

# ACI ファブリックを通過するパケット フローの確認

## 内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[ACI ファブリック パケット フローの判別](#)

[2つのエンドポイントが同じリーフ上にある単一 BD / 単一 EPG](#)

[2つのエンドポイントが異なるリーフ上にある単一 BD / 単一 EPG](#)

[同じリーフ上の各 EPG に 1つのエンドポイントがある単一 BD / 2つの EPG](#)

[同じリーフ上の各 EPG に 1つのエンドポイントがある 2つの BD / 2つの EPG \( ルートパケット \)](#)

## 概要

このドキュメントでは、さまざまな状況でアプリケーション セントリック インフラストラクチャ ( ACI ) ファブリックを通過するパケット フローを判別する方法について説明します。

注：このドキュメントで説明されているすべての状況には動作中の ACI ファブリックが含まれるため、ハードウェアのパケット フローをトレースすることができます。

## 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。

- 2つのスパイン スイッチと 2つのリーフ スイッチから構成される 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
- 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。

## ACI ファブリック パケット フローの判別

このセクションでは、ACI ファブリックが使用されるさまざまな状況とパケット フローの判別方法について説明します。

### 2つのエンドポイントが同じリーフ上にある単一 BD / 単一 EPG

このセクションでは、同一のリーフ スイッチ上にある同一のエンドポイント グループ ( EPG ) / ブリッジ ドメイン ( BD ) 内の 2 つのエンドポイントのためのハードウェアのプログラミングとパケット フローを確認する方法を説明します。仮想マシン ( VM ) が同一ホスト上で動作している場合、それらは同じ EPG にあるため、トラフィックはホストの仮想スイッチ ( VS ) に分離され、トラフィックがホストを離れる必要はありません。VM が別のホスト上で動作している場合、次の情報が適用されます。

まず確認すべきことは、リーフ スイッチ上の送信元と宛先の IP アドレスに関して、メディア アクセス制御 ( MAC ) アドレス情報を学習できるかどうかです。次に、この例で使用される MAC アドレスおよび IP アドレス情報を示します。

- 送信元MACアドレス : 0050.5695.17b7
- 送信元 IP アドレス : 192.168.3.2
- 宛先MACアドレス : 0050.5695.248f
- 宛先 IP アドレス : 192.168.3.3

この情報を確認するには、`show mac address-table` コマンドを入力します。

```
leaf2# show mac address-table
Legend:
* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen,+ - primary entry using vPC Peer-Link,
(T) - True, (F) - False
VLAN MAC Address Type age Secure NTFY Ports/SWID.SSID.LID
-----+-----+-----+-----+-----+-----+-----
16 0050.5695.248f dynamic - F F tunnel4
* 19      0050.5695.17b7      dynamic -      F   F   eth1/31
* 19      0050.5695.248f      dynamic -      F   F   eth1/31
```

ここで示すように、システムは同じ VLAN 上にある両方のエンドポイントの MAC アドレスを学習します。この VLAN はプラットフォームに依存しない ( PI ) VLAN であり、各スイッチにローカルで有効です。これが正しい PI VLAN であることを確認するには、`vsh_lc` に接続して CLI に次のコマンドを入力します。

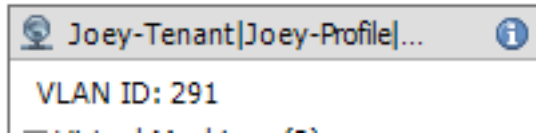
```
module-1# show system internal eltc info vlan brief
VLAN-Info
VlanId HW_VlanId Type Access_enc Access_enc Fabric_enc Fabric_enc BDVlan
Type Type
=====
9 11 BD_VLAN Unknown 0 VXLAN 16613250 9
```

```

10 12 BD_VLAN Unknown 0 VXLAN 15990734 10
13 13 FD_VLAN 802.1q 299 VXLAN 8507 10
16 14 BD_VLAN Unknown 0 VXLAN 16449431 16
17 15 FD_VLAN 802.1q 285 VXLAN 8493 16
18 16 BD_VLAN Unknown 0 VXLAN 15761386 18
19      17      FD_VLAN      802.1q      291      VXLAN      8499      18

```

HW\_VlanId は Broadcom が使用する VLAN です。VlanId は PI VLAN であり、VLAN プールから取得した Access\_enc VLAN 291 にマッピングされ、分散仮想スイッチ ( DVS ) ポートグループに伝搬される VLAN です。



このトラフィック フローは同じ BD および同じ VLAN 内にあるため、トラフィックは Broadcom ASIC 上でローカルに切り替える必要があります。Broadcom にハードウェアの正しいエントリがあることを確認するには、Broadcom シェルに接続し、レイヤ 2 ( L2 ) テーブルを参照します。

```

leaf2# bcm-shell-hw
unit is 0
Available Unit Numbers: 0
bcm-shell.0> l2 show
mac=00:22:bd:f8:19:ff vlan=19 GPORT=0x7f modid=2 port=127 Static
mac=00:50:56:95:68:c4 vlan=25 GPORT=0x5f modid=0 port=95/xe94 Hit
mac=00:22:bd:f8:19:ff vlan=16 GPORT=0x7f modid=2 port=127 Static
mac=00:22:bd:f8:19:ff vlan=29 GPORT=0x7f modid=2 port=127 Static
mac=00:22:bd:f8:19:ff vlan=32 GPORT=0x7f modid=2 port=127 Static
mac=00:22:bd:f8:19:ff vlan=26 GPORT=0x7f modid=2 port=127 Static
mac=00:50:56:95:24:8f vlan=17 GPORT=0x1f modid=0 port=31/xe30 Hit
mac=00:22:bd:f8:19:ff vlan=18 GPORT=0x7f modid=2 port=127 Static
mac=00:22:bd:f8:19:ff vlan=21 GPORT=0x7f modid=2 port=127 Static
mac=00:22:bd:f8:19:ff vlan=34 GPORT=0x7f modid=2 port=127 Static
mac=00:50:56:95:26:5e vlan=25 GPORT=0x5f modid=0 port=95/xe94 Hit
mac=00:50:56:95:c3:6f vlan=24 GPORT=0x5f modid=0 port=95/xe94 Hit
mac=00:50:56:95:5c:4d vlan=28 GPORT=0x1e modid=0 port=30/xe29 Hit
mac=00:22:bd:f8:19:ff vlan=12 GPORT=0x7f modid=2 port=127 Static Hit
mac=00:22:bd:f8:19:ff vlan=11 GPORT=0x7f modid=2 port=127 Static
mac=00:50:56:95:17:b7 vlan=17 GPORT=0x1f modid=0 port=31/xe30 Hit
mac=00:50:56:95:4e:d3 vlan=30 GPORT=0x1e modid=0 port=30/xe29 Hit
mac=00:22:bd:f8:19:ff vlan=14 GPORT=0x7f modid=2 port=127 Static

```

この出力から、Broadcom ASIC のプログラミングが正しいこと、およびトラフィックは VLAN 17 でローカルに切り替える必要があることがわかります。

## 2 つのエンドポイントが異なるリーフ上にある単一 BD / 単一 EPG

このセクションでは、同一のエンドポイント グループ ( EPG ) / ブリッジ ドメイン ( BD ) 内にあるが、異なるリーフ スイッチ上にある 2 つのエンドポイントのためのハードウェアのプログラミングとパケット フローを確認する方法を説明します。

まず確認すべきことは、リーフ スイッチ上の送信元と宛先の IP アドレスに関して、MAC アドレス情報を学習できるかどうかです。次に、この例で使用される MAC アドレスおよび IP アドレス情報を示します。

- 送信元 MAC アドレス : 0050.5695.17b7
- 送信元 IP アドレス : 192.168.3.2

- 宛先MACアドレス : 0050.5695.bd89
- 宛先IPアドレス : 192.168.3.11

この情報を確認するには、**show mac address-table** コマンドを両方のリーフスイッチの CLI に入力します。

```
leaf2# show mac address-table
Legend:
* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen,+ - primary entry using vPC Peer-Link,
(T) - True, (F) - False
VLAN MAC Address Type age Secure NTFY Ports/SWID.SSID.LID
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
* 19      0050.5695.17b7    dynamic    -       F     F     eth1/31
* 19 0050.5695.248f dynamic - F F eth1/31
```

```
leaf_1# show mac address-table
Legend:
* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen,+ - primary entry using vPC Peer-Link,
(T) - True, (F) - False
VLAN MAC Address Type age Secure NTFY Ports/SWID.SSID.LID
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
27 0050.5695.248f dynamic - F F tunnel7
27 0050.5695.17b7 dynamic - F F tunnel7
* 28      0050.5695.bd89    dynamic    -       F     F     eth1/25
```

出力からわかるように、送信元 IP アドレスは 2 番目のリーフスイッチ (**leaf2**) で学習され、宛先 IP アドレスは最初のリーフスイッチ (**leaf\_1**) で学習されます。これらは異なるリーフスイッチ上にあるので、トラフィックは 2 番目のリーフスイッチの NorthStar ASIC に送信してスパインスイッチへのアップストリームに送信できるようにする必要があります。NorthStar ロジックに従うには、ラインカードの **vsh** に接続します。

ローカル エントリのリストを表示するには、次のコマンドを入力します。

```
leaf2# vsh_lc
module-1# show platform internal ns forwarding 1st-12
error opening file
: No such file or directory

=====
TABLE INSTANCE : 0
=====
Legend:
POS: Entry Position O: Overlay Instance
V: Valid Bit MD/PT: Mod/Port
PT: Pointer Type(A=Adj, E=ECMP, D=DstEncap N=Invalid)
PTR: ECMP/Adj/DstEncap/MET pointer
ML: MET Last
ST: Static PTH: Num Paths
BN: Bounce CP: Copy To CPU
PA: Policy Applied PI: Policy Incomplete
DL: Dst Local SP: Spine Proxy
-----
MO SRC P M S B C P P D S
POS O VNID Address V DE MD/PT CLSS T PTR L T PTH N P A I L P
-----
111 0 fd7f82 00:22:bd:f8:19:ff 1 0 00/00 1 A 0 0 1 1 0 0 0 1 0 0
131 0 f1ffde 00:22:bd:f8:19:ff 1 0 00/00 1 A 0 0 1 1 0 0 0 1 0 0
169 0 f37fd3 00:50:56:95:26:5e 1 0 00/24 4002 A 0 0 0 1 0 0 0 1 0 0
```

```

331 0 f37fd2 00:50:56:95:5c:4d 1 0 00/2e 8003 A 0 0 0 1 0 0 0 1 0 0
719 0 f3ffce 00:22:bd:f8:19:ff 1 0 00/00 1 A 0 0 1 1 0 0 0 1 0 0
945 0 f7ffae 00:22:bd:f8:19:ff 1 0 00/00 1 A 0 0 1 1 0 0 0 1 0 0
1390 0 fa7f9a 00:22:bd:f8:19:ff 1 0 00/00 1 A 0 0 1 1 0 0 0 1 0 0
1454 0 efffee 00:22:bd:f8:19:ff 1 0 00/00 1 A 0 0 1 1 0 0 0 1 0 0
1690 0 f37fd3 00:22:bd:f8:19:ff 1 0 00/00 1 A 0 0 1 1 0 0 0 1 0 0
1720 0 f37fd3 00:50:56:95:c3:6f 1 0 00/24 c002 A 0 0 0 1 0 0 0 1 0 0
1902 0 flffde 00:50:56:95:4e:d3 1 0 00/2e 8006 A 0 0 0 1 0 0 0 1 0 0
2176 0 f07fea 00:50:56:95:17:b7 1 0 00/0f 8004 A 0 0 0 1 0 0 0 0 0 0
2819 0 faff97 00:22:bd:f8:19:ff 1 0 00/00 1 A 0 0 1 1 0 0 0 1 0 0
3297 0 f07fea 00:22:bd:f8:19:ff 1 0 00/00 1 A 0 0 1 1 0 0 0 1 0 0

```

```

=====
TABLE INSTANCE : 1
=====

```

```

Legend:
POS: Entry Position O: Overlay Instance
V: Valid Bit MD/PT: Mod/Port
PT: Pointer Type(A=Adj, E=ECMP, D=DstEncap N=Invalid)
PTR: ECMP/Adj/DstEncap/MET pointer
ML: MET Last
ST: Static PTH: Num Paths
BN: Bounce CP: Copy To CPU
PA: Policy Applied PI: Policy Incomplete
DL: Dst Local SP: Spine Proxy

```

```

-----
MO SRC P M S B C P P D S
POS O VNID Address V DE MD/PT CLSS T PTR L T PTH N P A I L P
-----
169 0 f37fd3 00:50:56:95:26:5e 1 0 00/24 4002 A e 0 0 1 0 0 0 0 1 0
331 0 f37fd2 00:50:56:95:5c:4d 1 0 00/2e 8003 A 9 0 0 1 0 0 0 0 1 0
1720 0 f37fd3 00:50:56:95:c3:6f 1 0 00/24 c002 A c 0 0 1 0 0 0 0 1 0
1902 0 flffde 00:50:56:95:4e:d3 1 0 00/2e 8006 A f 0 0 1 0 0 0 0 1 0
2176 0 f07fea 00:50:56:95:17:b7 1 0 00/0f 8004 A d 0 0 1 0 0 0 0 1 0
3507 0 fa7f9a 00:50:56:95:3e:ee 1 0 00/2e c005 A 10 0 0 1 0 0 0 0 1 0
3777 0 f37fd3 00:50:56:95:68:c4 1 1 04/04 4002 A 11 0 0 1 1 0 0 0 0 0
3921 0 f07fea 00:50:56:95:24:8f 1 0 00/0f 8004 A d 0 0 1 0 0 0 0 1 0

```

宛先エントリのリストを表示する (宛先 MAC アドレスを探す) には、次のコマンドを入力します。

```

module-1# show platform internal ns forwarding gst-12
error opening file
: No such file or directory

```

```

=====
TABLE INSTANCE : 0
=====

```

```

Legend:
POS: Entry Position O: Overlay Instance
V: Valid Bit MD/PT: Mod/Port
PT: Pointer Type(A=Adj, E=ECMP, D=DstEncap N=Invalid)
PTR: ECMP/Adj/DstEncap/MET pointer
ML: MET Last
ST: Static PTH: Num Paths
BN: Bounce CP: Copy To CPU
PA: Policy Applied PI: Policy Incomplete
DL: Dst Local SP: Spine Proxy

```

```

-----
MO SRC P M S B C P P D S
POS O VNID Address V DE MD/PT CLSS T PTR L T PTH N P A I L P
-----
2139 0 ff7f72 00:50:56:95:7b:16 1 0 00/00 8006 A d 0 0 1 0 0 0 0 1 0

```

```

2195 0 faff97 00:50:56:95:5d:6e 1 0 00/00 8005 A f 0 0 1 0 0 0 0 1 0
3379 0 f07fea 00:50:56:95:bd:89 1 1 00/00 8004 A 10 0 0 1 0 0 0 0 0
4143 0 f07fea 00:50:56:95:17:b7 1 0 00/00 8004 A a 0 0 1 0 0 0 0 1 0
4677 0 f07feb 00:50:56:95:68:c4 1 0 00/00 4002 A e 0 0 1 0 0 0 0 1 0
5704 0 f07fea 00:50:56:95:24:8f 1 0 00/00 8004 A a 0 0 1 0 0 0 0 1 0
6191 0 f7ffaf 00:50:56:95:00:33 1 0 00/00 4007 A c 0 0 1 0 0 0 0 1 0

```

次の出力の [Pointer] ( PTR ) フィールドをメモしてください。これが隣接関係ポインタです。この値は、宛先のカプセル化された VLAN を見つけるために次のコマンドで使用されます。これは 16 進数値ですので、10 進数値に変換する必要があります ( 0 x 10 は 10 進数の 16 )。

隣接関係ポインタとして 16 を使用して、次のコマンドを CLI に入力します。

```

module-1# show platform internal ns forwarding adj 16
error opening file
: No such file or directory

```

```

=====
TABLE INSTANCE : 0
=====
Legend
TD: TTL Dec Disable UP: USE PCID
DM: Dst Mac Rewrite SM: Src Mac Rewrite
RM IDX: Router Mac IDX SR: Seg-ID Rewrite
-----
ENCP T U USE D S RM S SRC
POS SEG-ID PTR D P PCI M DST-MAC M IDX R SEG-ID CLSS
-----
16 0 2ffa 0 0 0 1 00:0c:0c:0c:0c:0c 0 0 0 0

```

この出力の ENCP PTR 値をメモします。この値は、宛先トンネル エンドポイント ( TEP ) アドレスを探すために使用されます。

```

module-1# show platform internal ns forwarding encap 0x2ffa
error opening file
: No such file or directory

```

```

=====
TABLE INSTANCE : 0
=====
Legend
MD: Mode (LUX & RWX) LB: Loopback
LE: Loopback ECMP LB-PT: Loopback Port
ML: MET Last TD: TTL Dec Disable
DV: Dst Valid DT-PT: Dest Port
DT-NP: Dest Port Not-PC ET: Encap Type
OP: Override PIF Pinning HR: Higig DstMod RW
HG-MD: Higig DstMode KV: Keep VNTAG
-----
M PORT L L LB MET M T D DT DT E TST O H HG K M E
POS D FTAG B E PT PTR L D V PT NP T IDX P R MD V D T Dst MAC DIP
-----
12282 0 c00 0 1 0 0 0 0 0 0 0 3 7 0 0 0 0 0 3 00:00:00:00:00:00 192.168.56.93

```

この場合、フレームはローカル TEP の送信元 IP アドレスおよびリストされている TEP の宛先 IP アドレスを介して iVXLAN でカプセル化されます。ELTMC 出力に基づくと、その BD の VXLAN ID は 15761386 であるため、これが VXLAN パケットに付加された ID となります。トラフィックが相手側に達するとカプセル化が解除されます。また、宛先 MAC アドレスがローカルであるため、トラフィックは Broadcom からの I2 show コマンドでポートから転送されます。

## 同じリーフ上の各 EPG に 1 つのエンドポイントがある単一 BD / 2 つの EPG

このセクションでは、EPG が異なるが同一の BD を持つ 2 つのエンドポイントのためのハードウェアのプログラミングとパケット フローを確認する方法を説明します。トラフィックは同じリーフスイッチに流れます。これは、物理ローカルから物理ローカル(PL-to-PL)ブリッジパケットとも呼ばれます。ルーティングを行うためにレイヤ 3 ( L3 ) インターフェイスが必要なく、2 つのカプセル化された VLAN 間で通信が許可されるため、ブリッジされたと呼ばれます。

まず確認すべきことは、リーフ スイッチ上の送信元と宛先の IP アドレスに関して、想定されるインターフェイス上で ( この場合は 1/48 ) MAC アドレス情報を学習できるかどうかです。次に、この例で使用される MAC アドレスおよび IP アドレス情報を示します。

- 送信元MACアドレス : 0050.5695.908b
- 送信元 IP アドレス : 192.168.1.50
- 宛先MACアドレス : 0050.5695.bd89
- 宛先IPアドレス : 192.168.1.51

この情報を確認するには、`show mac address-table` コマンドを CLI に入力します。

```
leaf1# show mac address-table | grep 908b
* 34      0050.5695.908b    dynamic -      F      F      eth1/48
leaf1# show mac address-table | grep bd89
* 38      0050.5695.bd89    dynamic -      F      F      eth1/48
```

次に Broadcom ( BCM ) のシェルに入り、BCM が正しい MAC アドレス情報を学習することを確認する必要があります。

```
bcm-shell.0> 12 show
mac=00:50:56:95:bd:89 vlan=55 GPORT=0x30 modid=0 port=48/xe47
mac=00:50:56:95:90:8b vlan=54 GPORT=0x30 modid=0 port=48/xe47 Hit
```

この出力は、BCM が MAC アドレス情報を学習したことを示しています。ただし MAC アドレスは、異なる VLAN にあります。トラフィックが異なるカプセル化された VLAN ( 異なる EPG ) のホストから送信されているため、これは予想どおりです。

ELTMC に入り、2 つのカプセル化された VLAN の BD VLAN に対して BCM シェルで表示される HW\_VlanID を確認します。

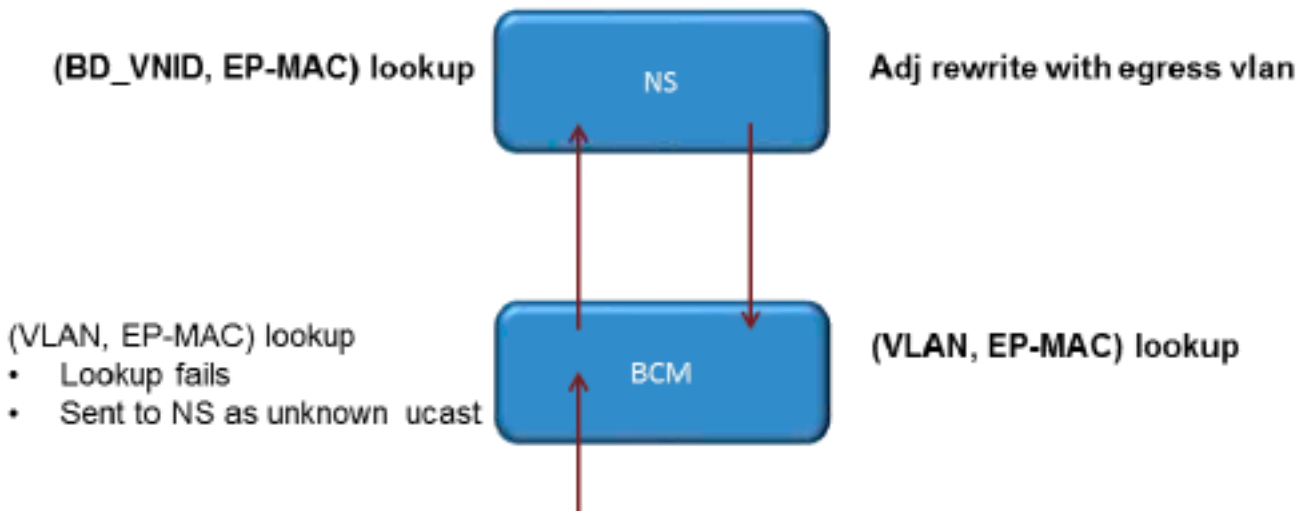
```
module-1# show system internal eltmc info vlan brief
VLAN-Info
VlanId HW_VlanId Type Access_enc Access_enc Fabric_enc Fabric_enc BDVlan
Type Type
=====
13 15 BD_CTRL_VLAN 802.1q 4093 VXLAN 16777209 0
14 16 BD_VLAN Unknown 0 VXLAN 15957970 14
15 17 BD_VLAN Unknown 0 VXLAN 16613250 15
16 18 FD_VLAN 802.1q 301 VXLAN 8509 15
17 19 BD_VLAN Unknown 0 VXLAN 16220082 17
18 46 BD_VLAN Unknown 0 VXLAN 14745592 18
19 50 BD_VLAN Unknown 0 VXLAN 16646015 19
20 51 FD_VLAN 802.1q 502 VXLAN 8794 19
21 23 BD_VLAN Unknown 0 VXLAN 16121792 21
22 24 FD_VLAN 802.1q 538 VXLAN 8830 21
23 25 BD_VLAN Unknown 0 VXLAN 15826915 23
24 28 FD_VLAN 802.1q 537 VXLAN 8829 23
25 26 BD_VLAN Unknown 0 VXLAN 16351138 25
```

```

26 29 FD_VLAN 802.1q 500 VXLAN 8792 25
27 27 BD_VLAN Unknown 0 VXLAN 16678779 27
28 30 FD_VLAN 802.1q 534 VXLAN 8826 27
29 52 BD_VLAN Unknown 0 VXLAN 15859681 29
31 47 FD_VLAN 802.1q 602 VXLAN 9194 18
32 31 FD_VLAN 802.1q 292 VXLAN 8500 55
33 20 BD_VLAN Unknown 0 VXLAN 15761386 33
34      54      FD_VLAN      802.1q      299      VXLAN      8507      54
35 33 BD_VLAN Unknown 0 VXLAN 16449431 35
38      55      FD_VLAN      802.1q      300      VXLAN      8508      54
39 53 FD_VLAN 802.1q 501 VXLAN 8793 29

```

この ELTMC の出力では、各エントリの HW\_VlanId がトラフィックがスイッチに入ったときにタグ付けされる Access\_enc にマッピングされていること (VMware ポートグループが仮想化されているかどうか確認)、および VlanId が MAC アドレステーブルに表示された PI VLAN であることがわかります。この場合 BD VLAN が同一である (両方とも VLAN 54 上にある) ため、ブリッジされた接続となります。次の図は、BCM から NorthStar へのインタラクションを示します。



NorthStarはパケットを調整し、宛先IPアドレスのHW\_VlanIdを使用して出力フレームを書き換えます。これにより、BCMはそのVLANでローカルヒットし、ポート1/48を介して送信します。

## 同じリーフ上の各 EPG に 1 つのエンドポイントがある 2 つの BD / 2 つの EPG ( ルート パケット )

このセクションでは、異なる BD を使用する異なる EPG の 2 つのエンドポイントのためのハードウェアのプログラミングとパケット フローを確認する方法を説明します。トラフィックは同じリーフスイッチに向かいますが、ルーティングが必要です。これは PL-to-PL でルートされたパケットとも呼ばれます。

まず確認すべきことは、リーフスイッチ上の送信元と宛先の IP アドレスに関して、想定されるインターフェイス上で ( この場合は 1/48 ) MAC アドレス情報を学習できるかどうかです。次に、この例で使用される MAC アドレスおよび IP アドレス情報を示します。

- 送信元MACアドレス : 0050.5695.908b
- 送信元 IP アドレス : 192.168.1.50
- デフォルト ゲートウェイ : 192.168.1.1
- 宛先MACアドレス : 0050.5695.bd89
- 宛先IPアドレス : 192.168.3.51



- デフォルト ゲートウェイ : 192.168.3.1

L2 情報は MAC アドレス テーブルで確認できますが、L3 のルーティングされたトラフィックについては、ソリューションの重要な部分はエンドポイント マネージャ ( EPM ) です。 EPM は特定のデバイスのエンドポイントをすべて追跡するプロセスです。

EPM が最初のリーフ スイッチ ( Leaf1 ) の 2 つのエンドポイントを把握できることを確認します

。

```
leaf1# show endpoint ip 192.168.1.50
```

```
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
+-----+-----+-----+-----+-----+-----+-----+-----+
56             vlan-299      0050.5695.908b L             eth1/48
Joey-Tenant:Joey-Internal    vlan-299      192.168.1.50 L
```

送信元 IP アドレスは Ethernet 1/48 で学習され、このスイッチに対してローカルです。

```
leaf1# show endpoint ip 192.168.3.51
```

```
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
+-----+-----+-----+-----+-----+-----+-----+-----+
44             vlan-291      0050.5695.bd89 L             eth1/48
Joey-Tenant:Joey-Internal    vlan-291      192.168.3.51 L
```

ここで示すように、宛先 IP アドレスは Ethernet 1/48 で学習され、このスイッチに対してローカルです。

これらのエンドポイントの詳細情報を取得するには、ラインカード ( LC ) に接続します。

```
leaf1# vsh_lc
```

```
module-1# show system internal epmc endpoint ip 192.168.1.50
```

```
MAC : 0050.5695.908b ::: Num IPs : 1
IP# 0 : 192.168.1.50 ::: IP# 0 flags :
Vlan id : 56 ::: Vlan vnid : 8507 ::: BD vnid : 15990734
VRF vnid : 2523136 ::: phy if : 0x1a02f000 ::: tunnel if : 0
Interface : Ethernet1/48
VTEP tunnel if : N/A ::: Flags : 0x80004c04
Ref count : 5 ::: sclass : 0x2ab5
Timestamp : 02/01/1970 00:43:53.129731
last mv timestamp 12/31/1969 19:00:00.000000 ::: ep move count : 0
previous if : 0 ::: loop detection count : 0
EP Flags : local,IP,MAC,class-set,timer,
Aging:Timer-type : Host-tracker timeout ::: Timeout-left : 423 ::: Hit-bit :
Yes ::: Timer-reset count : 406
```

```
PD handles:
```

```
Bcm l2 hit-bit : Yes
```

```
[L2]: Asic : NS ::: ADJ : 0x14 ::: LST SA : 0x83a ::: LST DA : 0x83a :::
GST ING : 0xedb ::: BCM : Yes
[L3-0]: Asic : NS ::: ADJ : 0x14 ::: LST SA : 0xe56 ::: LST DA : 0xe56 :::
GST ING : 0x12ae ::: BCM : Yes
:::
```

**VRF vnid および BD vnid の値をメモします。**

```
module-1# show system internal epmc endpoint ip 192.168.3.51
```

```
MAC : 0050.5695.bd89 ::: Num IPs : 1
IP# 0 : 192.168.3.51 ::: IP# 0 flags :
Vlan id : 44 ::: Vlan vnid : 8499 ::: BD vnid : 15761386
VRF vnid : 2523136 ::: phy if : 0x1a02f000 ::: tunnel if : 0
Interface : Ethernet1/48
VTEP tunnel if : N/A ::: Flags : 0x80004c04
Ref count : 5 ::: sclass : 0x8004
Timestamp : 02/01/1970 00:43:53.130524
last mv timestamp 12/31/1969 19:00:00.000000 ::: ep move count : 0
previous if : 0 ::: loop detection count : 0
EP Flags : local,IP,MAC,class-set,timer,
Aging:Timer-type : Host-tracker timeout ::: Timeout-left : 532 ::: Hit-bit :
Yes ::: Timer-reset count : 1
```

PD handles:

```
Bcm l2 hit-bit : Yes
[L2]: Asic : NS ::: ADJ : 0x15 ::: LST SA : 0x28e ::: LST DA : 0x28e :::
GST ING : 0xd33 ::: BCM : Yes
[L3-0]: Asic : NS ::: ADJ : 0x15 ::: LST SA : 0x497b ::: LST DA : 0x497b :::
GST ING : 0x1e98 ::: BCM : Yes
:::
```

この出力の VRF vnid 値は、どちらのルートもルーティング テーブルの同じ Virtual Routing and Forwarding ( VRF ) の一部である ( 同一のコンテキスト ) ため、同じです。BD vnid 値は、2 つのエンドポイントが異なる BD にあるため、異なります。

NorthStar テーブルで L2 レベルの MAC アドレスのハードウェア プログラミングを確認したとおり、同じように L3 テーブルを確認することができます。

```
module-1# show platform internal ns forwarding lst-13
```

```
error opening file
: No such file or directory
```

```
=====
TABLE INSTANCE : 0
=====
```

Legend:

```
POS: Entry Position O: Overlay Instance
V: Valid Bit MD/PT: Mod/Port
PT: Pointer Type(A=Adj, E=ECMP, D=DstEncap N=Invalid)
PTR: ECMP/Adj/DstEncap/MET pointer
ML: MET Last
ST: Static PTH: Num Paths
BN: Bounce CP: Copy To CPU
PA: Policy Applied PI: Policy Incomplete
DL: Dst Local SP: Spine Proxy
```

```
-----
MO SRC P M S B C P P D S
POS O VNID Address V DE MD/PT CLSS T PTR L T PTH N P A I L P
-----
```

```
2881 0 268000 192.168.1.1 1 0 00/00 1 A 0 0 1 1 0 0 0 1 0 0
```

```

3003 0 208001 80.80.80.10 1 0 00/14 800d A 0 0 0 1 0 0 0 1 0 0
3051 0 208001 30.30.30.30 1 0 00/14 c009 A 0 0 0 1 0 0 0 0 0 0
3328 0 268000 192.168.2.1 1 0 00/00 1 A 0 0 1 1 0 0 0 1 0 0
3670 0 268000 192.168.1.50      1 0 00/09 2ab5 A  0 0 0  1 0 0 0 0 0 0
3721 0 2b8001 50.50.50.1 1 0 00/00 1 A 0 0 1 1 0 0 0 1 0 0
3903 0 268000 192.168.3.1      1 0 00/00  1 A  0 0 1  1 0 0 0 1 0 0
18811 0 268000 192.168.3.51 1 0 00/09 8004 A 0 0 0 1 0 0 0 0 0 0

```

次の図は、ASIC を通じたフローを示します。

