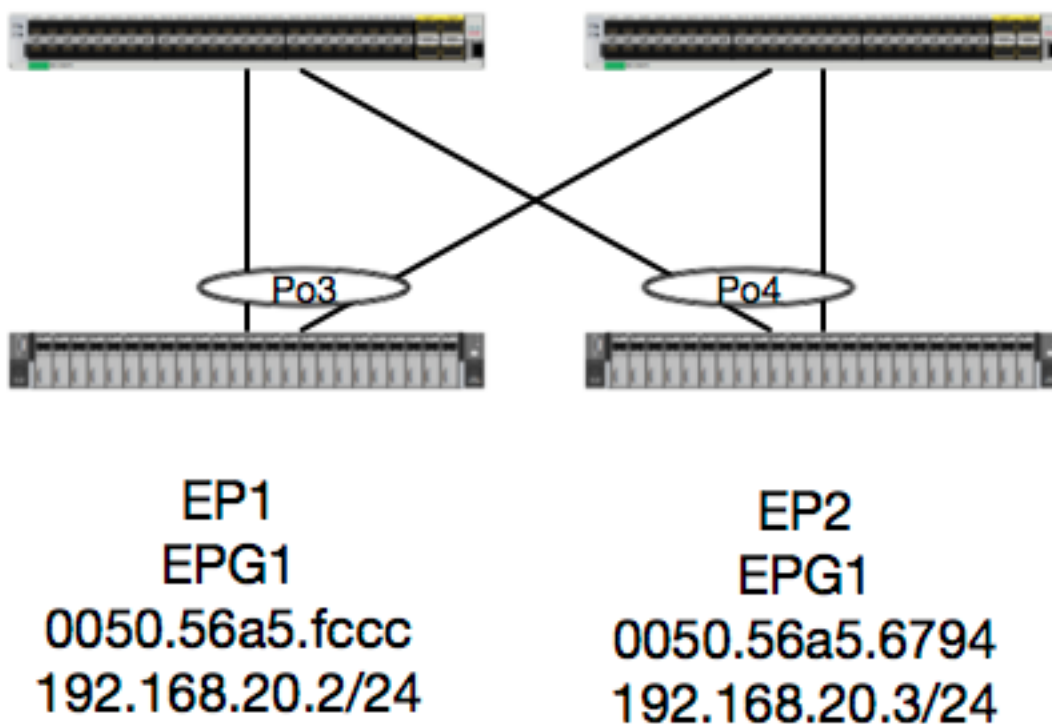
- EX ハードウェアを使用する、スパイン スイッチと 2 つのリーフ スイッチによって構成される ACI ファブリック
- 各リーフ スイッチに向かう 2 つのアップリンクを持つ ESXi ホスト
- ルータとして機能する Nexus 5000 デバイス
- 初期設定に使用されるアプリケーション ポリシー インフラストラクチャ コントローラ (APIC)

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期 (デフォルト) 設定の状態から起動しています。対象のネットワークが実稼働中である場合には、どのようなコマンドについても、その潜在的な影響について確実に理解しておく必要があります。

シナリオ

同じ EPG/同じリーフでの 2 つの EP - スイッチド フレーム

トポロジ



このトポロジの場合、EP1 から EP2 へのフローは L2 フローになり、送信元トラフィックがどのリーフに入ってくるとしても、ローカルで切り替える必要があります。レイヤ 2 (L2) のフローについて、MAC アドレス テーブルを最初に確認し、スイッチがフレームを受信しているか、またどこで受信しているかを判別する必要があります。

```
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
```

カプセル化 VLAN を確認するために、EP データベースを確かめることもできます。

```
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
```

FD_VLAN 30 が一致していることが分かりますが、いつでもソフトウェアでのマッピングを検証することができます。

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

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

もちろん、前面パネルのカプセル化として VLAN 30 が VLAN 2268 にマッピングされることを確認するためにハードウェアを確かめることができます。

```
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>

EP がソフトウェアで学習されることを考えると、ハードウェアでこれらの EP の L2 情報がプログラムされていることを確認することもできます。新しいハードウェアには、ハードウェアにおけるソフトウェアの状態を示すハードウェア アブストラクション レイヤ (HAL) があります。HAL の役割は、ソフトウェア プログラミング要求を受け取って、ハードウェアに渡すことです。

エンドポイントに関する L2 ハードウェア情報を確認するには、特定の MAC アドレスの HAL の L2 テーブルを参照できます。

```

leaf4# vsh_lc
module-1# show platform internal hal ep 12 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 1 0 0

```

```

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

```

ここまでで、ハードウェアについてまとめることができましたが、今度は ELAM について、またパケットがどこに送信される必要があるかを考えましょう。

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

```

Leaf4はAsic 0スライス1のフレームを受信しました。新しいハードウェアのELAMでは、トラブルシューティングに非常に重要な新しいフィールドがあります。ovector_idxがあります。このインデックスは、フレーム/パケットが転送される物理ポートのインデックスです。ovector_idxを設定すると、このコマンドを使用して、どのポートにマッピングされているか調べることができます。

```

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

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

```

```

RSP: RWSrcPortstate Table          Ing Lbl:      Ingress Acl Label
UC:  UCPcCfg                       Egr Lbl:      Egress Acl Label
UM:  UCPcMbr                       Reprogram:
PROF ID:      Lport Profile Id
VS:  VifStateTable                 HI:           LportProfile Hw
Install
RV:  Rw VifTable
Num. of Sandboxes: 1

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

```

```

=====
| 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 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
-      -      800 0    0 1    0 0
1a005000 Eth1/6    1 0    b     0 e 0 d 1a 1a  1  0 0 0 0 0 0 0 0 0 0 0 0
-      -      800 0    0 1    0 0
1a006000 Eth1/7    0 26   5     0 f 0 e 1c 1c  1  0 0 0 0 0 0 0 0 0 0 0 0
D-256 -      800 0    0 1    e 0
1a007000 Eth1/8    0 2e   7     0 10 0 f 1e 1e  1  0 0 0 0 0 0 0 0 0 0 0 0
D-84  -      800 0    0 1    30 0
1a01e000 Eth1/31   1 0    2d    0 37 1 e 1c 9c  1  0 0 0 0 0 0 0 0 0 0 0 0
-      -      0 0    0 1    0 0
1a01f000 Eth1/32 1 0    3d    0 38 1 f 1e 9e 1  0 0 0 0 0 0 0 0 0 0 0 0
-      -      0 0    0 1    0 0
1a030000 Eth1/49   0 2    1     0 49 1 20 38 b8  1  0 0 0 0 0 0 0 0 0 0 0 0
D-24d -      400 0    0 0    1 0
1a031000 Eth1/50   0 3    3     0 29 1 0 0 80  1  0 0 0 0 0 0 0 0 0 0 0 0
D-350 -      400 0    0 0    1 0

```

スイッチは、パケットがインターフェイスEthernet 1/32から転送されると考えています。MACアドレスを学習したPO4ですか。

```

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

```

```

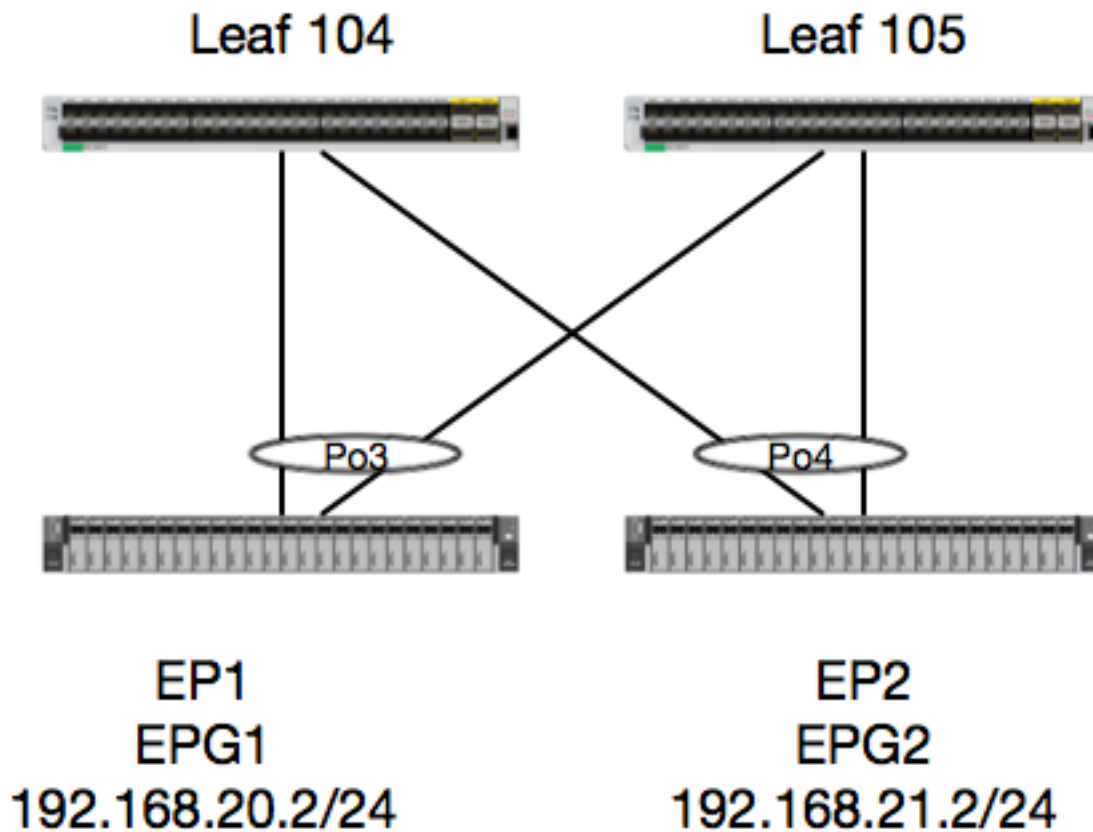
-----
Group Port-      Type      Protocol  Member Ports
  Channel
-----
1     Po1(SU)     Eth       LACP      Eth1/5(P)
2     Po2(SU)     Eth       LACP      Eth1/6(P)
3     Po3(SU)     Eth       LACP      Eth1/31(P)
4     Po4(SU)     Eth       LACP      Eth1/32(P)

```

そのとおりです。パケットはインターフェイス 1/32 から宛先ホストに転送されます。

別の EPG/同じリーフでの 2 つの EP - ルーテッド パケット

トポロジ



この例では、EP1 から EP2 へのパケットのフローを追跡します。EP1 と EP2 は同じ vPC リーフ ペアに存在しています。2 つの EP は、異なる BD を使用する、異なる EPG に存在します。

最初に EP データベースを確認し、EP を学習しているかどうかを確認めます。

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

```

EPはすでに学習されており、IP情報を把握しているため、ハードウェアのEP学習情報を確認することができます。

```

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

```

```

LEGEND:
-----
VrfName:          Vrf Name                      T:              Type
(P1: Physical, Vl: Virtual, Xr: Remote)
EP IP:           Endpoint IP
S Class:         S Class                      Age Intvl:      Age
Interval
S T:             Static Ep                      S E:
Secure EP
L D:             Learn Disable                  B N D:          Bind
Notify Disable
E N D:           Epg Notify Disable            B E:
Bounce Enable
I D L:           IVxlan Dont Learn              SPI:
Source Policy Incomplete
DPI:             Dest Policy Incomplete        SPA:
Source Policy Applied
DPA:             Dest Policy Applied            DSS:           Dest
Shared Service
IL:             Is Local                      VUB:           Vnid
Use Bd
SO:             SA Only                      EP NH L3IfName: EP
Next Hop L3 If Name
NHT:            Next Hop Type (L2: L2 Entry L3: L3 Next Hop)  BD Name:       L2 NH
BD Name
EP Mac:         EP Mac                      L3 IfName:     L3 NH
If Name
L2 IfName:     L2 If Name                      FD Name:       L2
Entry FD Name
IP:            L3 NH IP

```

L3 EP Count: 12

```

=====
=====
B E I S D S D D V EP-NH
N |
Vrf          EP          S      Age  S S L N N B D P P P P S I U S L3
H | BD          EP          L3      L2      FD
Name          T IP          Class Intvl T E D D D E L I I A A S L B O
IfName        T | Name      Mac      IfName      Ifname      Name      IP
=====
=====
common*rewall P1 10.6.112.1          1      0      1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -

```



```

L3 - 00:00:00:00:00:00 - - - 0.0.0.0
common*rewall Pl 10.6.114.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
common*rewall Pl 10.6.114.129 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
common*efault Pl 100.100.101.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
Joey-T*ternal Pl 192.168.1.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
Joey-T*ternal Xr 192.168.1.100 8013 128 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 -
L3 - 00:0c:0c:0c:0c:0c Tunnel12 Tunnel12 - 0.0.0.0
Joey-T*ternal2 Pl 192.168.3.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
Joey-T*ternal Pl 192.168.20.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
Joey-T*ternal Pl 192.168.20.2 800a 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 -
L2 BD-28 00:50:56:a5:fc:cc - Po3 FD-30 -
Joey-T*ternal Pl 192.168.21.1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0
Joey-T*ternal Pl 192.168.21.2 800c 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 -
L2 BD-7 00:50:56:a5:0c:11 - Po4 FD-8 -
Joey-T*ternal Pl 2001:0:0:100::1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 -
L3 - 00:00:00:00:00:00 - - - 0.0.0.0

```

HAL Layer3 (I3) のテーブルでは、I3 学習済み EP に関する VLAN/ポート情報が提供されるため非常に役立ちます。Po4 の宛先が存在するため、パケットは Po4 のいずれかのポートから転送される必要があります。

ELAM を実行してみて、どのような結果が得られるか確認してみましょう。

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

```

パケットをトリガーし、「ovector_idx」が0x9Eであることがわかりました。ovectorインデックスは、パケットを転送する必要がある発信の物理インターフェイスインデックスです。どのポートがそのインデックスに設定されているか見てみましょう。

```

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

```

```

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

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

```

```

=====
| Rep |          Uc   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 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

```

ポート 1/32 から送信する必要があるので、これは正しいでしょうか。

```

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

```

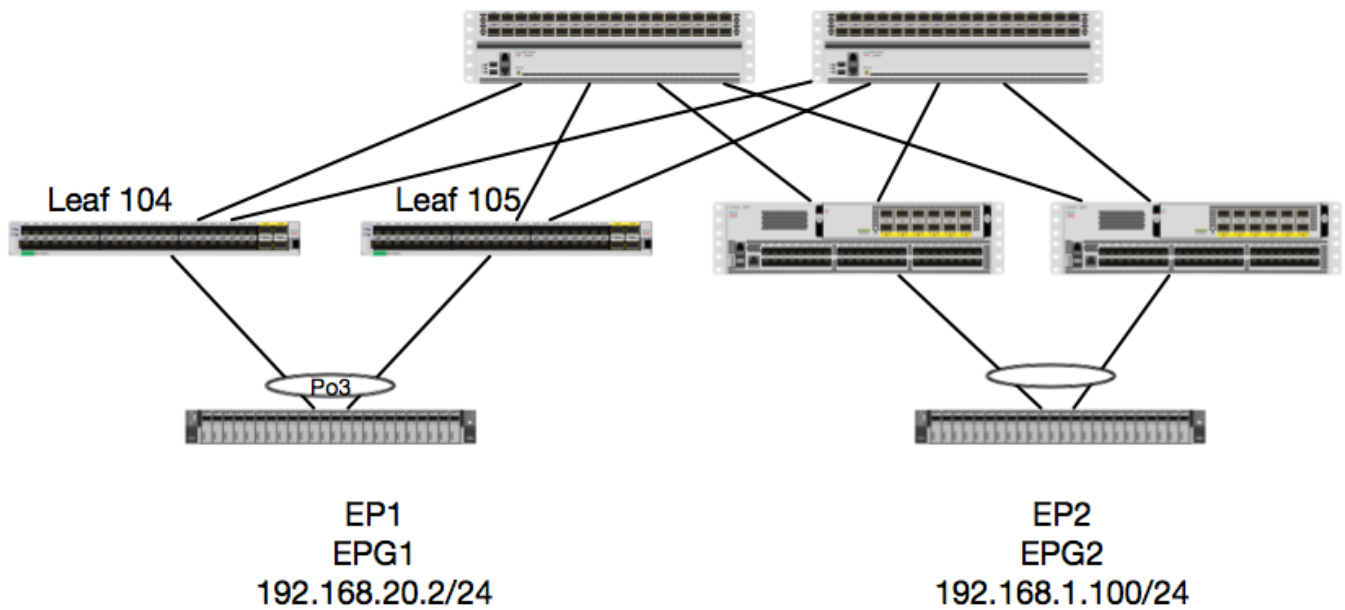
F - Configuration failed

```
-----  
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)
```

はい、これは適切です。

別の EPG/別のリーフでの 2 つの EP - ルーテッド パケット

トポロジ



この例では、EP1 から EP2 へのパケットのフローを追跡します。EP1 は EX vPC ペアに存在し、EP2 はリモートの Generation 1 vPC リーフのペアに存在しています。2 つの EP は、異なる BD を使用する、異なる EPG に存在します。

もう一度、EP が学習されている場所を確認してみましょう。

```
leaf4# show endpoint ip 192.168.20.2
```

```
Legend:
```

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

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

calo2-leaf4# show endpoint ip 192.168.1.100

Legend:

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

```

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

```

ここで、ハードウェアがどのようにプログラムされているか確かめてみましょう。

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 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

```

ハードウェアはEPがTunnel 2上にあると認識しています。Tunnel 2の宛先は何ですか。

```

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
```

宛先は vPC の外部にあるため、その宛先 IP はリモート リーフの vPC の仮想 IP である必要があります。リモート リーフを確認してみます。次をご覧ください。

```
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
```

リモート vPC ペアから宛先 EP を学習しています。ELAM が何を認識するかを確認し、パケットが正しく転送されていることを確かめてみましょう。

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
```

ここで、EX ハードウェアのリモートの宛先には、パケット フローをトラブルシューティングする際に非常に重要になる 2 つの ELAM の値があります。ovector_idx は前の例と同様で、encap_idx は次のようになります。

```
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
```

EX ハードウェアでは、パケットが転送される宛先ポートを有効にすることができます。これまでは、encap インデックスを確認し、宛先のインデックスが正しいトンネルであることを確認するのみでした。ここでは、どのポートが 8B にマップされているかを確認できます。

```
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
As:            Asic
S1:           Slice
Ss:           Slice SrcId
srcid)
L S:           Local Slot
L3:           Is L3
  P:          PifTable
  RP:         Rw PifTable
  IP:         If Profile Table
  RS:         Rw SrcId Table
  DP:         DPort Table
  SP:         SrcPortState Table
  RSP:        RwSrcPortstate Table
  UC:         UCPcCfg
  UM:         UCPcMbr
PROF ID:       Lport Profile Id
  VS:        VifStateTable
Install
  RV:        Rw VifTable
Num. of Sandboxes: 1

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

```

```

=====
=====
| Rep |
      Uc  Uc
      I PC  Pc
NI Vif  RwV  Ing  Egr  | V R | PROF H
IfId    Ifname  P Cfg  MbrID As AP S1 Sp Ss Ovec S | P P P S P Sp Sp C M L | 3 Idx Idx L3
L3 Tid  Tid   Lbl  Lbl  | S V | ID  I
=====
=====
1a004000 Eth1/5      1 0    1d    0 d 0 c 18 18  1  0 0 0 0 0 0 0 0 0 0 0 0
-        -          800 0    0 1    0    0
1a005000 Eth1/6      1 0    b     0 e 0 d 1a 1a  1  0 0 0 0 0 0 0 0 0 0 0 0
-        -          800 0    0 1    0    0
1a006000 Eth1/7      0 26   5     0 f 0 e 1c 1c  1  0 0 0 0 0 0 0 0 0 0 0 0
D-256   -          800 0    0 1    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

```

スイッチはインターフェイスEth1/49のスパンに転送すべきと考えています。しかし、encapが正しいことを確認するにはどうすればよいのですか。

まず、トンネルに関するハードウェア情報を確認する必要があります。次の HAL コマンドを実行して、これを行えます。

```

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

```


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

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

Num. of Sandboxes: 1

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

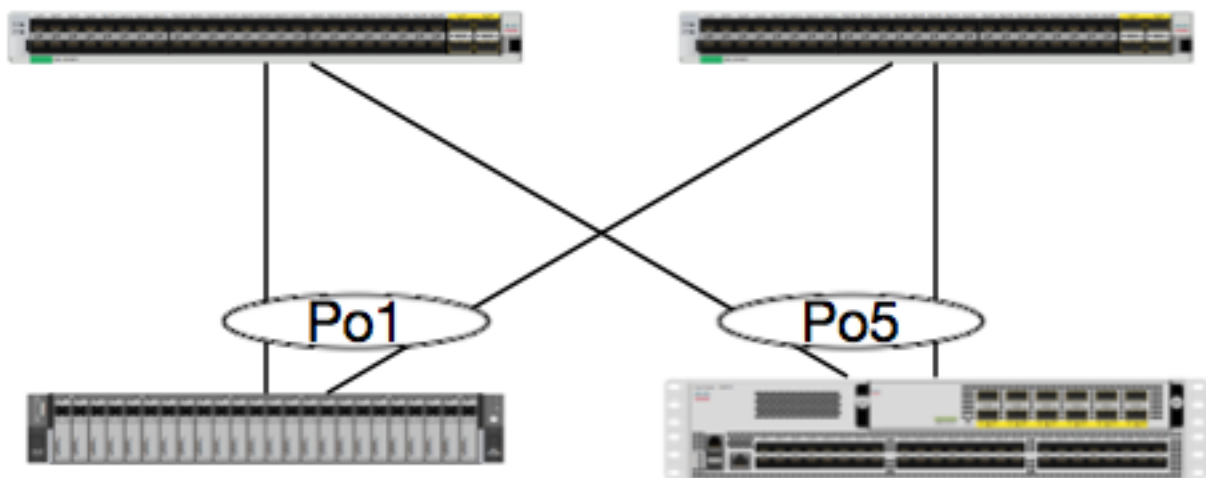
```
=====
=====
ifId      IP          HwVrfId BDXlate SrcTepIdx DstInfoIdx RwEncapIdx ECMPIdx  ECMPMbrIdx Num
L2Index  RwDmacIdx
=====
=====
18010002 192.168.120.67 2        1        3a9a     3005     6         0         0         2
1a030000 0                <---- RwEncapIdx is 6! Same as the "encap_idx"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

1a031000 1

このトンネルには、値 6 の RwEncapIdx (Encap インデックスの書き換え) があります。これは、ELAM に表示されている内容です。

1 EP → L3アウト - ルーテッドフロー

トポロジ



EP1
EPG1
0050.56a5.50ab
192.168.20.10/24

N5K -OSPF
100.100.100.100/32

この例では、OSPFを実行しているN5KのループバックにICMPを送信するEP1からのパケットのパケットフローを追跡します。N5Kは、同じEXスイッチペア上のL3Out経由で接続されます。

このドキュメントの最初で、ローカル EP プログラミングについて確認しているため、ここでは

ハードウェアで EP が正しく学習されていることを前提とし、ルートの検証に話を進めましよう

。

最初に、OSPF の状態とルーティング テーブルを確認してみましょう。

```
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
```

ルーティングテーブルに10.10.27.3の5Kのネクストホップが示されていることがわかっています。良いスタートですが、どのハードウェアが搭載されているかを確認するにはどうしたらいいですか。

まず、ハードウェアの隣接関係テーブルをチェックし、ARP が 10.10.27.3 に解決されていること、また正しいインターフェイスにプログラムされていることを確認します。

```
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 アドレスは、次のように 5K 上のアドレスと一致しています。

```
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
```

EXプラットフォームには、VRFに割り当てられた「hw_vrf_idx」があります。このインデックスは、ハードウェアプログラミングを確認するときに参照されます。インデックスを検索してみましょう。

```
module-1# show system internal eltmc info vrf jr:sb
```

```
VRF-TABLE: jr:sb
vrf_type:          tenant    :::      context_id:        6
overlay_index:     0        :::      vnid:              2129921
scope:             5        :::      sclass:            16386
v4_table_id:       0x5      :::      v6_table_id:       0x80000005
intf_count:        5        :::      intrn_vlan_id:     0
```

```

VRF Intf:      Vlan11   ::: src_plcy_incomp:      0
vniid_hex:    0x208001 ::: ingress_policy:      0x1
vrf_intf_list: Vlan28,Vlan16,Vlan9,Vlan11,loopback2,
hw_vrf_idx:      4612  ::: nb_egr_outer_bd:      0
sb_egr_outer_bd:      0
vrf_bd_list:  28,16,11,9,
sb_egr_outer_bd:      0 :::      sdk_vrf_id:      5

```

[SDK Info]:

```

vrf_name:      jr:sb
vrf_id:        5      :::      hw_vrf_idx:      4612
vrf_vniid:    2129921 :::      is_infra:      0
tornbinfracwbd: 0      ::: torsbinfracwbd: 0
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
vniid:        2129921
internal infra vlan: 0
external router mac:00:22:bd:f8:19:ff

```

v6 table

```

vrf type:      1
vrf id:        5
vniid:        2129921
internal infra vlan: 0
external router mac:00:22:bd:f8:19:ff

```

:::

隣接関係を検出すると、HAL はルートをプログラムします。次のコマンドを使用してこれを確認できます。

```
module-1# show platform internal hal 13 routes | head
```

LEGEND:

```

LID: Logical ID          RID: Route ID          PID: Physical ID      NB-ID:Next-Base ID
HIT IDX: Next-Hop HitIndex  CLP : Class Priority  TBI: Trie Base Index |
SC : Sup-Copy            SSR: Src Sup-Redirect  DSR: Dst Sup-Redirect TDD :TTL Disable
NB: NextBaseType        SDC : Src Direct Connect  TRO: Trie Offset      |
SPI: Src Policy Inc     DPI: Dst Policy Inc     DR : Default Route    LE  :Learn Enable
[E:Ecmp/A:Adj]         ILL : Is Link Local    ISS: Is Shared Services |
RT : Route Type         FWD: Forwarding        HR : Host Routes      EP  :Ext Prefixes
DLR: Default Lpm Route  CLSS: Class Id         RDEL: Route in Deletion |
BNE: Bind Notify Enable SNE: Sclass Notify Enable BE : Bounce Enable    IDL :Ivxlan
DoNotLearn DL : Dest Local          SA : Src Only         AI : Age Interval
|
SF : Static Flag        SH : Src Hit           DH: Dest Hit

```

```
module-1# show platform internal hal 13 routes
```

LEGEND:

|

```

-----
LID: Logical ID          RID: Route ID          PID: Physical ID      NB-ID:Next-Base ID
HIT IDX: Next-Hop HitIndex  CLP : Class Priority  TBI: Trie Base Index |
SC : Sup-Copy           SSR: Src Sup-Redirect  DSR: Dst Sup-Redirect  TDD :TTL Disable
NB: NextBaseType       SDC : Src Direct Connect  TRO: Trie Offset      |
SPI: Src Policy Inc     DPI: Dst Policy Inc     DR : Default Route     LE  :Learn Enable
[E:Ecmp/A:Adj]         ILL : Is Link Local     ISS: Is Shared Services |
RT : Route Type         FWD: Forwarding        HR : Host Routes       EP  :Ext Prefixes
DLR: Default Lpm Route   CLSS: Class Id         RDEL: Route in Deletion |
BNE: Bind Notify Enable  SNE: Sclass Notify Enable  BE : Bounce Enable     IDL :Ivxlan
DoNotLearn DL : Dest Local          SA : Src Only          AI : Age Interval
|
SF : Static Flag        SH : Src Hit           DH: Dest Hit
|

```

```

-----
|          |          |          |          | LID |<-----
- Trie ----->|<Dleft Trie>|          |          |          |          |          |          |          |          |
| VRF |          |          |          |          | RT| RID |          | LID | Type| PID | FPID/| HIT
|N| NB-ID | NB Hw | PID | FPID/| TBI |TRO|Ifindex|CLSS|CLP| AI |SH|DH| Flags
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|B|          | Idx |          | TID |-----|---|-----|-----|---|---|---|---|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|<-----|
- DLEFT ----->|-----|-----|-----|---|-----|-----|---|---|-----|
|          |          |          |          |          |          |          |          |          |          |          | |
|N| NB-ID | NB Hw |          |          |          |          |          |          |          |          |          |
|          |          |          |          |          |          |          |          |          |          |          |
|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|          | 0| f| 3| 0| 0| 0|spi,dpi
この出力は、ネクストホップのルートに関する情報を示します。4612は、jr:sb VRFの
hw_vrf_idxです。ネクストホップを確認するために、TCAMの「NB Hw Idx」が次の表に対して使
用されます。

```

```

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 RwVRF : 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)

```

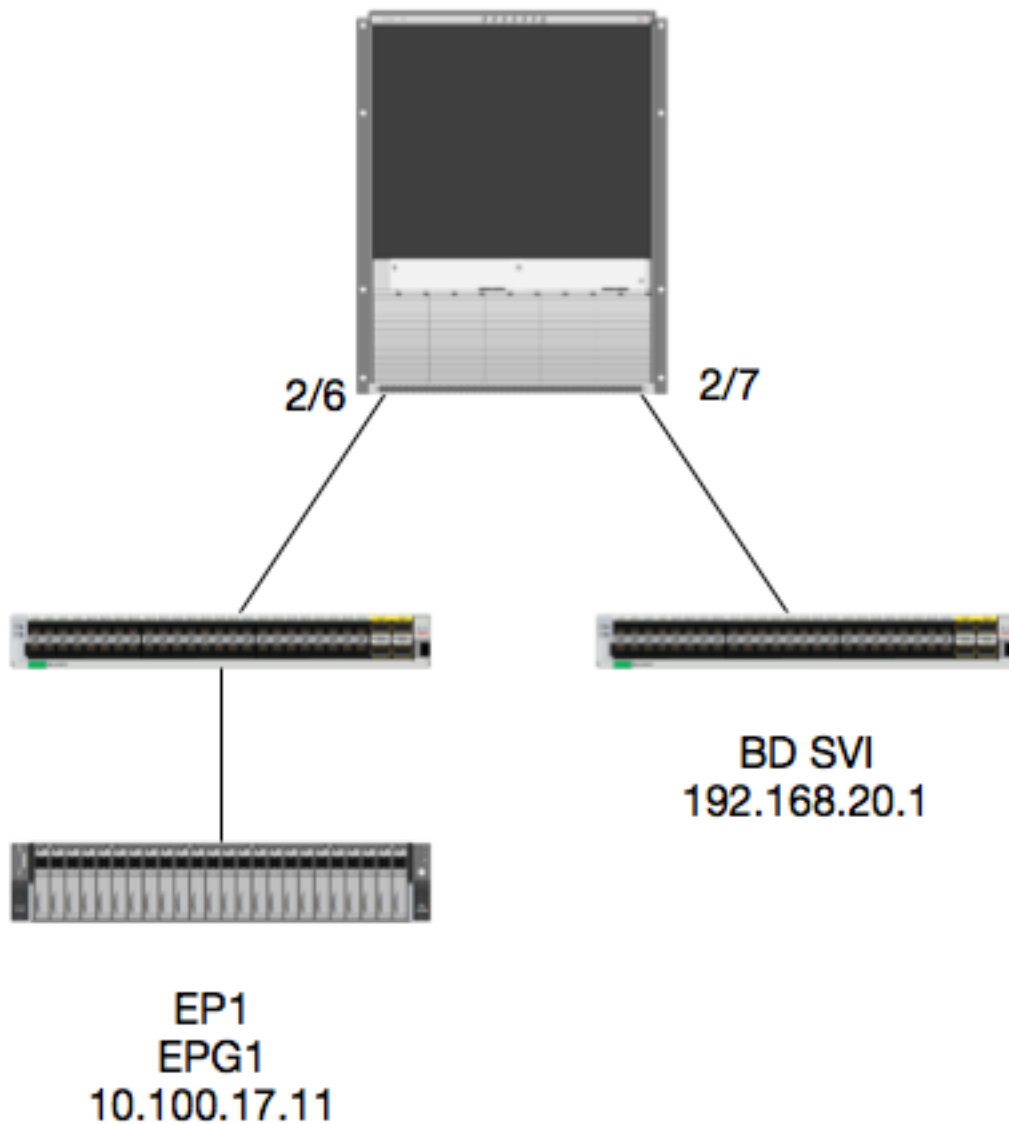


```
leaf6# cat elam_report.txt | grep adj
    sug_lurw_vec.dst_addr.adj: 0x8C604F0288FC
    sug_lurw_vec.dst_addr.adj.padfield: 0x04F0288FC
    sug_lurw_vec.dst_addr.adj.idx: 0x2318
    sug_lurw_vec.adj_vld: 0x0

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

1 EP → リモートEPまたはSVI – スパイン検証

トポロジ



ロジック

この例では、EP1からRemote BD Switched Virtual Interface(SVI)宛てのパケットのパケットフローを追跡します。この例の目的は、スパイン転送を確認して、パケットが正しいリーフに送信されることを確認することです。パケットが入カリーフのスパイン プロキシに送信されたことを前提とします。

パケットはルックアップのためにスパイン プロキシに送信されるため、スパインでまず宛先 IP の Council of Oracles Protocol (COOP) を確認します。

```
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
```

どのリーフがその TEP アドレスを持つか確かめてみましょう。

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

パケットは Module 2、Port 6 のスパインで受信されることが分かっているため、Module 2 に接続して、ポート レイアウトを確認します。

```
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.
calol-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
Sl:	Slice	Sp:	Slice Port
Ss:	Slice SrcId	Ovec:	Ovector (slice
srcid)			
L S:	Local Slot	Reprogram:	
L3:	Is L3		
P:	PifTable	Xla Idx:	Xlate Idx
RP:	Rw PifTable	Ovx Idx:	OXlate Idx
IP:	If Profile Table	N L3:	Num. of L3 Ifs

```

RS:   Rw SrcId Table           NI L3:           Num. of Infra L3 Ifs
DP:   DPort Table             Vif Tid:        Vif Tid
SP:   SrcPortState Table      RwV Tid:        RwVif Tid
RSP:  RwSrcPortstate Table    Ing Lbl:        Ingress Acl Label
UC:   UCPcCfg                 Egr Lbl:        Egress Acl Label
UM:   UCPcMbr                 Reprogram:
PROF ID:           Lport Profile Id
VS:   VifStateTable           HI:             LportProfile Hw
Install
RV:   Rw VifTable
Num. of Sandboxes: 1

```

```

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

```

```

=====
=====

```

													Reprogram														
Rep	Uc				Uc				L					R	U	U	X	L	Xla	Ovx	N						
NI Vif	RwV	Ing	Egr	I	PC	Pc	V	R	PROF	H	S	P	P	P	S	P	Sp	Sp	C	M	L	3	Idx	Idx	L3		
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	
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	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	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	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	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	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	0
D-3ea	D-1a9	100	0	0	0	1	0																				

Ethernet 2/6は、Leaf 6に接続するインターフェイスはASIC 0 SLICE 1上にあります

これで、どの ASIC で ELAM を実行するかが分かりました。ASIC 0 です。

```

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

```

ELAMを確認すると、ovector インデックスが分かります。

Front Panel ELAM drove `sug_elam_out_sidebnd_no_spare_vec.ovector_idx: 0xb8`

さて、どのように 0xb8 をポートにマッピングしたらよいでしょうか。パケットはルックアップのためにファブリック モジュール (FM) に送信されるため、内部ポート マッピングを調べて宛先 FM を見つけます。

```
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

```
=====
```

ASIC0 / Ovec B8を使用して、MbrId 0x7を取得します。

この MbrId は、FM 上のインターフェイスにマッピングされる USD のインターフェイスです。この MbrId は 16進数であり、10進数に変換する必要があることに注意してください。

USD インターフェイスを確認し、Port 7 を調べることによって、どの FM かが分かります。

```
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
```

「スロット」は 0 から始まりますが、FM の番号付けは 1 から始まります。そのため、ここでリストされている番号に 1 を追加する必要があります。つまり、パケットは FM 23 に送信されま

シンセティック IP

Alpine と同様、COOP ルックアップ用のハッシュを決定するための、外部用 IP アドレスとして使用されるシンセティック IP があります。これを検出するには、次のコマンドを実行して、内部用 DST IP を grep する必要があります。

```
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
```

これは、1.203.211.185 が合成 IP であることを示しています。これをもとに、FM elam の「Outer DST IP」を設定することもできます。次のように、FM 上でトリガーする必要があります。

ファブリック モジュール 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
```

このように、レポートがすべてダンプされますが、トリガーしたこのパケットの ovector_idx を確認します。


```
b4      fc0-lc1:3-0 1 0    33    0 51 3 8 10 d0  1  0 0 0 0 0 0 0 0 0 0 0 0 0 0
-      -      0  0    0 0  0  0
```

このovectorは、ASIC 0/SLICE 0上のLC1 (スロット2のラインカードは0ベースであるため) にマッピングされます。ELAMがLC上で最初に行っていることがわかるように、次のスライスでトリガーされます。

```
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-insell3)# set inner ipv4 src_ip 10.100.17.11 dst_ip 192.168.20.1
module-2(DBG-TAH-elam-insell3)# start
stat
module-2(DBG-TAH-elam-insell3)# stat
ELAM STATUS
=====
Asic 0 Slice 0 Status Armed
Asic 0 Slice 1 Status Armed
```

```
module-2(DBG-TAH-elam-insell3)# stat
ELAM STATUS
=====
Asic 0 Slice 0 Status Triggered <---- Packet triggered from FM
Asic 0 Slice 1 Status Triggered <---- Packet triggered from Front Panel
```

このELAMのovectorは sug_elam_out_sidebnd_no_spare_vec.ovector_idx:0x98 です。これは、「hal l2 port gpd」から分かりますが、LC上の適切なインターフェイスにマッピングされます。

```
=====
```

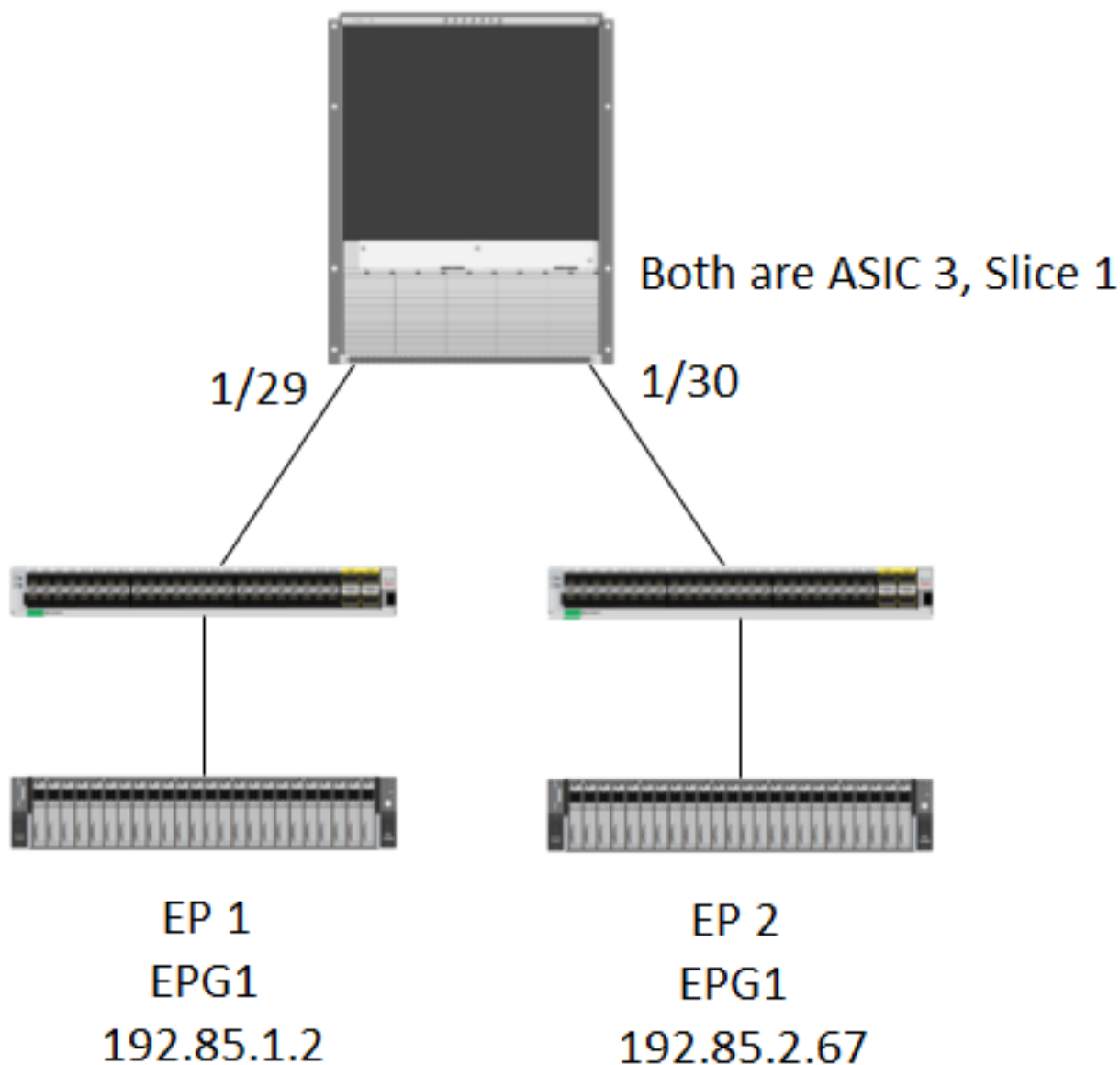
										Reprogram									
Uc					Uc														
Rep					I PC	Pc					L	R I R D	R U U X	L Xla OvX N					
NI Vif	RwV	Ing	Egr	V R	PROF H						P P P S P	Sp	Sp	C M L	3 Idx Idx L3				
IfId	Ifname	P Cfg	MbrID	As AP	S1 Sp	Ss	Ovec	S	P P P S P	Sp	Sp	C M L	3 Idx Idx L3						
L3 Tid	Tid	Lbl	Lbl	S V	ID	I													
1f5	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											
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 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 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 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 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 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 0 0 0	1												
D-3ea	D-1a9	100 0	0 0	1 0															

```
=====
```

Ethernet 2/7は、リーフ5に接続するインターフェイスです。

追加シナリオ：「hal internal-port pi」出力にない Ovector の取得

トポロジ



ロジック

「show platform internal hal l2 internal-port pi」テーブルに Ovector がないパケットを捕捉するいくつかのシナリオがあります。以下のシナリオでは、実際には FM から戻ってくるパケットを捕捉するため、別のテーブルを確認して、パケットがどの前面パネルポートを選択しているかを確認する必要があります。

上記のトポロジは、中継トラフィックを学習する（プロキシルーティングなし）、まったく異なる環境のものです。モジュールは N9K-X9732C-EX です。

```
@module-1# debug platform internal tah elam asic 3
@module-1(DBG-elam)# trigger reset
@module-1(DBG-elam)# trigg init in-select 13 out-select 0
@module-1(DBG-elam-insel13)# set inner ipv4 src_ip 192.85.1.2 dst_ip 192.85.2.67
@module-1(DBG-elam-insel13)# star
@module-1(DBG-elam-insel13)# stat
ELAM STATUS
=====
Asic 3 Slice 0 Status Armed
```

Asic 3 Slice 1 Status Triggered

@module-1(DBG-elam-insel13)# report | grep ovector
sug_elam_out_sidebnd_no_spare_vec.ovector_idx: 0xA0 <<<<<<<<<<<<<<<<<<<<< now we look for this in
the "hal internal-port pi" command

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

No sandboxes exist
Num. of Sandboxes: 1
Legend:

IfId: Interface Id IfName: Interface Name
As: Asic AP: Asic Port
Sl: Slice SP: Slice Port
Ss: Slice SrcId Ovec: Ovector
UcPcCfgId: Uc Pc CfgId Lb Mbrid: LB MbrId
Sandbox_ID: 0, BMP: 0x0
Internal Port Count: 24

```
=====
```

IfId	IfName	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
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

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	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
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
1	D-3f2	D-7a	100	0	0	0	2	0							
1a01d000	Eth1/30	0	38	20	3	39	1	10	20	a0	1	0	0	0	0
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
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
1	D-154	D-5d	100	0	0	0	2	0							

1/30は、リーフ102に接続する物理インターフェイスで、トポロジによって確認されます。ASIC 3、スライス1