Troubleshoot ACI Intra-Fabric Forwarding - Layer 2 Forwarding

Contents

Introduction

This document describes steps to understand and troubleshoot Layer 2 Forwarding in ACI

Background Information

The material from this document was extracted from the Troubleshooting Cisco Application Centric Infrastructure, Second Edition book, specifically the **Intra-Fabric fowarding - L2 forwarding: two endpoints in same BD - no unicast routing** chapter.

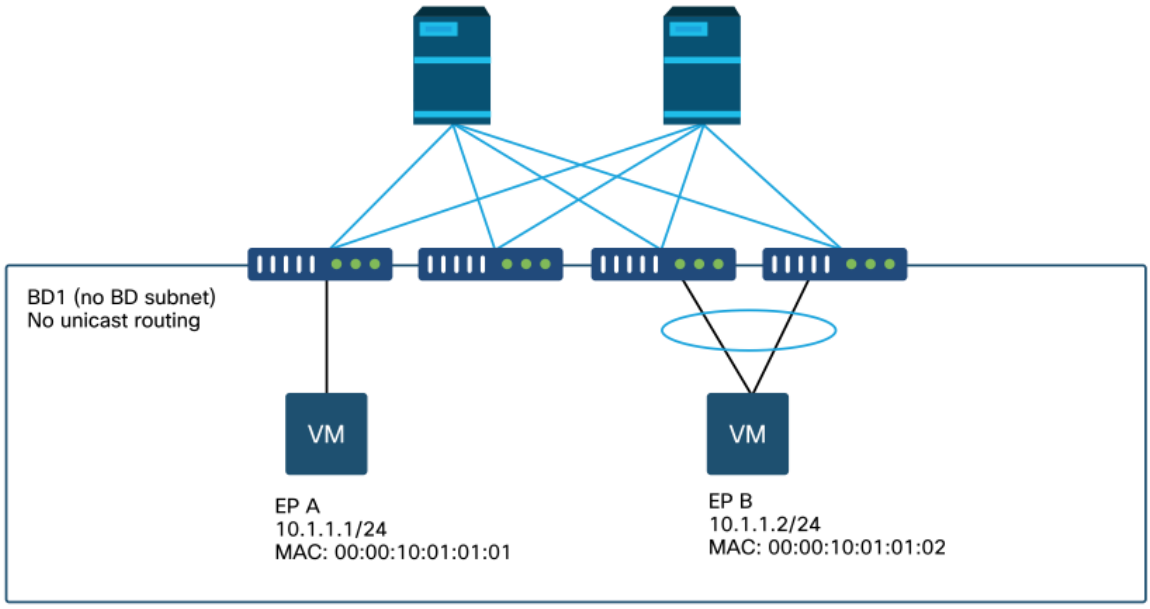
Overview

This section explains a troubleshooting example where endpoints in the same bridge domain and

same subnet can't talk to each other. The figure below illustrates the topology where the BD doesn't have any subnets and has unicast routing disabled.

Typically, when troubleshooting traffic flows with endpoint connectivity, the suggestion is to start identifying a pair of endpoints. Refer to the topology below with EPs A and B. These will respectively have IP addresses 10.1.1.1/24 and 10.1.1.2/24. The MAC addresses will respectively be 00:00:10:01:01:01 and 00:00:10:01:01:02.

Topology



In this section there are three scenarios:

1. Known Layer 2 unicast flow.
2. Unknown Layer 2 unicast flow with BD in flood mode.
3. Unknown Layer 2 unicast flow with BD in hardware-proxy mode.

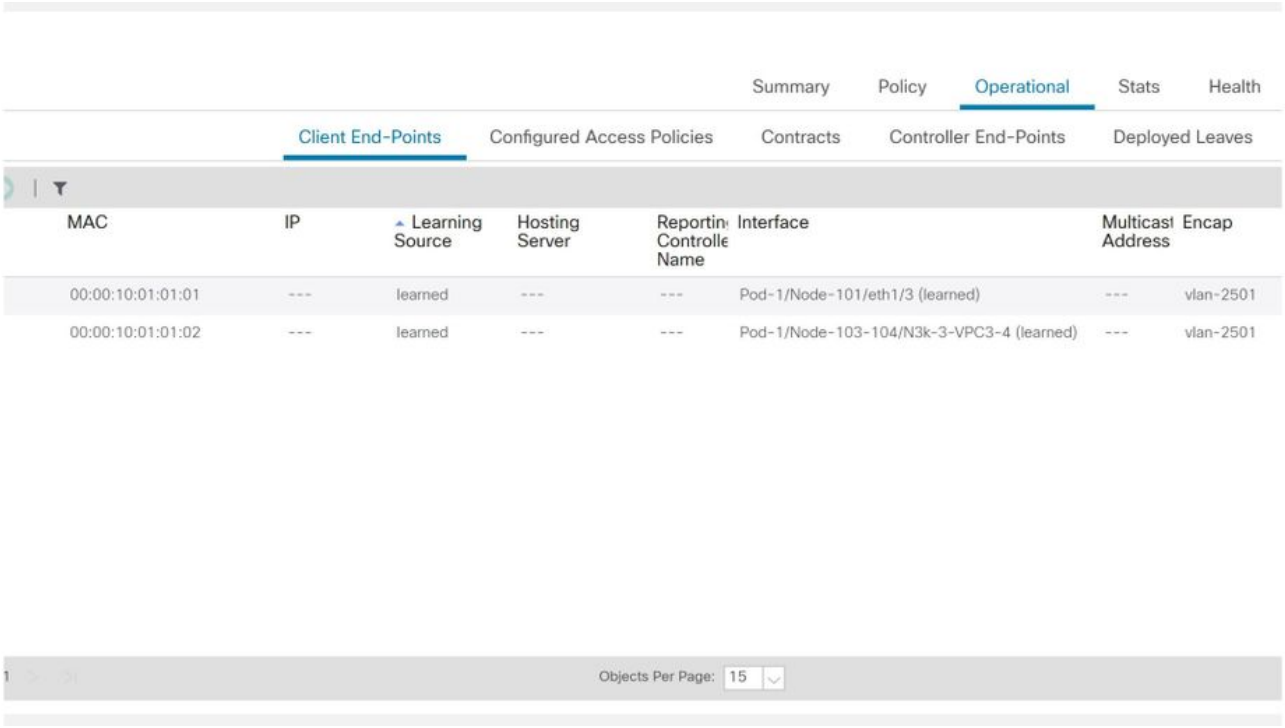
The troubleshooting flows that will be followed can be summarized by the following scheme:

- Level 1 check: GUI validation of the config, faults and endpoints learned.
- Level 2 check: CLI on the leaf switches: Check if the source and destination leaf switches learn the endpoints.Check if spine nodes learn the endpoint in COOP.
- Level 3 check: packet capture: ELAM (ELAM Assistant or CLI) to validate the frame is there.fTriage to track the flow.

GUI check

The first level of troubleshooting is validating from the GUI that the endpoint MAC was learned properly. This can be done from the operational tab of the EPG where the endpoint sits.

'EPG Operational tab > Client End-Points'



In this scenario, both endpoints A and B are shown in the GUI. The GUI shows their MAC addresses, the interface where they are connected to the fabric, and the encapsulation — in this case both are in encap VLAN 2501.

It is expected that the IP address isn't learnt from the ACI fabric as the unicast routing has been disabled at the BD level.

Refer to the learning source column in the screenshot above. If it denotes 'learned', the ACI leaf switch received at least one packet from the endpoint.

Since in this case the endpoints are learnt from the ACI fabric, move on to the next troubleshooting case for known Layer 2 unicast traffic.

Troubleshooting workflow for known Layer 2 unicast traffic

Ingress leaf source EP MAC learning

In case of Layer 2 forwarding in the same BD, ACI will only learn the source MAC and forward based on the destination MAC. MAC addresses are learnt in the scope of the BD.

First, check if the endpoint is learned:

```
leaf1# show endpoint  mac  0000.1001.0101
Legend:
s - arp              H - vtep             V - vpc-attached     p - peer-aged
R - peer-attached-rl B - bounce           S - static           M - span
D - bounce-to-proxy  O - peer-attached    a - local-aged       m - svc-mgr
L - local            E - shared-service
+----------------------------------+---------------+-----------------+--------------+----------
---+
    VLAN/                           Encap           MAC Address       MAC Info/       Interface
    Domain                          VLAN            IP Address        IP Info
+----------------------------------+---------------+-----------------+--------------+----------
```

```
---+
4/Prod:VRF1                              vlan-2501    0000.1001.0101 L
eth1/3
```

The above output gives the following information:

- MAC address 0000.1001.0101 is learnt locally (Flag is L for local) on port ethernet 1/3 with encapsulation vlan-2501 in vrf Prod:VRF1.
- Refer to the 'VLAN/Domain' column in the above output. The VLAN ID listed there is the internal VLAN.

Ingress leaf destination MAC endpoint lookup

Assume the destination MAC is known (known unicast).

```
leaf1# show endpoint mac 0000.1001.0102
Legend:
s - arp              H - vtep             V - vpc-attached     p - peer-aged
R - peer-attached-rl B - bounce           S - static           M - span
D - bounce-to-proxy  O - peer-attached    a - local-aged       m - svc-mgr
L - local            E - shared-service
+-----------------------------------+---------------+-----------------+--------------+----------
---+
    VLAN/                           Encap           MAC Address       MAC Info/       Interface
    Domain                          VLAN            IP Address        IP Info
+-----------------------------------+---------------+-----------------+--------------+----------
---+
7/Prod:VRF1                          vxlan-16351141    0000.1001.0102
tunnel4
```

The above output gives the following information:

- MAC address 0000.1001.0102 is not learned locally.
- It is learned from interface tunnel 4.
- It is learned in encapsulation VXLAN-16351141 which corresponds to the BD_VNID (VXLAN Network ID) of the bridge domain.

Next, check the destination of the tunnel interface using the 'show interface tunnel <x>' command

```
leaf1# show interface tunnel 4
Tunnel4 is up
  MTU 9000 bytes, BW 0 Kbit
  Transport protocol is in VRF "overlay-1"
  Tunnel protocol/transport is ivxlan
  Tunnel source 10.0.88.95/32 (lo0)
  Tunnel destination 10.0.96.66
  Last clearing of "show interface" counters never
  Tx
  0 packets output, 1 minute output rate 0 packets/sec
  Rx
  0 packets input, 1 minute input rate 0 packets/sec
```

So, the packet will be encapsulated in VXLAN with source TEP IP 10.0.88.95 (assigned to loopback0) and sent towards the destination TEP IP 10.0.96.66.

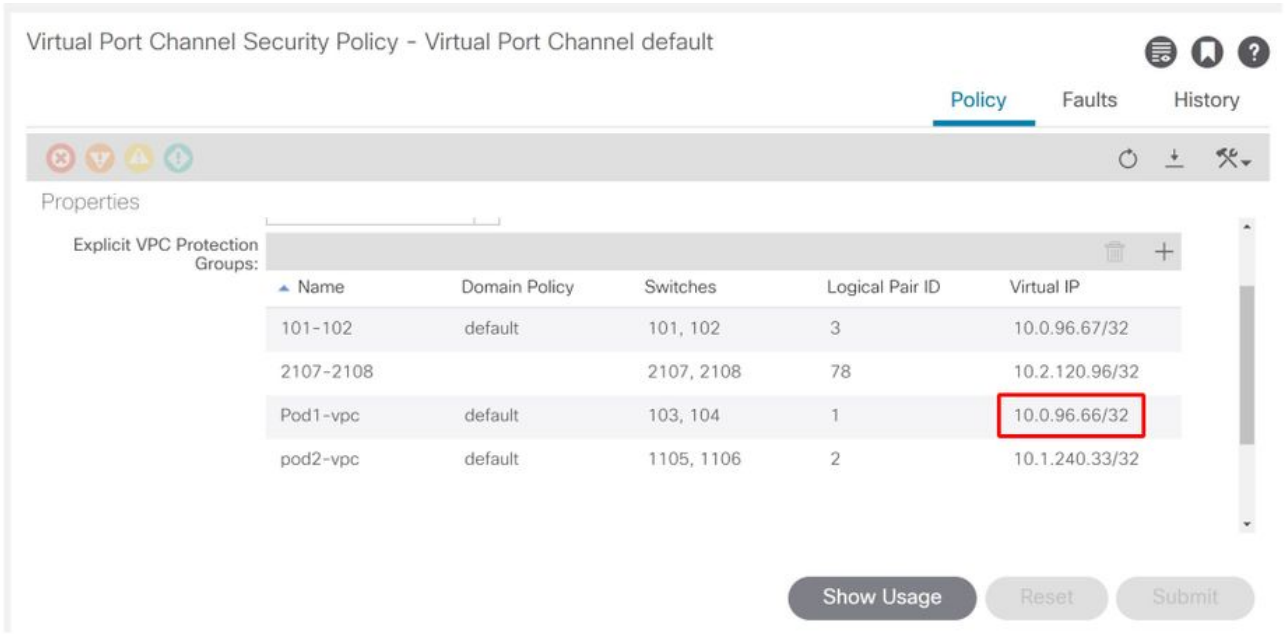
Confirm the source IP:

```
leaf1# show ip interface loopback 0 vrf overlay-1
IP Interface Status for VRF "overlay-1"
lo0, Interface status: protocol-up/link-up/admin-up, iod: 4, mode: ptep
 IP address: 10.0.88.95, IP subnet: 10.0.88.95/32
 IP broadcast address: 255.255.255.255
 IP primary address route-preference: 0, tag: 0
```

The destination TEP IP 10.0.96.66 can be one of the following:

- PTEP address of another leaf (can be checked using acidiag fnvread)
- VPC VIP (can be seen in 'GUI > Fabric > Access Policies > Policies > Switch > Virtual Port Channel default' (see screenshot below)
- Some loopback IP on a spine switch. Use 'show ip interface vrf overlay-1' command on the spine switch to verify this.

Explicit VPC Protection Groups



Ingress leaf switch sending to spine switch

The ingress leaf will now encapsulate the frame into VXLAN with the outer destination IP set to 10.0.96.66 which is the tunnel destination IP listed in the previous 'show interface tunnel 4' command. It will encapsulate it in VXLAN with the VNID of the bridge domain - vxlan-16351141 - as shown in the previous 'show endpoint mac 0000.1001.0102' command output.

Based on the IS-IS route in VRF overlay-1 determine where to send it:

```
leaf1# show ip route 10.0.96.66 vrf overlay-1
IP Route Table for VRF "overlay-1"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.0.96.66/32, ubest/mbest: 4/0
  *via 10.0.88.65, Eth1/49.10, [115/3], 2w5d, isis-isis_infra, isis-l1-int
  *via 10.0.88.94, Eth1/50.128, [115/3], 2w5d, isis-isis_infra, isis-l1-int
```

So, there is ECMP (equal cost multipath) routing to the destination using eth1/49 and 1/50 which

are the fabric uplinks to the spine switches.

Spine forwarding

The VRF overlay-1 routing table on the spine shows that host route 10.0.96.66 is reachable via either to leaf3 or leaf4. This is expected as 10.0.96.66 is the VPC VIP of leaf switches 103 and 104:

```
spine1# show ip route 10.0.96.66 vrf overlay-1
IP Route Table for VRF "overlay-1"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.0.96.66/32, ubest/mbest: 2/0
  *via 10.0.88.91, eth1/3.35, [115/2], 02w05d, isis-isis_infra, isis-l1-int
  *via 10.0.88.90, eth1/4.39, [115/2], 02w05d, isis-isis_infra, isis-l1-int

spine1# show lldp neighbors | egrep "1\/3 |1\/4 "
leaf3             Eth1/3          120        BR          Eth1/49
leaf4             Eth1/4          120        BR          Eth1/49
```

Egress leaf remote EP MAC learning

In this case, the destination TEP is a VPC pair so the packet will arrive on either leaf3 or leaf4. Refer to the command outputs below. Leaf4 should show similar output. Given they are part of the same VPC pair, all endpoints are synchronized between the two leaf switches.

Endpoint learning for Layer 2 traffic on the egress leaf is based on the source MAC address which is learned in the BD corresponding to the VNID in the received packet. This can be verified in the endpoint table.

The source MAC address lies behind tunnel 26 in VXLAN-16351141.

Tunnel 26 goes to TEP IP 10.0.88.95 which is leaf1:

```
leaf3# show endpoint  mac 0000.1001.0101
Legend:
s - arp              H - vtep             V - vpc-attached     p - peer-aged
R - peer-attached-rl B - bounce           S - static           M - span
D - bounce-to-proxy  O - peer-attached    a - local-aged       m - svc-mgr
L - local            E - shared-service
+------------------------------------+---------------+-----------------+--------------+----------
---+
    VLAN/                           Encap           MAC Address       MAC Info/       Interface
    Domain                          VLAN            IP Address        IP Info
+------------------------------------+---------------+-----------------+--------------+----------
---+
136/Prod:VRF1                        vxlan-16351141    0000.1001.0101
tunnel26

leaf3# show interface tunnel 26
Tunnel26 is up
  MTU 9000 bytes, BW 0 Kbit
  Transport protocol is in VRF "overlay-1"
```

```
  Tunnel protocol/transport is ivxlan
  Tunnel source 10.0.88.91/32 (lo0)
  Tunnel destination 10.0.88.95
  Last clearing of "show interface" counters never
  Tx
  0 packets output, 1 minute output rate 0 packets/sec
  Rx
  0 packets input, 1 minute input rate 0 packets/sec

leaf3# acidiag fnvread | egrep "10.0.88.95"
   101        1                leaf1      FDO20160TPA      10.0.88.95/32    leaf
active   0
```

Egress leaf destination MAC lookup

The 'show endpoint' command confirms the destination MAC is learned behind port-channel 1 and uses encapsulation VLAN-2501

```
leaf3# show endpoint mac 0000.1001.0102
Legend:
s - arp              H - vtep             V - vpc-attached     p - peer-aged
R - peer-attached-rl B - bounce           S - static           M - span
D - bounce-to-proxy  O - peer-attached    a - local-aged       m - svc-mgr
L - local            E - shared-service
+-----------------------------------+---------------+-----------------+--------------+----------
---+
    VLAN/                           Encap           MAC Address       MAC Info/       Interface
    Domain                          VLAN            IP Address        IP Info
+-----------------------------------+---------------+-----------------+--------------+----------
---+
135/Prod:VRF1                             vlan-2501    0000.1001.0102 LpV
po1
```

This indicates that the frame is leaving the ACI fabric on leaf3 interface port-channel 1 with encap VLAN ID 2501. You can find the BD VNID under the Tenant Operational tab in the GUI.

Validate both endpoints are learned properly in the spine switch COOP EP repo

The COOP EP repo should be synchronized across all the spine nodes. the COOP EP repo can be checked using the BD VNID as a key and entering the EP MAC address.

The source MAC address of this flow is learned from tunnel next-hop 10.0.88.95 which is the TEP IP of leaf1. Additionally, the command output shows VNID 16351141 which corresponds to the correct bridge domain.

```
spine1# show coop internal info repo ep key 16351141 00:00:10:01:01:01

Repo Hdr Checksum : 24197
Repo Hdr record timestamp : 10 01 2019 10:16:50 278195866
Repo Hdr last pub timestamp : 10 01 2019 10:16:50 283699467
Repo Hdr last dampen timestamp : 01 01 1970 00:00:00 0
Repo Hdr dampen penalty : 0
Repo Hdr flags : IN_OBJ EXPORT ACTIVE
EP bd vnid : 16351141
EP mac :  00:00:10:01:01:01
flags : 0x80
repo flags : 0x122
Vrf vnid : 2097154
```

```
Epg vnid : 0
EVPN Seq no : 0
Remote publish timestamp: 01 01 1970 00:00:00 0
Snapshot timestamp: 10 01 2019 10:16:50 278195866
Tunnel nh : 10.0.88.95
MAC Tunnel  : 10.0.88.95
IPv4 Tunnel : 10.0.88.95
IPv6 Tunnel : 10.0.88.95
ETEP Tunnel : 0.0.0.0
```

The destination MAC of this flow is learned against the VPC VIP 10.0.96.66 of leaf3 and leaf4. The EP BD VNID 16351141 is listed as well, which corresponds to the correct BD.

```
spine1# show coop internal info repo ep key 15302583 00:00:10:01:01:02

Repo Hdr Checksum : 16897
Repo Hdr record timestamp : 10 01 2019 11:05:46 351360334
Repo Hdr last pub timestamp : 10 01 2019 11:05:46 352019546
Repo Hdr last dampen timestamp : 01 01 1970 00:00:00 0
Repo Hdr dampen penalty : 0
Repo Hdr flags : IN_OBJ EXPORT ACTIVE
EP bd vnid : 16351141
 EP mac :  00:00:10:01:01:02
flags : 0x90
repo flags : 0x122
Vrf vnid : 2097154
Epg vnid : 0
EVPN Seq no : 0
Remote publish timestamp: 01 01 1970 00:00:00 0
Snapshot timestamp: 10 01 2019 11:05:46 351360334
Tunnel nh : 10.0.96.66
MAC Tunnel  : 10.0.96.66
IPv4 Tunnel : 10.0.96.66
IPv6 Tunnel : 10.0.96.66
ETEP Tunnel : 0.0.0.0
```

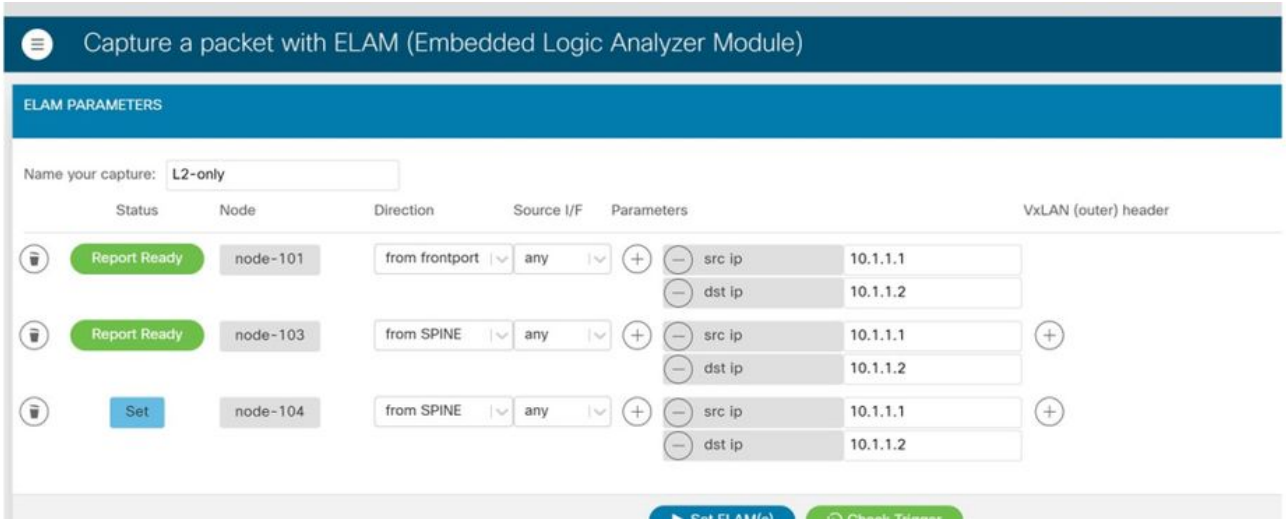
ELAM output using ELAM Assistant

ELAM Assistant is a powerful ACI App which can simplify the execution of ELAM captures on an ACI fabric.

ELAM Assistant triggers can be started simultaneously on multiple leaf nodes. As a result, specific packets can be checked in parallel in leaf1, leaf3 and leaf4.

The configured ELAM capture will appear as shown below. As observed, the packet is seen on leaf1 (node-101) and leaf3 (node-103).

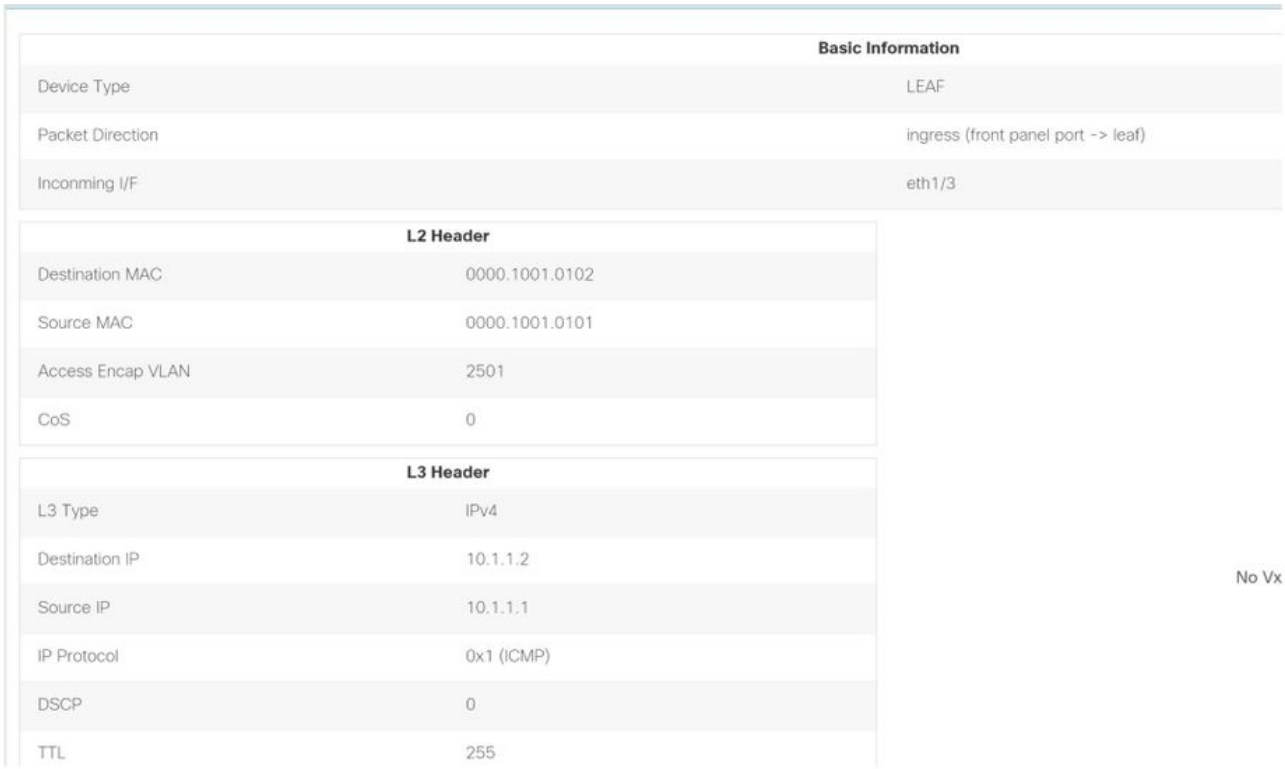
ELAM Assistant — parameters



The report of leaf1 (node-101) shows the following:

- The Captured Packet Information output confirms the packet enters on eth1/3 and has the correct MAC and IP information.
- The packet forwarding information shows it's forwarded on eth1/49 to TEP IP 10.0.96.66.

ELAM Assistant — leaf1 (node-101) — Captured Packet Information



Basic Information	
Device Type	LEAF
Packet Direction	ingress (front panel port -> leaf)
Inconming I/F	eth1/3

L2 Header	
Destination MAC	0000.1001.0102
Source MAC	0000.1001.0101
Access Encap VLAN	2501
CoS	0

L3 Header	
L3 Type	IPv4
Destination IP	10.1.1.2
Source IP	10.1.1.1
IP Protocol	0x1 (ICMP)
DSCP	0
TTL	255

No Vx

ELAM Assistant — leaf1 (node-101) — Packet Forwarding Information

Packet Forwarding Information

	Forward Result
Destination Type	To another ACI node (or AVS/AVE)
Destination TEP	10.0.96.66 (vPC (103_104))
Destination Physical Port	eth1/49
Sent to SUP/CPU instead	no
SUP Redirect Reason (SUP code)	NONE

	Contract
Destination EPG pcTag (dclass)	32770 (Prod:App:EPG1)
Source EPG pcTag (sclass)	32770 (Prod:App:EPG1)
Contract was applied	1 (Contract was applied on this node)

	Drop

On leaf3 (node-103) on the egress leaf, the following is observed:

In the Captured Packet Information on leaf3, it enters from eth1/49. The outer IP address confirms the following:

- Source TEP: 10.0.88.95
- Destination TEP: 10.0.96.66
- VNID: 16351141 (BD VNID)

ELAM Assistant — leaf3 (node-103) — Captured Packet Information

Captured Packet Information

	Basic Information
Device Type	LEAF
Packet Direction	egress (spine LC -> leaf)
Inconming I/F	eth1/49

L3 Header (Outer VxLAN)	
L3 Type	IPv4
Destination IP	10.0.96.66 (vPC (103_104))
Source IP	10.0.88.95 (bdsol-aci32-leaf1)
IP Protocol	0x11 (UDP)
DSCP	0
TTL	31
Don't Fragment Bit	0x0 (0x0)

L4 Header (Outer VxLAN)	
L4 Type	iVxLAN
DL (Don't Learn) Bit	0 (not set)
Src Policy Applied Bit	1 (Contract was applied on the previous node)
Dst Policy Applied Bit	1 (Contract was applied on the previous node)
Source EPG (sclass / src pcTag)	0x8002 / 32770 (Prod:App:EPG1)
VRF/BD VNID	15302583 (Prod:BD1)

The Packet Forwarding Information shows the traffic is forwarded on port-channel 1 and specifically ethernet 1/12.

Packet Forwarding Information

	Forward Result
Destination Type	To a local port
Destination Logical Port	Po1
Destination Physical Port	eth1/12
Sent to SUP/CPU instead	no
SUP Redirect Reason (SUP code)	NONE

	Contract
Destination EPG pcTag (dclass)	32770 (Prod:App:EPG1)
Source EPG pcTag (sclass)	32770 (Prod:App:EPG1)
Contract was applied	1 (Contract was applied on this node)

	Drop
Drop Code	no drop

Ingress leaf ELAM using CLI

It is recommended to use ELAM Assistant as it simplifies the operation of running ELAM captures. However, it is also possible to use CLI commands on ACI switches to generate an ELAM report. Below is an example of how this would be done.

Use the trigger sequence shown to capture the packet on the ingress leaf. Refer to the "Tools" section for more info regarding ELAM options.

- In this example, the ASIC is 'tah' as the leaf (part number ending '-EX').
- 'in-select 6' is used to capture a packet coming from a downlink port without a VXLAN encap.
- 'out-select 1' ensures the drop vector is also shown (in case of a packet drop).
- The 'reset' command is needed to make sure any previous triggers have been cleaned.
- Even though this is a bridged flow ELAM has visibility into the IP header. As a result, 'ipv4 src_ip' and 'dst_ip' can be used to set up the trigger.

```
module-1# debug platform internal tah elam asic 0
module-1(DBG-elam)# trigger init in-select ?
 10  Outerl4-innerl4-ieth
 13  Outer(l2|l3|l4)-inner(l2|l3|l4)-noieth
 14  Outer(l2(vntag)|l3|l4)-inner(l2|l3|l4)-ieth
 15  Outer(l2|l3|l4)-inner(l2|l3|l4)-ieth
 6   Outerl2-outerl3-outerl4
 7   Innerl2-innerl3-innerl4
 8   Outerl2-innerl2-ieth
 9   Outerl3-innerl3

module-1(DBG-elam)# trigger init in-select 6 out-select 1
module-1(DBG-elam-insel6)# reset
module-1(DBG-elam-insel6)# set outer ipv4 src_ip 10.1.1.1 dst_ip 10.1.1.2
module-1(DBG-elam-insel6)# start
```

To see if the packet was received, check the ELAM status. If there is a trigger, that means a packet matching the conditions was caught.

```
module-1(DBG-elam-insel6)# status
ELAM STATUS
===========
Asic 0 Slice 0 Status Triggered
Asic 0 Slice 1 Status Armed
```

The next output shows the report is displayed using the 'ereport' command. The output is very long, so only the beginning is pasted here. But note that the full report is saved for later analysis in a location in the leaf file system. The file name also contains the timestamps when the ELAM was taken.

```
leaf1# ls -al /var/log/dme/log/elam_2019-09-30-03m-23h-14s.txt
-rw-rw-rw- 1 root root 699106 Sep 30 23:03 /var/log/dme/log/elam_2019-09-30-03m-23h-14s.txt
```

The 'ereport' validates the packet has been received and the information is as expected (source and destination MAC, source, and destination IP, etc.)

```
module-1(DBG-elam-insel6)# ereport
Python available. Continue ELAM decode with LC Pkg
ELAM REPORT
```

```
================================================================================================
===========
                                                    Trigger/Basic Information
================================================================================================
===========
ELAM Report File                        : /tmp/logs/elam_2019-09-30-03m-23h-14s.txt
In-Select Trigger                       : Outerl2-outerl3-outerl4( 6 )
Out-Select Trigger                      : Pktrw-sideband-drpvec( 1 )
ELAM Captured Device                    : LEAF
Packet Direction                        : ingress
Triggered ASIC type                     : Sugarbowl
Triggered ASIC instance                 : 0
Triggered Slice                         : 0
Incoming Interface                      : 0x24( 0x24 )
( Slice Source ID(Ss) in "show plat int hal l2 port gpd" )

================================================================================================
===========
                                                        Captured Packet

------------------------------------------------------------------------------------------------
-----------
Outer Packet Attributes
------------------------------------------------------------------------------------------------
-----------
Outer Packet Attributes       : l2uc ipv4 ip ipuc ipv4uc
Opcode                        : OPCODE_UC

------------------------------------------------------------------------------------------------
-----------
Outer L2 Header
------------------------------------------------------------------------------------------------
-----------
Destination MAC               : 0000.1001.0102
Source MAC                    : 0000.1001.0101
802.1Q tag is valid           : yes( 0x1 )
CoS                           : 0( 0x0 )
Access Encap VLAN             : 2501( 0x9C5 )

------------------------------------------------------------------------------------------------
-----------
Outer L3 Header
------------------------------------------------------------------------------------------------
-----------
L3 Type                       : IPv4
IP Version                    : 4
DSCP                          : 0
IP Packet Length              : 84 ( = IP header(28 bytes) + IP payload )
Don't Fragment Bit            : not set
TTL                           : 255
IP Protocol Number            : ICMP
IP CheckSum                   : 51097( 0xC799 )
Destination IP                : 10.1.1.2
Source IP                     : 10.1.1.1

================================================================================================
===========
                                                            Forwarding Lookup ( FPB )
================================================================================================
===========
------------------------------------------------------------------------------------------------
-----------
Destination MAC (Lookup Key)
```

```
-------------------------------------------------------------------------------------------------
-----------
Dst MAC Lookup was performed            : yes
Dst MAC Lookup BD                       : 522( 0x20A )
( Hw BDID in "show plat int hal l2 bd pi" )
Dst MAC Address                         : 0000.1001.0102

-------------------------------------------------------------------------------------------------
-----------
Destination MAC (Lookup Result)
-------------------------------------------------------------------------------------------------
-----------
Dst MAC is Hit                          : yes
Dst MAC is Hit Index                    : 6443( 0x192B )
( phy_id in "show plat int hal objects ep l2 mac (MAC) extensions" )
or ( HIT IDX in "show plat int hal l3 nexthops" for L3OUT/L3 EP)
.....
```

Using fTriage to follow the flow

fTriage is run from an APIC CLI and can be used to follow the full path through the ACI fabric. Specify at least the ingress leaf (node-101), the source IP and the destination IP. In this specific case it's a bridged (Layer 2) flow, so the fTriage bridge option is to be used.

Note that fTriage generates a log file in the current directory. This log file will contain all logs and ELAM reports gathered. This allows the packet to be captured at every hop. The short version of the output is below:

```
apic1# ftriage bridge -ii LEAF:101 -sip 10.1.1.1 -dip 10.1.1.2
fTriage Status: {"dbgFtriage": {"attributes": {"operState": "InProgress", "pid": "12181",
"apicId": "1", "id": "0"}}}
Starting ftriage
Log file name for the current run is: ftlog_2019-10-01-18-53-24-125.txt
2019-10-01 18:53:24,129 INFO     /controller/bin/ftriage bridge -ii LEAF:101 -sip 10.1.1.1 -dip
10.1.1.2
2019-10-01 18:53:49,280 INFO     ftriage:     main:1165 Invoking ftriage with default password
and default username: apic#fallback\\admin
2019-10-01 18:54:10,204 INFO     ftriage:     main:839  L2 frame Seen on leaf1 Ingress: Eth1/3
Egress: Eth1/49 Vnid: 15302583
2019-10-01 18:54:10,422 INFO     ftriage:     main:242  ingress encap string vlan-2501
2019-10-01 18:54:10,427 INFO     ftriage:     main:271  Building ingress BD(s), Ctx
2019-10-01 18:54:12,288 INFO     ftriage:     main:294  Ingress BD(s) Prod:BD1
2019-10-01 18:54:12,288 INFO     ftriage:     main:301  Ingress Ctx: Prod:VRF1
2019-10-01 18:54:12,397 INFO     ftriage:   pktrec:490  leaf1: Collecting transient losses
snapshot for LC module: 1
2019-10-01 18:54:30,079 INFO     ftriage:     main:933  SMAC 00:00:10:01:01:01 DMAC
00:00:10:01:01:02
2019-10-01 18:54:30,080 INFO     ftriage:  unicast:973  leaf1: <- is ingress node
2019-10-01 18:54:30,320 INFO     ftriage:  unicast:1215 leaf1: Dst EP is remote
2019-10-01 18:54:31,155 INFO     ftriage:     misc:659  leaf1: L2 frame getting bridged in SUG
2019-10-01 18:54:31,380 INFO     ftriage:     misc:657  leaf1: Dst MAC is present in SUG L2 tbl
2019-10-01 18:54:31,826 INFO     ftriage:     misc:657  leaf1: RwDMAC DIPo(10.0.96.66) is one of
dst TEPs ['10.0.96.66']
2019-10-01 18:56:16,249 INFO     ftriage:     main:622  Found peer-node spine1 and IF: Eth1/1 in
candidate list
2019-10-01 18:56:21,346 INFO     ftriage:     node:643  spine1: Extracted Internal-port GPD Info
for lc: 1
2019-10-01 18:56:21,348 INFO     ftriage:     fcls:4414 spine1: LC trigger ELAM with IFS: Eth1/1
Asic :0 Slice: 0 Srcid: 32
2019-10-01 18:56:54,424 INFO     ftriage:     main:839  L2 frame Seen on spine1 Ingress: Eth1/1
Egress: LC-1/0 FC-24/0 Port-0 Vnid: 15302583
```

```
2019-10-01 18:56:54,424 INFO     ftriage:   pktrec:490  spine1: Collecting transient losses
snapshot for LC module: 1
2019-10-01 18:57:15,093 INFO     ftriage:      fib:332  spine1: Transit in spine
2019-10-01 18:57:21,394 INFO     ftriage:  unicast:1252 spine1: Enter dbg_sub_nexthop with
Transit inst: ig infra: False glbs.dipo: 10.0.96.66
2019-10-01 18:57:21,508 INFO     ftriage:  unicast:1417 spine1: EP is known in COOP (DIPo =
10.0.96.66)
2019-10-01 18:57:25,537 INFO     ftriage:  unicast:1458 spine1: Infra route 10.0.96.66 present
in RIB
2019-10-01 18:57:25,537 INFO     ftriage:     node:1331 spine1: Mapped LC interface: LC-1/0 FC-
24/0 Port-0 to FC interface: FC-24/0 LC-1/0 Port-0
2019-10-01 18:57:30,616 INFO     ftriage:     node:460  spine1: Extracted GPD Info for fc: 24
2019-10-01 18:57:30,617 INFO     ftriage:     fcls:5748 spine1: FC trigger ELAM with IFS: FC-
24/0 LC-1/0 Port-0 Asic :0 Slice: 2 Srcid: 0
2019-10-01 18:57:49,611 INFO     ftriage:  unicast:1774 L2 frame Seen on FC of node: spine1
with Ingress: FC-24/0 LC-1/0 Port-0  Egress: FC-24/0 LC-1/0 Port-0 Vnid: 15302583
2019-10-01 18:57:49,611 INFO     ftriage:   pktrec:487  spine1: Collecting transient losses
snapshot for FC module: 24
2019-10-01 18:57:53,110 INFO     ftriage:     node:1339 spine1: Mapped FC interface: FC-24/0 LC-
1/0 Port-0 to LC interface: LC-1/0 FC-24/0 Port-0
2019-10-01 18:57:53,111 INFO     ftriage:  unicast:1474 spine1: Capturing Spine  Transit pkt-
type L2 frame on egress LC on Node: spine1 IFS: LC-1/0 FC-24/0 Port-0
2019-10-01 18:57:53,530 INFO     ftriage:     fcls:4414 spine1: LC trigger ELAM with IFS: LC-1/0
FC-24/0 Port-0 Asic :0 Slice: 0 Srcid: 64
2019-10-01 18:58:26,497 INFO     ftriage:  unicast:1510 spine1: L2 frame Spine egress  Transit
pkt Seen on spine1 Ingress: LC-1/0 FC-24/0 Port-0 Egress: Eth1/3 Vnid: 15302583
2019-10-01 18:58:26,498 INFO     ftriage:   pktrec:490  spine1: Collecting transient losses
snapshot for LC module: 1
2019-10-01 18:59:28,634 INFO     ftriage:     main:622  Found peer-node leaf3 and IF: Eth1/49 in
candidate list
2019-10-01 18:59:39,235 INFO     ftriage:     main:839  L2 frame Seen on leaf3 Ingress: Eth1/49
Egress: Eth1/12 (Po1) Vnid: 11364
2019-10-01 18:59:39,350 INFO     ftriage:   pktrec:490  leaf3: Collecting transient losses
snapshot for LC module: 1
2019-10-01 18:59:54,373 INFO     ftriage:     main:522  Computed egress encap string vlan-2501
2019-10-01 18:59:54,379 INFO     ftriage:     main:313  Building egress BD(s), Ctx
2019-10-01 18:59:57,152 INFO     ftriage:     main:331  Egress Ctx Prod:VRF1
2019-10-01 18:59:57,153 INFO     ftriage:     main:332  Egress BD(s): Prod:BD1
2019-10-01 18:59:59,230 INFO     ftriage:  unicast:1252 leaf3: Enter dbg_sub_nexthop with Local
inst: eg infra: False glbs.dipo: 10.0.96.66
2019-10-01 18:59:59,231 INFO     ftriage:  unicast:1257 leaf3: dbg_sub_nexthop invokes
dbg_sub_eg for vip
2019-10-01 18:59:59,231 INFO     ftriage:  unicast:1784 leaf3: <- is egress node
2019-10-01 18:59:59,377 INFO     ftriage:  unicast:1833 leaf3: Dst EP is local
2019-10-01 18:59:59,378 INFO     ftriage:     misc:657  leaf3: EP if(Po1) same as egr if(Po1)
2019-10-01 18:59:59,378 INFO     ftriage:     misc:659  leaf3: L2 frame getting bridged in SUG
2019-10-01 18:59:59,613 INFO     ftriage:     misc:657  leaf3: Dst MAC is present in SUG L2 tbl
2019-10-01 19:00:06,122 INFO     ftriage:     main:961  Packet is Exiting fabric with peer-
device: n3k-3 and peer-port: Ethernet1/16
```

Troubleshooting workflow for unknown Layer 2 unicast traffic — BD in flood mode

In this example, the destination MAC is unknown. The destination MAC lookup on the ingress leaf shows no output.

```
leaf1# show endpoint mac 0000.1001.0102
Legend:
s - arp              H - vtep             V - vpc-attached     p - peer-aged
R - peer-attached-rl B - bounce           S - static           M - span
```

```
D - bounce-to-proxy  O - peer-attached    a - local-aged       m - svc-mgr
L - local            E - shared-service
+-----------------------------------+--------------+----------------+-------------+----------
---+
    VLAN/                           Encap          MAC Address      MAC Info/       Interface
    Domain                          VLAN           IP Address       IP Info
+-----------------------------------+--------------+----------------+-------------+----------
---+
```

Given the BD is set to 'Flood' for L2 Unknown Unicast, here is what will happen at a high level:

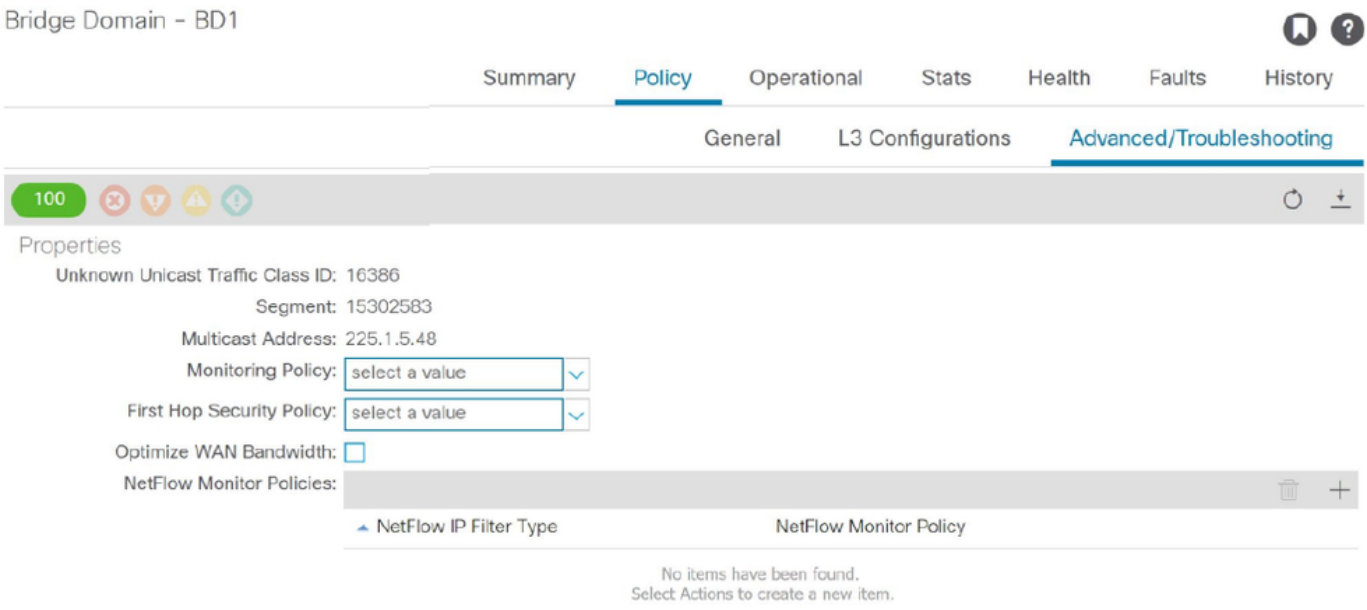
1. Ingress leaf will hash the packet header to assign it to one of the FTAGs (from 0 to 15).
2. Ingress leaf will encapsulate the frame in a VXLAN packet with the BD VNID. The outer destination IP will be the BD GIPo + FTAG.
3. It will be flooded in the fabric following a tree topology and should reach every leaf node that has the BD deployed.

This section will highlight what can be checked.

Finding BD GIPo

The GUI identifies multicast group 225.1.5.48 used by the BD for multi-destination traffic.

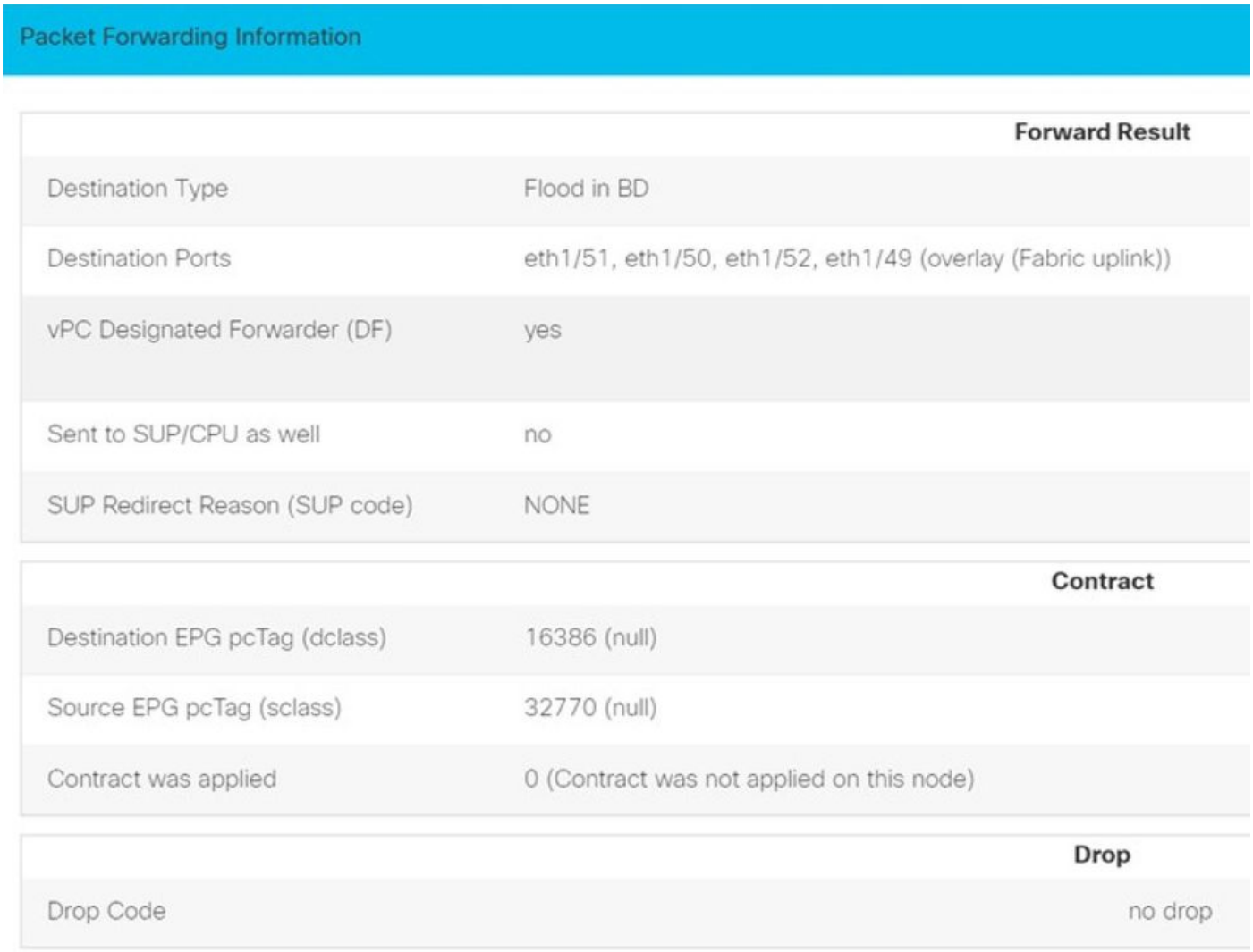
BD GIPo



ELAM — ingress leaf — flooded traffic

Using ELAM Assistant, the ELAM report on the ingress leaf is checked. This shows that the frame was flooded in the BD and is egressing on all fabric uplinks (here eth1/49, 1/50,1/51 and 1/52).

ELAM Assistant - ingress leaf - Packet Forwarding Information



To find the FTAG value selected by the ingress leaf, go to the raw report of the ELAM Assistant.

```
 sug_lu2ba_sb_info.mc_info.mc_info_nopad.ftag: 0xC
```

When converting the hexadecimal value of 0xC to decimal, this results in FTAG 12.

Drawing the FTAG topology

FTAG topology is computed by IS-IS. A tree topology is created for each FTAG value, with a root and output interface list which allows for an optimal load spread topology.

Display the local FTAG topology using the following command. In the example below, we're using FTAG ID 12 topology on spine1.

```
spine1# show isis internal mcast routes ftag
IS-IS process: isis_infra
 VRF : default
FTAG Routes
====================================

FTAG ID:  12  [Enabled] Cost:(   2/  11/   0)
--------------------------------
  Root port: Ethernet1/4.39
  OIF List:
```

```
    Ethernet1/11.11
    Ethernet1/12.12
```

Drawing the full FTAG topology in a large ACI fabric can prove to be a long and complex task. The 'aci-ftag-viewer' Python script (https://github.com/agccie/aci-ftag-viewer) can be copied onto an APIC. It generates the complete FTAG topology of the fabric in a single pass.

The output below displays the FTAG 12 tree in Pod1 of a Multi-Pod fabric and includes the FTAG topology across the IPN devices.

This shows that if traffic enters the ACI fabric from leaf101 it will traverse the following paths as listed in the script's output below.

```
admin@apic1:tmp> python aci_ftag_viewer.py --ftag 12 --pod 1
################################################################################
#  Pod 1 FTAG 12
#  Root spine-204
#  active nodes: 8, inactive nodes: 1
################################################################################
spine-204
 +- 1/1 -------- 1/52 leaf-101
 +- 1/2 -------- 1/52 leaf-102
 +- 1/3 -------- 1/52 leaf-103
 +- 1/4 -------- 1/52 leaf-104
                      +- 1/49 -------- 1/4 spine-201
                      |                    +- 1/11 ...... (EXT) Eth2/13 n7706-01-Multipod-A1
                      |                    +- 1/12 ...... (EXT) Eth2/9 n7706-01-Multipod-A2
                      |
                      +- 1/50 -------- 1/4 spine-202
                      |                    +- 1/11 ...... (EXT) Eth2/14 n7706-01-Multipod-A1
                      |                    +- 1/12 ...... (EXT) Eth2/10 n7706-01-Multipod-A2
                      |
                      +- 1/51 -------- 2/4 spine-203
                                           +- 2/11 ...... (EXT) Eth2/15 n7706-01-Multipod-A1
                                           +- 2/12 ...... (EXT) Eth2/11 n7706-01-Multipod-A2
 +- 1/11 ...... (EXT) Eth2/16 n7706-01-Multipod-A1
 +- 1/12 ...... (EXT) Eth2/12 n7706-01-Multipod-A2
```

ELAM — egress leaf — flooded traffic

In this case, the flooded traffic reaches every leaf in the ACI fabric. So, it will reach both leaf3 and leaf4 which are the VPC pair. Both of those leaf nodes have a VPC to the destination. To avoid duplicate packets, the VPC pair elects only one leaf to forward the flooded traffic to the destination. The elected leaf is called VPC DF leaf (VPC designated forwarder leaf).

This can be checked in ELAM using the following trigger on both leaf nodes.

```
module-1# debug platform internal tah elam asic 0
module-1(DBG-elam)# trigger reset
module-1(DBG-elam)# trigger init in-select 14 out-select 1
module-1(DBG-elam-insel14)# set inner ipv4 src_ip 10.1.1.1 dst_ip 10.1.1.2
module-1(DBG-elam-insel14)# start
```

leaf3 output:

```
module-1(DBG-elam-insel14)# ereport | egrep vpc.*df
 sug_lub_latch_results_vec.lub4_1.vpc_df: 0x1
```

leaf4 output:

```
module-1(DBG-elam-insel14)# ereport | egrep vpc.*df
 sug_lub_latch_results_vec.lub4_1.vpc_df: 0x0
```

In the above output, leaf3 has value '0x1' set for the 'vpc_df' field, whereas leaf4 has '0x0' set for the 'vpc_df' field. Hence the designated forwarder will be leaf3. leaf3 will forward the flooded packet on its VPC link to the destination EP.

Troubleshooting workflow for unknown Layer 2 unicast traffic — BD in hardware proxy

The current scenario listed is the one for Layer 2 unknown unicast traffic with the BD in hardware proxy mode. In this scenario, given the ingress leaf does not know the destination MAC address, it will forward the packet to the spine anycast proxy-mac address. The spine will perform a COOP lookup for the destination MAC.

If the lookup succeeds as shown below, the spine will rewrite the outer destination IP to the tunnel destination (here 10.0.96.66) and will send it to the leaf3-leaf4 VPC pair.

```
spine1# show coop internal info repo ep key 15302583 00:00:10:01:01:02

Repo Hdr Checksum : 16897
Repo Hdr record timestamp : 10 01 2019 11:05:46 351360334
Repo Hdr last pub timestamp : 10 01 2019 11:05:46 352019546
Repo Hdr last dampen timestamp : 01 01 1970 00:00:00 0
Repo Hdr dampen penalty : 0
Repo Hdr flags : IN_OBJ EXPORT ACTIVE
EP bd vnid : 16351141
 EP mac :  00:00:10:01:01:02
flags : 0x90
repo flags : 0x122
Vrf vnid : 2097154
Epg vnid : 0
EVPN Seq no : 0
Remote publish timestamp: 01 01 1970 00:00:00 0
Snapshot timestamp: 10 01 2019 11:05:46 351360334
Tunnel nh : 10.0.96.66
MAC Tunnel  : 10.0.96.66
IPv4 Tunnel : 10.0.96.66
IPv6 Tunnel : 10.0.96.66
ETEP Tunnel : 0.0.0.0
```

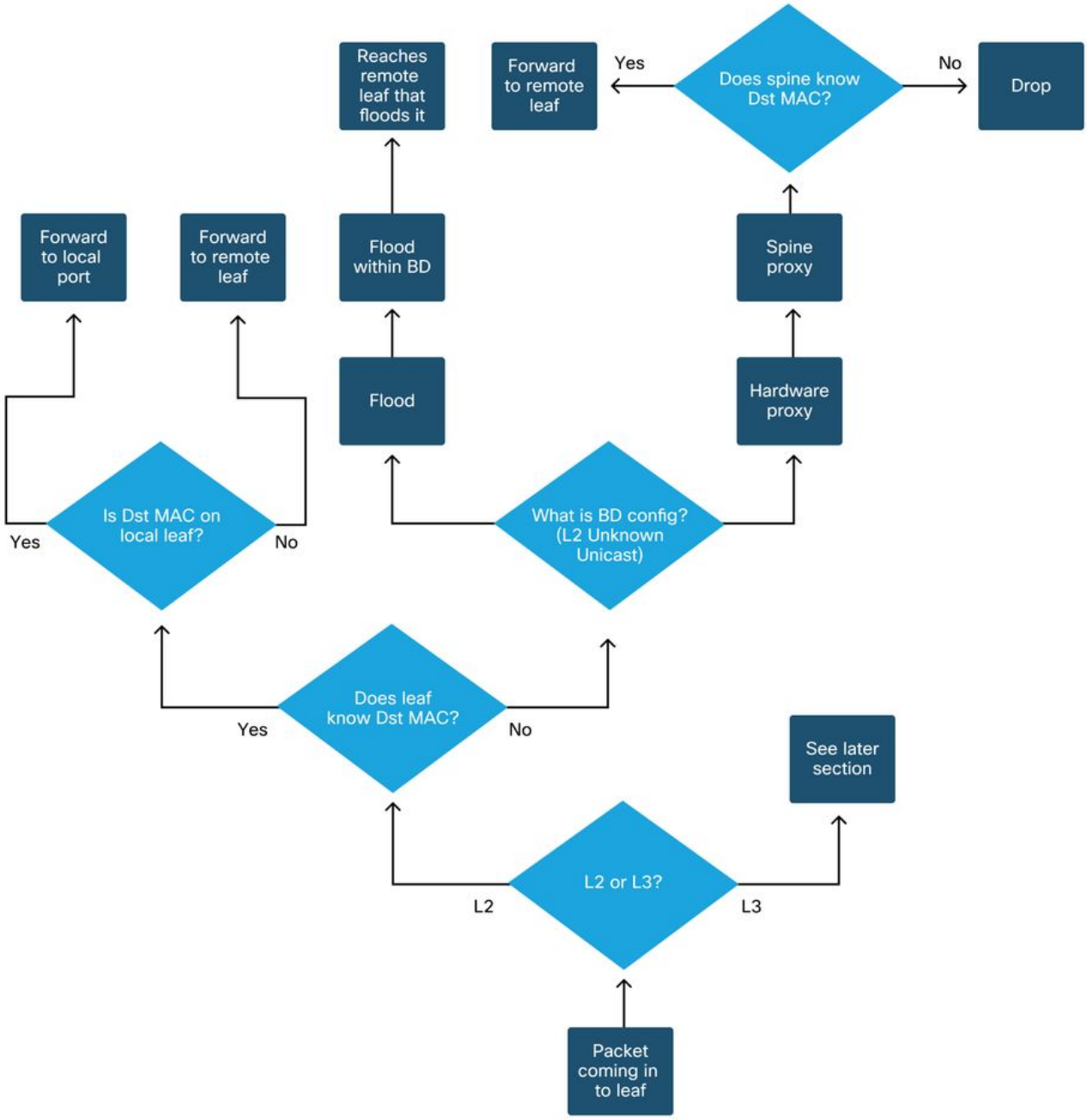
If the lookup fails (endpoint is unknown in the ACI fabric), the spine will drop the unknown unicast.

```
spine1# show coop internal info repo ep key 15302583 00:00:10:01:01:02
Key not found in repo
```

Layer 2 Forwarding Summary

The following diagram summarizes the possible forwarding behavior for Layer 2 traffic in the ACI fabric.

ACI fabric Layer 2 forwarding behavior


Packet coming in to leaf
L2 or L3?
L2
L3
See later section
Does leaf know Dst MAC?
Yes
No
Is Dst MAC on local leaf?
Yes
No
Forward to local port
Forward to remote leaf
What is BD config? (L2 Unknown Unicast)
Flood
Flood within BD
Reaches remote leaf that floods it
Hardware proxy
Spine proxy
Does spine know Dst MAC?
Yes
No
Forward to remote leaf
Drop