The bridge to possible

A Network Engineer's Blueprint for ACI Forwarding

Joe Young, ACI Technical Leader, Customer Experience

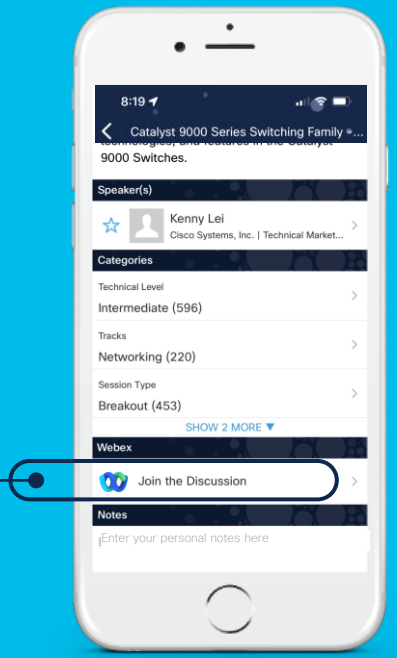
Cisco Webex App

Questions?

Use Cisco Webex App to chat
with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space



Webex spaces will be moderated until February 24, 2023.

Agenda

- What's Different About ACI Forwarding?
  - (iVXLAN, contracts, endpoint learning)
- Proxy Forwarding
- ACI Forwarding Tables
  - Endpoint tables, routing tables, hardware lookups
- Understanding the Configuration Options
- The Anatomy of an ACI Switch

Agenda

- Understanding the Tools
  - UI Tools
  - Elam
  - Ftriage
  - Span / ERSPAN
  - Flow Telemetry / netflow
- Debugging and Walking Through ACI Flows
  - (Routed, Bridged, BUM, Proxied)

Glossary of Acronymns

Acronyms	Definitions
ACI	Application Centric Infrastructure
APIC	Application Policy Infrastructure Controller
EP	Endpoint
EPG	Endpoint Group
BD	Bridge Domain
VRF	Virtual Routing and Forwarding
COOP	Council of Oracle Protocol
VxLAN	Virtual eXtensible LAN

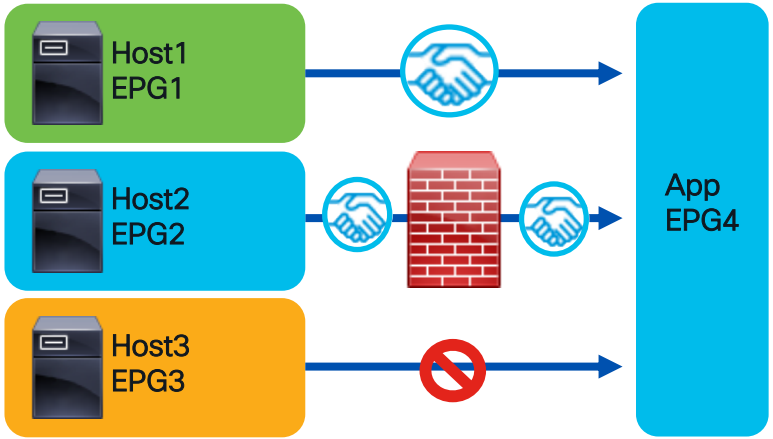
VxLAN packet acronyms

Acronyms	Definitions
d*XXX*o	Outer Destination XXX (dIPo = Outer Destination IP)
s*XXX*o	Outer Source XXX (sIPo = Outer Source IP)
d*XXX*i	Inner Destination XXX (dIPi = Inner Destination IP)
s*XXX*i	Inner Source XXX (sIPi = Inner Source IP)
GIPo	Outer Multicast Group IP
VNID	Virtual Network Identifier

What’s Different About ACI Forwarding?

What is “Application Centric”?

- Traditional networks use ACL’s to classify traffic
  - Usually based on L3 or L2 addresses
  - Makes security decisions (permit, deny, log, etc)
  - Makes forwarding decisions (policy based routing)
- ACI can classify traffic based on its EPG
- Traffic inherits the forwarding and security policy of the EPG



How is “Application Centric” Achieved?

Sources and Destinations Must be Classified into EPG’s

Endpoints

- Used by App EPG’s
- Represents the network identity of an end device
- Learned dynamically or configured statically

Policy-Prefixes

- Used by External EPG’s
- Classifies destination by longest prefix match
- Also used for shared-services
- Configured

PcTags

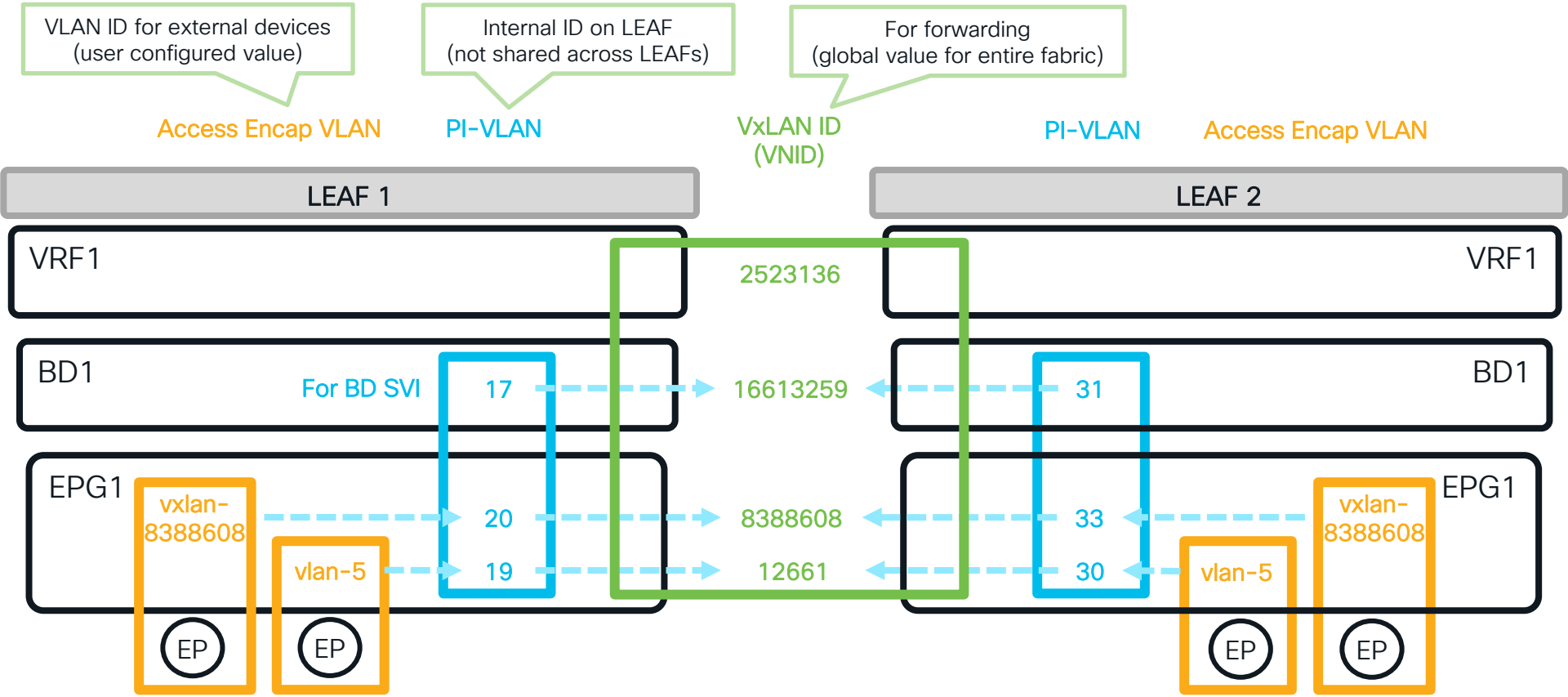
- The security ID of an EPG
- Used in contracts. Ex: Permit PcTag 1000 to PcTag 2000
- Sclass/dclass imply PcTag direction

Contracts

- Defines security and sometimes forwarding (pbr) policy between epgs
- Essentially an ACL between PcTags
- Consumer/Provider rather than src/dest

Vlan Types

※ PI-VLAN : Platform Independent VLAN



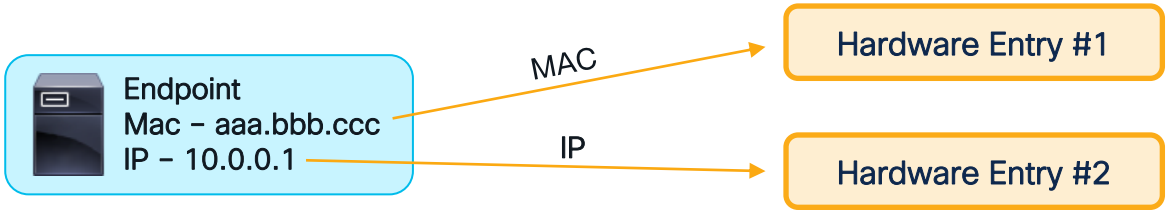
What is an Endpoint?

At the APIC level an Endpoint is a Mac address with zero or more IP/IPv6 Addresses

fvCEp
<epg-dn>/cep-00:00:00:00:0a

fvIp
<epg-dn>/cep-00:00:00:00:0a/ip-[10.0.0.10]

At the Switch level an Endpoint is a Mac address **OR** an IP/IPv6 Address



What is an Endpoint?

An Endpoint joins both forwarding and security policy

Local Learn

```
leaf103# show system internal epm end ip 192.168.200.11

MAC : 0000.1111.2222 ::: Num IPs : 1
IP# 0 : 192.168.200.11 ::: IP# 0 flags :  ::: l3-sw-hit: No
Vlan id : 2 ::: Vlan vnid : 12661 ::: VRF name : CL2022:vrf1
BD vnid : 16613259 ::: VRF vnid : 2523136
Phy If : 0x40018000 ::: Tunnel If : 0
Interface : Ethernet1/25/1
Flags : 0x80005c04 ::: sclass : 32771 ::: Ref count : 5
EP Create Timestamp : 11/01/2021 14:06:25.769904
EP Update Timestamp : 11/04/2021 18:51:54.387104
EP Flags : local|IP|MAC|host-tracked|sclass|timer|
```

Remote Learn

```
leaf103# show system internal epm endpoint ip 192.168.100.10

MAC : 0000.0000.0000 ::: Num IPs : 1
IP# 0 : 192.168.100.10 ::: IP# 0 flags :  ::: l3-sw-hit: No
Vlan id : 0 ::: Vlan vnid : 0 ::: VRF name : CL2022:vrf1
BD vnid : 0 ::: VRF vnid : 2523136
Phy If : 0 ::: Tunnel If : 0x18010001
Interface : Tunnel1
Flags : 0x80004400 ::: sclass : 49154 ::: Ref count : 3
EP Create Timestamp : 11/04/2021 16:38:13.570615
EP Update Timestamp : 11/04/2021 18:51:54.386595
EP Flags : IP|sclass|timer|
```

VNID

Interface/TEP

PcTag

What is a TEP? (Tunnel Endpoint)

- IP addresses allocated for overlay communication
- VXLAN Traffic is sent to the TEP + VNID of destination

Most Common TEP Types

TEP Type	What is it?	What is it for?
Physical TEP (PTEP)	Unique Overlay IP Address for each individual Leaf/Spine	Non-vpc dataplane, l3out communication, apic-leaf comm, etc
VPC TEP (VTEP)	Unique Overlay IP Address for each VPC Pair	Traffic destined to endpoints that are connected behind VPC
Proxy TEP	Spine Anycast IP's used for proxy traffic	Leafs send to these TEPs when doing proxy forwarding

```
a-leaf101# show ip interface loopback0
IP Interface Status for VRF "overlay-1"
lo0, Interface status: protocol-up/link-up/admin-up, iod: 4, mode: ptep
```

What are Tunnels?

- Leafs/Spines Install Tunnel Interface to each known TEP.
- Used for VXLAN Dataplane

How are Tunnels Learned?

Dataplane Learns

```
leaf# moquery -c tunnelIf -f 'tunnel.If.id=="tunnel1"'

id                  : tunnel1
dest                : 10.0.72.67
idRequestorDn       : sys/*/db-dtep/dtep-[10.0.72.67]
```

Through BGP (l3out routes)

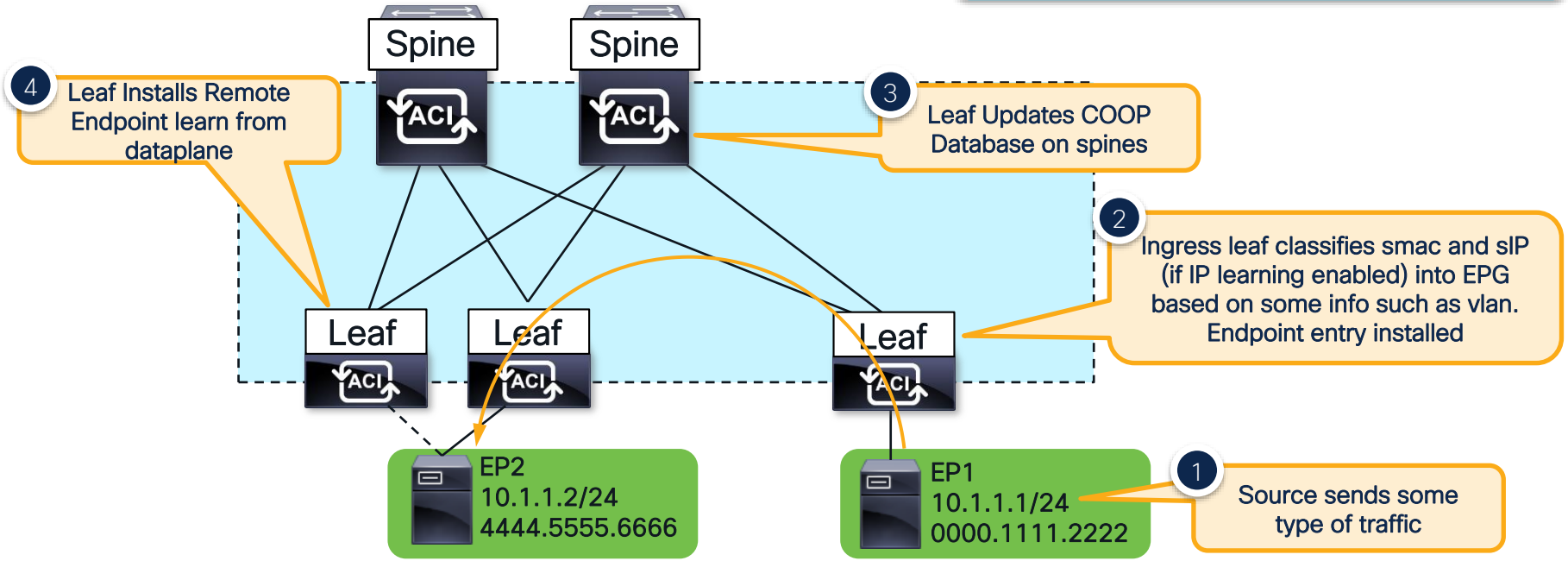
```
leaf# moquery -c tunnelIf -f 'tunnel.If.id=="tunnel1"'

id                  : tunnel1
dest                : 10.0.72.64
idRequestorDn       : sys/bgp/*/db-dtep/dtep-[10.0.72.64]
```

Local POD ISIS Database

```
leaf# moquery -c tunnelIf -f 'tunnel.If.id=="tunnel1"'

# tunnel.If
id                  : tunnel1
dest                : 10.0.152.64
idRequestorDn       : sys/isis/*/lvl-l1/db-dtep/dtep-[10.0.152.64]
```

How is an Endpoint Learned?

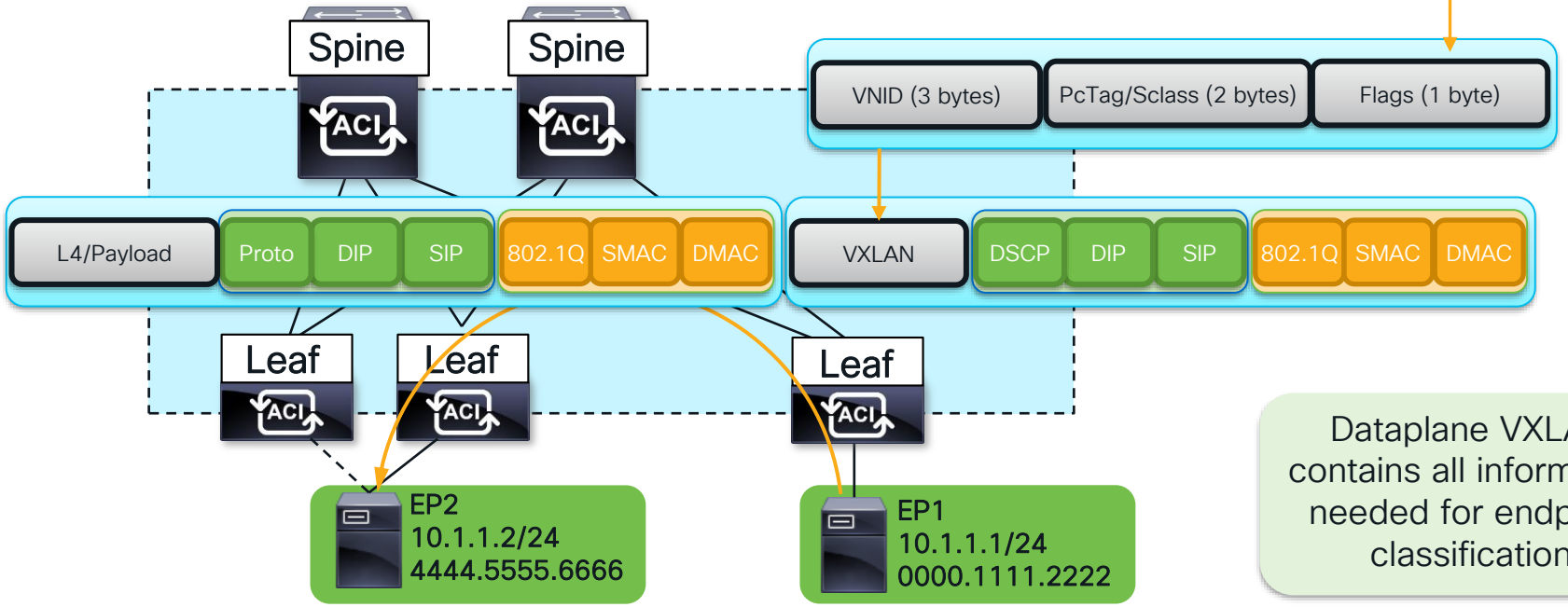
How does the Egress leaf classify traffic into the correct EPG?



Overlay iVXLAN

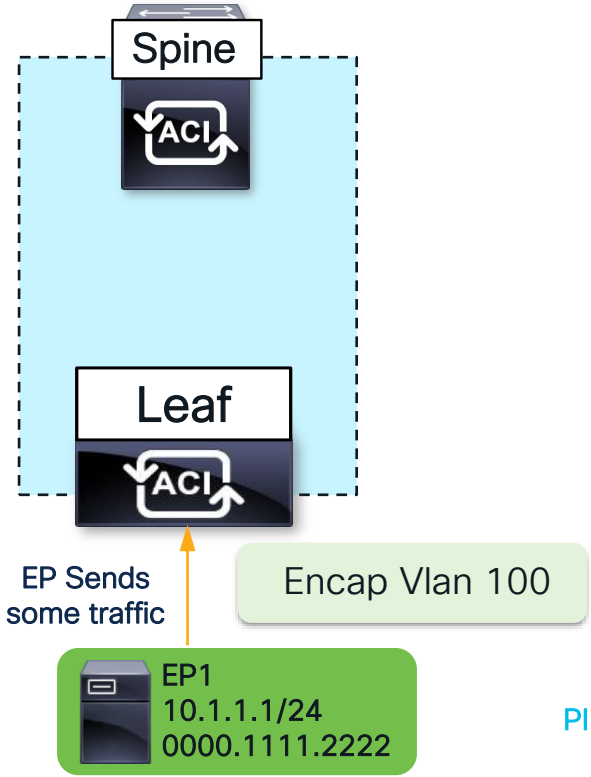
ACI uses VXLAN with some additional bits

Bit pos 4 – Source Policy Applied
Bit pos 5 – Destination Policy Applied
Bit pos 7 – Don't learn



Dataplane VXLAN contains all information needed for endpoint classification

How is an Endpoint Learned?




```
leaf103# show system internal epm vlan 2 detail

VLAN 2
VLAN type : FD vlan
hw id : 34 ::: sclass : 32771
access enc : (802.1Q, 100)
fabric enc : (VXLAN, 12661)
Object store EP db version : 4
BD vlan id : 1 ::: BD vnid : 16613259 ::: VRF vnid : 2523136
Valid : Yes ::: Incomplete : No  ::: Learn Enable : Yes
```

PI-VLAN

```
leaf103# show vlan encap-id 100

VLAN Name                          Status    Ports
---- ----------------------------- --------- -------------------------------
2    CL2022:ap1:epg2               active    Eth1/25/3
```

Checking Endpoints

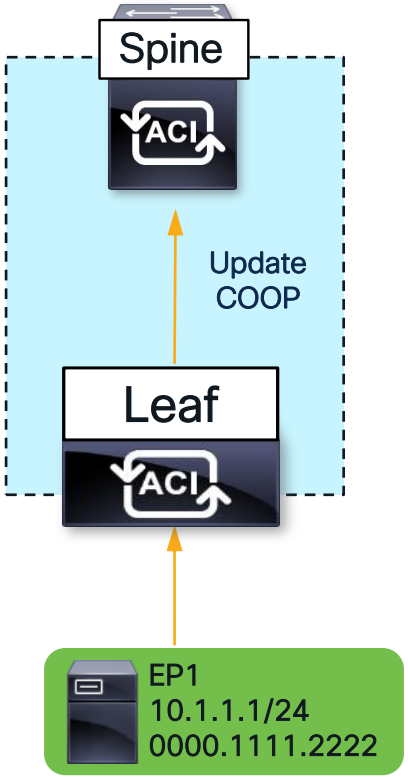
Reference commands can be run from leafs or apics

```
#Check object model for Mac Address Endpoint
moquery -c epmMacEp -f 'epm.MacEp.addr=="00:00:AA:AA:BB:BB"'

#Check object model for IP Address Endpoint
moquery -c epmIpEp -f 'epm.IpEp.addr=="192.168.200.11"'
```

Reference commands can be run from leafs only

```
#Check endpoint manager process directly
show system internal epm endpoint mac 0000.aaaa.bbbb
show system internal epm endpoint ip 192.168.200.11

#Check hardware level endpoint process directly
vsh_lc -c "show system internal epmc endpoint mac 0000.aaaa.bbbb"
vsh_lc -c "show system internal epmc endpoint ip 192.168.200.11"
```

How is an Endpoint Learned?



The Leaf Updates COOP on Spines

```
spine1005# show coop internal info ip-db | grep -B 1 -A 15
192.168.200.11
--------------------------------
IP address : 192.168.200.11
Vrf : 2523136
Flags : 0
EP bd vnid : 16613259
EP mac :  00:00:AA:AA:BB:BB
Publisher Id : 10.0.64.70
Record timestamp : 11 05 2021 17:02:56 217794556
Publish timestamp : 11 05 2021 17:02:56 220584642
Seq No: 0
Remote publish timestamp: 01 01 1970 00:00:00 0
URIB Tunnel Info
Num tunnels : 1
        Tunnel address : 10.0.64.70
        Tunnel ref count : 1
```

VNID info should match the info on leaf

Leaf TEP that owns this EP:
#From APIC
moquery -c ipv4Addr -f 'ipv4.Addr.addr=="10.0.64.70"'

Checking COOP

Reference commands can be run from spines or apics

Query COOP for l2 entry:

```
moquery -c coopEpRec -f 'coop.EpRec.mac=="00:00:AA:AA:BB:BB"'
```

Query COOP for l3 entry and get parent l2 entry:

```
moquery -c coopEpRec -x rsp-subtree=children 'rsp-subtree-filter=eq(coopIpv4Rec.addr,"1.1.1.1")' rsp-subtree-include=required
```

Query COOP for l3 only entry (such as an SVI IP):

```
moquery -c coopIpOnlyRec -f 'coop.IpOnlyRec.addr=="192.168.100.10"'
```

Query COOP for l3 ep:

```
moquery -c coopIpv4Rec -f 'coop.Ipv4Rec.addr=="192.168.100.10"'
```

How is Traffic Classified with no EP Learn?

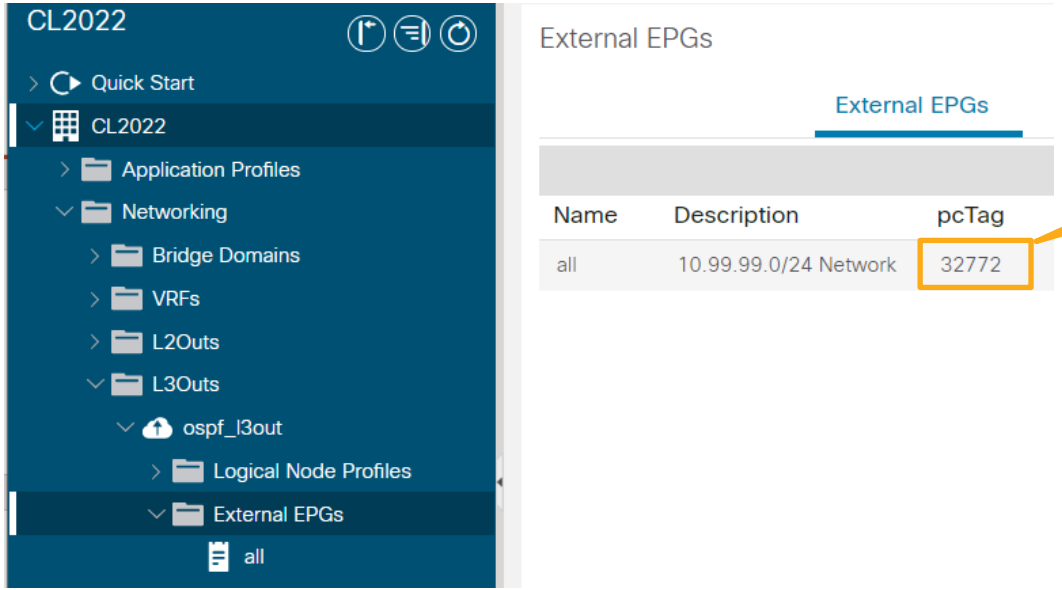
In most of these cases, the pcTag is based on a policy-prefix lookup

- There will be no endpoint learn in several cases
- Source/dest is behind an l3out
- Source/dest is in another vrf
- Endpoint learning is disabled by some option
- If ingress leaf doesn't apply policy, egress leaf should (indicated via policy-applied bits in ivxlan header)

How is Traffic Classified with no EP Learn?

Destination Behind L3out

```
leaf101# vsh_lc -c "show forwarding route 10.99.99.100 platform vrf CL2022:vrf1"
!
Policy Prefix 10.99.99.0/24
!
vrf: 16(0x10), routed_if: 0x0 epc_class: 32772(0x8004)
```



Classification based on longest l3out policy prefix

How is Traffic Classified with no EP Learn?

Destination is unknown and is proxied

```
leaf101# show ip route 192.168.200.20 vrf CL2022:vrf1

192.168.200.0/24, ubest/mbest: 1/0, attached, direct, pervasive
    *via 10.0.176.66%overlay-1, [1/0], 4d05h, static, tag 4294967294
        recursive next hop: 10.0.176.66/32%overlay-1

leaf101# vsh_lc -c "show forwarding route 192.168.200.20 platform vrf CL2022:vrf1"
!
Policy Prefix 0.0.0.0/0
!
Vrf: 16(0x10), routed_if: 0x0 epc_class: 1(0x1)
```

"Pervasive" indicates this is a BD or EPG subnet (fvSubnet). Send to spine proxy-addr

-pcTag of 1 indicates the fabric owns the subnet, don't apply policy
-policy applied flags not set in ivxlan header

Don't apply policy, Forward to proxy Anycast!

```
leaf101# show isis dtep vrf overlay-1 | egrep "Type|PROXY"
DTEP-Address      Role    Encapsulation   Type
10.0.176.66       SPINE   N/A             PHYSICAL,PROXY-ACAST-V4
10.0.176.65       SPINE   N/A             PHYSICAL,PROXY-ACAST-MAC
10.0.176.64       SPINE   N/A             PHYSICAL,PROXY-ACAST-V6
```

How is Traffic Classified with no EP Learn?

Shared Services Classification

Destination is in shared services **provider** EPG (different vrf)

```
leaf# show ip route 192.168.255.10 vrf CL2022:vrf1
192.168.255.0/24, ubest/mbest: 1/0, attached, direct, pervasive
    *via 10.0.176.66%overlay-1, [1/0], static, tag !!!, rwVnid: vxlan-2457601
        recursive next hop: 10.0.176.66/32%overlay-1

leaf# vsh_lc -c "show forwarding route 192.168.255.10 plat vrf CL2022:vrf1"
Prefix:192.168.255.0/24, Update_time:Fri Nov  5 20:57:00 2021
!
Policy Prefix 0.0.0.0/0
!
Flags: IN-HW, SHRD-SVC,
vrf: 16(0x10), routed_if: 0x0 epc_class: 36(0x24)
```

PcTag of provider epg

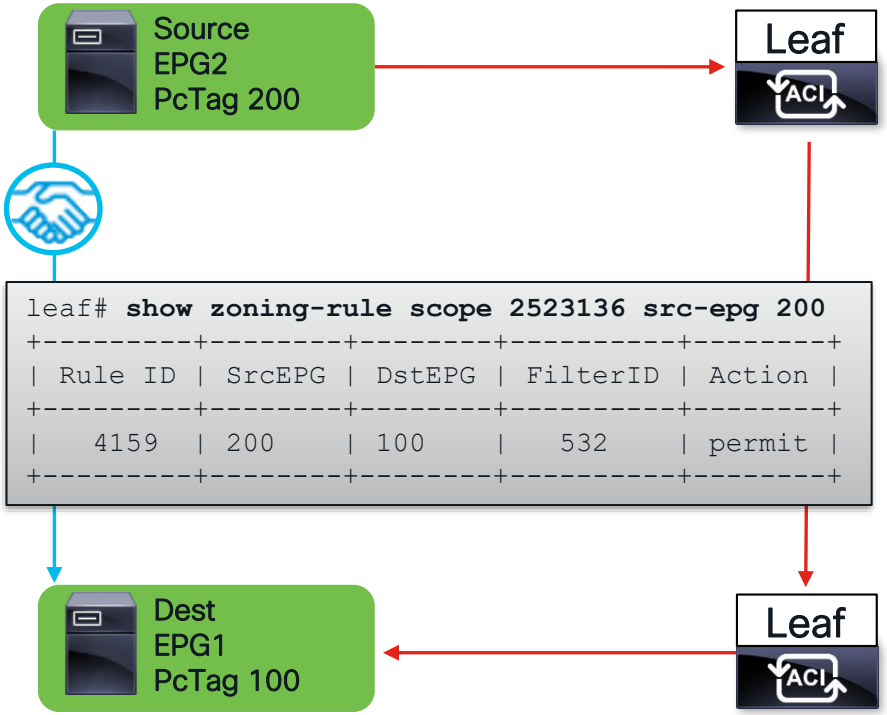
Destination is in shared services consumer EPG (different vrf)

```
leaf# show ip route 192.168.100.10 vrf CL2022:vrf2
192.168.100.0/24, ubest/mbest: 1/0, attached, direct, pervasive
    *via 10.0.176.66%overlay-1, [1/0], static, rwVnid: vxlan-2523136
        recursive next hop: 10.0.176.66/32%overlay-1

leaf# vsh_lc -c "show forwarding route 192.168.100.10 plat vrf CL2022:vrf2"
Prefix:192.168.100.0/24, Update_time:Tue Nov  9 14:34:05 2021
!
Policy Prefix 0.0.0.0/0
!
Flags: IN-HW, SHRD-SVC,
vrf: 10(0xa), routed_if: 0x0 epc_class: 14(0xe)
```

Reserved tag for shared services consumer. Policy applied in consumer vrf

Check hidden slide for impact of “Policy Control Enforcement Direction” setting

Contracts and Forwarding




```
leaf# show zoning-rule scope 2523136 src-epg 200
+---------+--------+--------+----------+--------+
| Rule ID | SrcEPG | DstEPG | FilterID | Action |
+---------+--------+--------+----------+--------+
|   4159  | 200    | 100    |   532    | permit |
+---------+--------+--------+----------+--------+
```

Ingress

Contract Found?

- Yes → Set policy-applied bits in ivxlan. Permit, deny, redir, log
- No → If LPM is BD/EPG subnet, forward and don’t set policy-applied bits in ivxlan. Otherwise, drop!

Egress

Policy-Applied Bits set?

- Yes → Don’t do contract lookup. Forward.
- No → Do contract lookup. Permit, deny, redir, log

Policy enforcement table

Where is policy enforced?

VRF Enforcement Setting

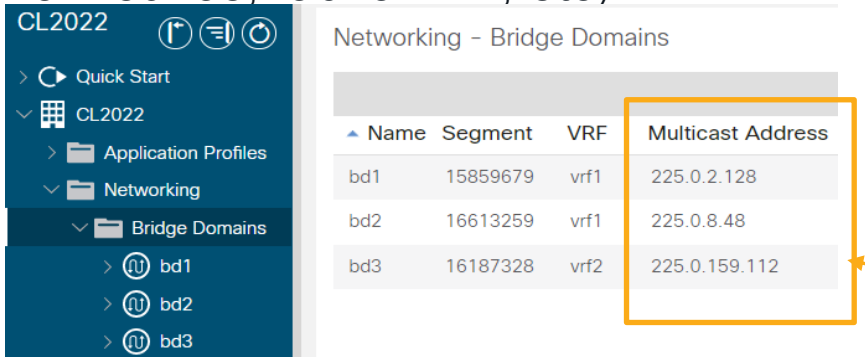
Flow Direction	INGRESS	EGRESS
EPG to unknown EPG	Applied Egress	Unchanged
EPG to known EPG	Applied Ingress	Unchanged
EPG to L3out	Applied Ingress/non-BL	Applied Egress/BL
L3out to unknown EPG	Applied Egress/non-BL	Applied Egress
L3out to known EPG	Applied Egress/non-BL	Applied Ingress/BL
L3out to L3out	Applied Ingress	Applied Egress

Policy enforcement affects only traffic to or from the L3Out.
There are no behavior changes in EPG-to-EPG.

What About Flooded Traffic?

The following traffic may be flooded:

- Broadcast
- Multicast
- Unknown Unicast
- Control Plane maintenance (EP announce, fabric ARP, etc)



Name	Segment	VRF	Multicast Address
bd1	15859679	vrf1	225.0.2.128
bd2	16613259	vrf1	225.0.8.48
bd3	16187328	vrf2	225.0.159.112

How does ACI flood?

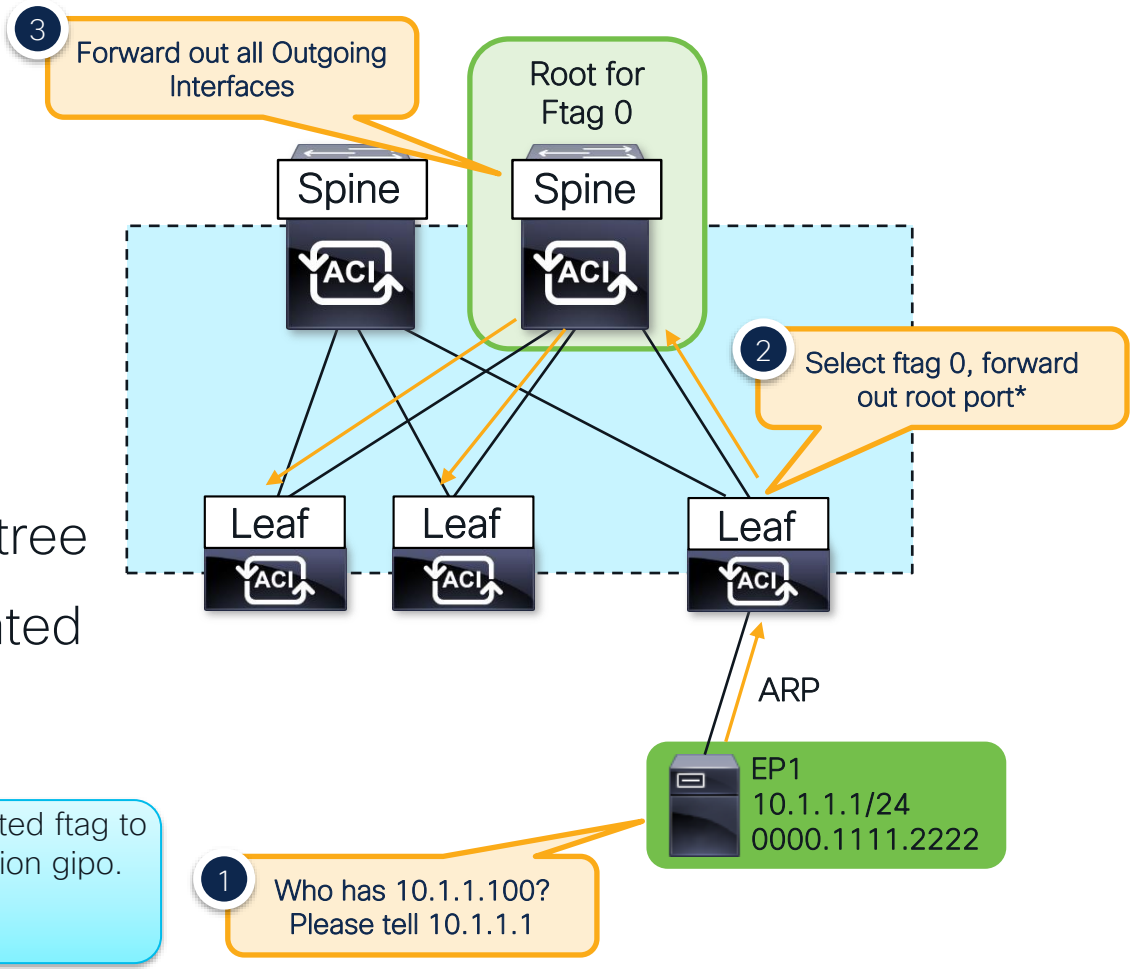
- Flooded traffic is sent to the BD GiPo (l2 flood) or VRF GiPo (l3 flood)
- The GiPo is an overlay multicast address allocated to a BD or VRF
- Flooding is done on a loop-free tree called an FTAG

Security policy NOT applied

GiPo

What are FTAGs?

- FTAGs are loop-free trees within the overlay used by flooded traffic
- FTAGs are picked per flow from values 0 – 0xc
- One spine is root for each tree
- Outgoing interfaces calculated by ISIS



*Note, the ingress leaf communicates the selected ftag to the rest of the fabric by adding it to the destination gipo. If the gipo is 225.0.0.0 and the ftag is 0x9, the destination address would be 225.0.0.9

Checking FTAGs

Find the outgoing interfaces for a tree

Check FTAG tree on ingress leaf

```
leaf101# show isis internal mcast routes ftag

FTAG Routes
====================================

 FTAG ID:   0  [Enabled] Cost:(   1/   7/   0)
 ---------------------------------
    Root port: Ethernet1/54.6
    OIF List:
      Ethernet1/53.5
!
!ommitted rest of ftags
```

Leaf forwards to root port and any additional OIFs

Check FTAG tree on root spine

```
spine1005# show isis internal mcast routes ftag

FTAG Routes
================================================

 FTAG ID:   0 [Root] [Enabled] Cost:(   0/   0/   0)
 ---------------------------------
    Root port: -
    OIF List:
      Ethernet1/1.20
      Ethernet1/2.21
      Ethernet1/3.19
!ommitted rest of ftags
```

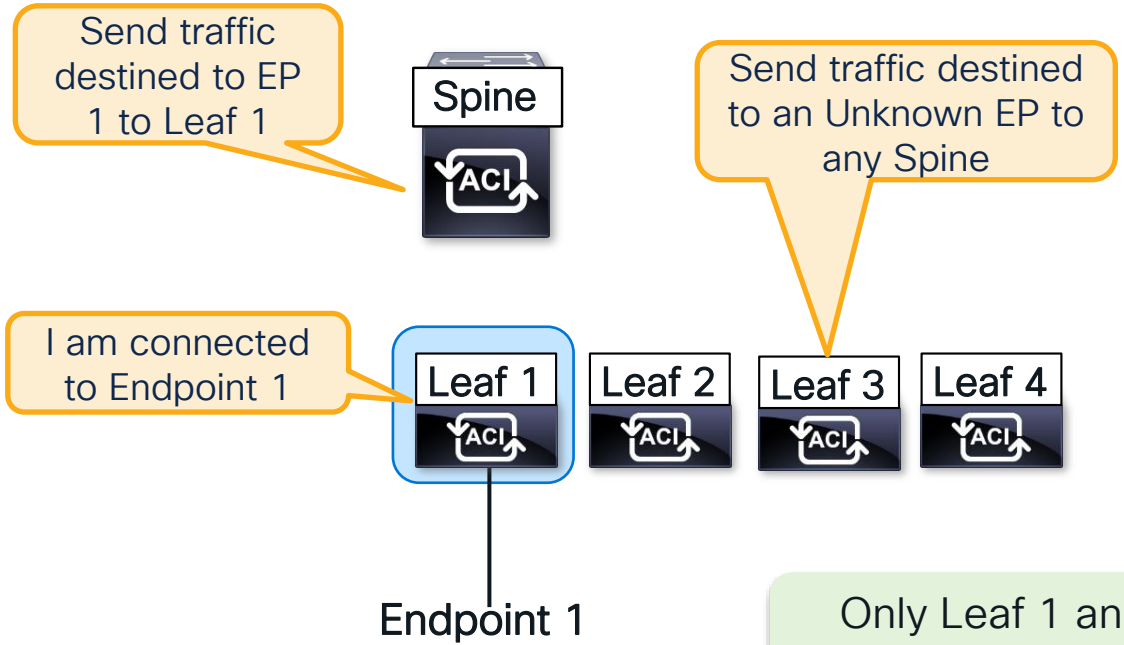
This spine is the root for ftag 0

Forward out all of these interfaces

Proxy Forwarding

What is Proxy Forwarding?

Why? Scaling out Endpoint Learning



Spines own separate anycast TEP's for mac, ipv4, and ipv6 proxy lookups

Only Leaf 1 and Spines have to program Endpoint 1 in hardware

How to check the Spine-Proxy TEP

```
leaf1# show ip route vrf CL2022:vrf1

192.168.0.0/24, ubest/mbest: 1/0, attached, direct, pervasive
    *via 10.0.16.64%overlay-1, [1/0], 00:21:39, static
```

BD Subnet (Pervasive Route)

next-hop should be SPINE-PROXY

```
leaf1# show isis dteps vrf overlay-1 | grep PROXY
10.0.16.65          SPINE    N/A              PHYSICAL,PROXY-ACAST-MAC
10.0.16.64          SPINE    N/A              PHYSICAL,PROXY-ACAST-V4
10.0.16.67          SPINE    N/A              PHYSICAL,PROXY-ACAST-V6
```

next-hop of Pervasive Route is IPv4 Spine Proxy TEP

Three types of Spine Proxy TEP

- Proxy-Acast-MAC
  - ✓ Spine-Proxy for L2 traffic (L2 Unknown Unicast mode “Hardware Proxy”)
- Proxy-Acast-V4
  - ✓ Spine-Proxy for IPv4 traffic (includes ARP Request with ARP Flooding mode “OFF”)
- Proxy-Acast-V6
  - ✓ Spine-Proxy for IPv6 traffic

What is COOP?

COOP is the proxy-database of ACI

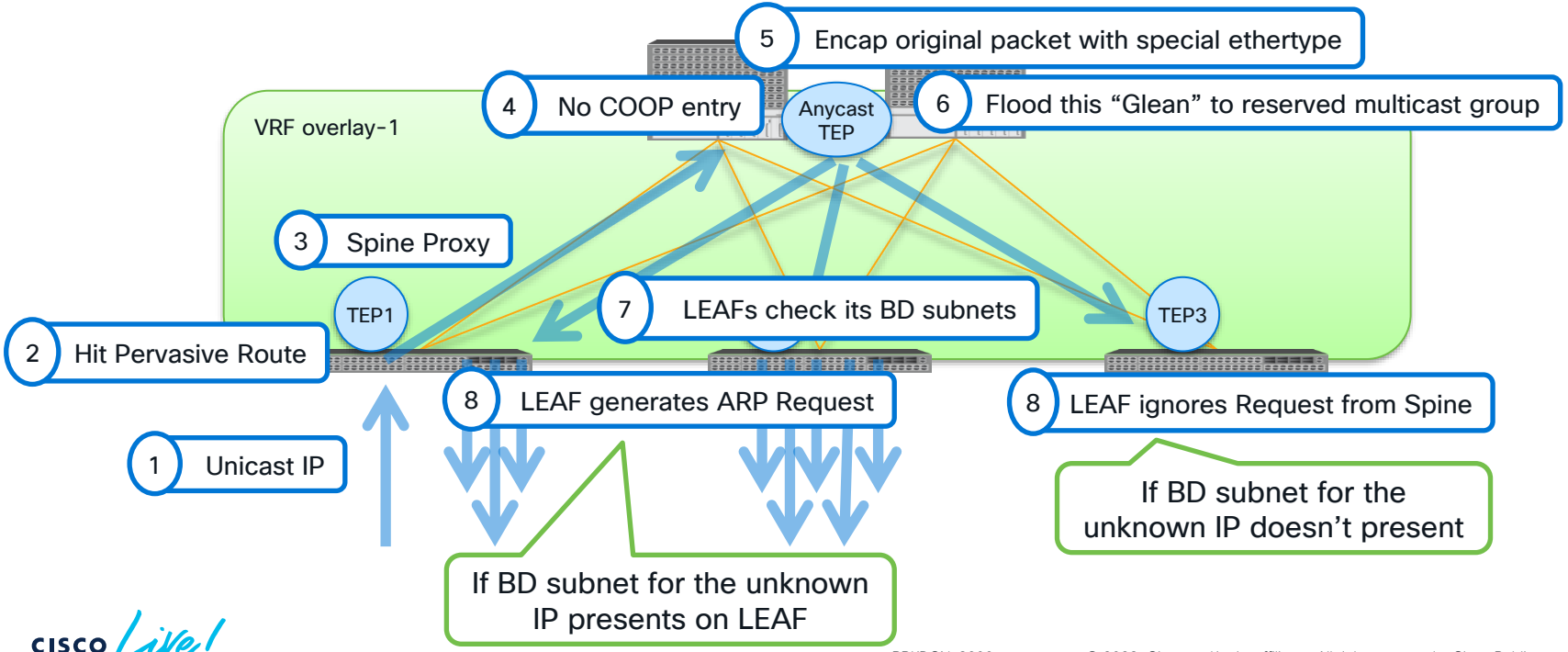
- Council of Oracles Protocol – A TCP protocol for citizens (Leafs) to publish records to oracles (Spines).
- Used for announcing endpoints, fabric owned IP's, multicast information, and more
- Synced across Pods/Sites with BGP EVPN
- Each Endpoint Record contains all information to forward (VNID, leaf TEP, mac, etc)
- COOP records pushed into hardware on spines
- For modular spines, scale is achieved by pushing each EP onto only two Fabric Modules

What if the Endpoint isn't in COOP? (ARP Glean)

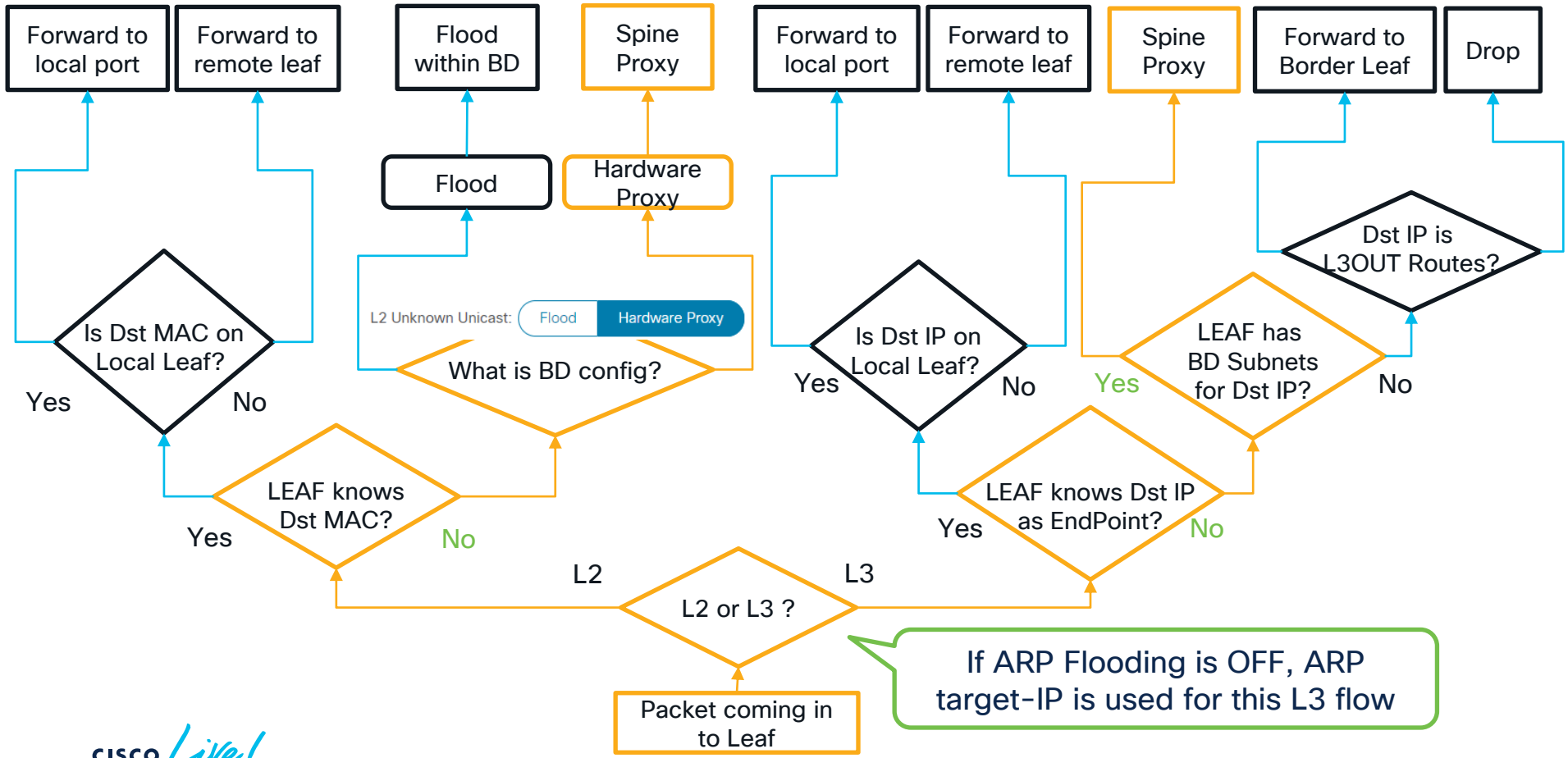
What if Spine's COOP DB doesn't know the destination when proxy'ed?

✗ L2 Traffic : Drop

✓ L3 Traffic : ARP Glean



Spine Proxy Summary

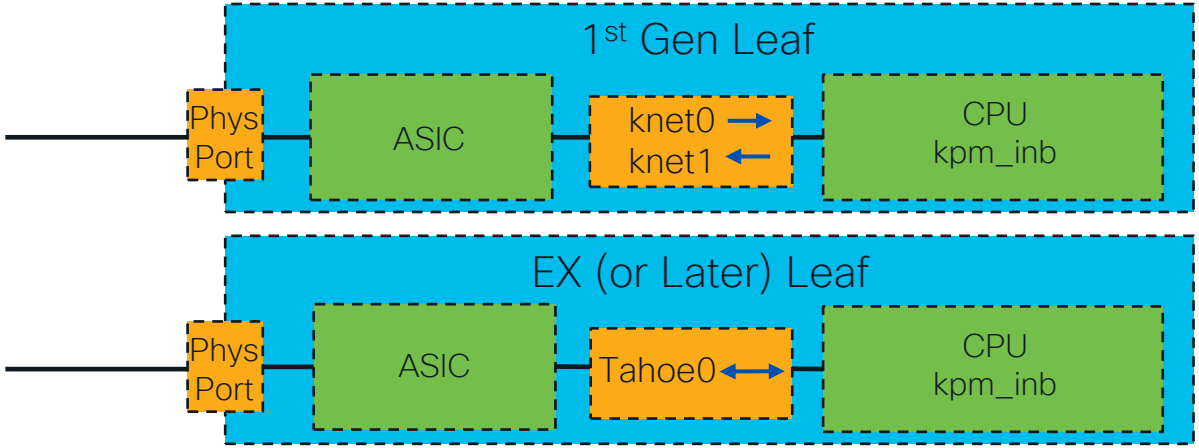


If ARP Flooding is OFF, ARP target-IP is used for this L3 flow

Capturing a Glean with Tcpdump

ACI Leafs and Spines contain pseudo interfaces for traffic to and from the CPU

- Traffic on the on the knet or tahoe pseudo interface will have a special ieth header. It must be decoded.
- Starting in 3.2 the *knet_parser.py* script is available on the switch cli to decode



- For traffic going to the cpu check knet0 and kpm_inb
- For traffic coming from the cpu check knet1 and kpm_inb
- For traffic to and from the cpu check Tahoe0 and kpm_inb

*Note, not all traffic will show up on the kpm_inb interface. However, all traffic shows on the pseudo interface
*Gen1 and 2 Modular spines use psdev0, psdev1, and psdev2 interfaces. Gen 2 fixed spines use tahoe0. Gen 1 fixed spines use knet0-3

Capturing a Glean with Tcpdump

Gen2 or Later Leaf

Egress Leaf Verification

```
tcpdump -xxxvei tahoe0 -w /bootflash/tahoe0.pcap
knet_parser.py --file /bootflash/tahoe0.pcap --pcap --decoder tahoe

Frame 111
  Time: 2019-05-16T16:56:33.059831+00:
  Header: ieth_extn CPU Receive
        sup_qnum:0x14, sup_code:0x21, istack:ISTACK_SUP_CODE_SPINE_GLEAN(0x21)
  Header: ieth
        sup_tx:0, ttl_bypass:0, opcode:0x6, bd:0x120e, outer_bd:0x27, dl:0, span:0, traceroute:0, tclass:0
        src_idx:0x3a, src_chip:0x0, src_port:0x19, src_is_tunnel:1, src_is_peer:1
        dst_idx:0x0, dst_chip:0x0, dst_port:0x0, dst_is_tunnel:0
  Len: 148
  Eth: 000d.0d0d.0d0d > 0100.5e7f.fff1, len/ethertype:0x8100(802.1q)
  802.1q: vlan:2, cos:5, len/ethertype:0x800(ipv4)
  ipv4: 10.0.116.64 > 239.255.255.241, len:130, ttl:249, id:0x0, df:0, mf:0, offset:0x0, dscp:32, prot:17(udp)
  udp: (ivxlan) 0 > 48879, len:110
  ivxlan: n:1, l:1, i:1,
        vnid: 0x2b0000
        lb:0, dl:1, exception:0, src_policy:0, dst_policy:0, src_class:0
        mcast(routed:0, ingress_encap:0/802.1q), ac_bank:0, src_port:0x0
  Eth: 000c.0c0c.0c0c > ffff.ffff.ffff, len/ethertype:0xfff2(aci-glean)
  ipv4: 172.16.1.1 > 172.16.2.2, len:84, ttl:63, id:0x71f9, df:1, mf:0, offset:0x0, dscp:0, prot:1(icmp)
  icmp: echo request id:0x9092, seq:0x1980
```

Decode type should be tahoe for tahoe interface

RX sup traffic rather than TX

Switch recognizes this as a Glean

Traffic that triggered Glean

Capturing a Glean with Tcpdump

Egress Leaf Verification

Gen1 Leaf Example

knet0 would show Rx traffic (similar output as Tahoe0)

```
tcpdump -xxxvei knet0 -w /bootflash/knet0.pcap
knet_parser.py --file /bootflash/knet0.pcap --pcap --decoder knet
```

knet1 would show Tx traffic

```
tcpdump -xxxvei knet1 -w /bootflash/knet1.pcap
knet_parser.py --file /bootflash/knet1.pcap --pcap --decoder knet
```

No decode necessary for kpm_inb (cpu) interface…Gleans aren't easily readable

```
tcpdump -xxxvei kpm_inb ether proto 0xfff2
a-leaf102# tcpdump -xxxvei kpm_inb ether proto 0xfff2
tcpdump: listening on kpm_inb, link-type EN10MB (Ethernet), capture size 65535 bytes
15:27:37.663580 00:0c:0c:0c:0c:0c (oui Unknown) > Broadcast, ethertype Unknown (0xfff2), length 94:
        0x0000:  ffff ffff ffff 000c 0c0c 0c0c fff2 4500
        0x0010:  0054 aa4b 4000 3f01 825d 0404 0464 0303
        0x0020:  0396 0800 0dc6 2384 38db 5275 dd5c 0000
        0x0030:  0000 9e35 0100 0000 0000 1011 1213 1415
        0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425
        0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233
```

Layer 3 Unicast – Glean Scenario

Verify ARP on Remote Leaf

Egress Leaf Verification

```
a-leaf205#show ip arp internal event-history event | grep -F -B 1 172.16.2.2
73) Event:E_DEBUG_DSF, length:127, at 316928 usecs after Wed May  1 08:31:53 2019
Updating epm ifidx: 1a01e000 vlan: 105 ip: 172.16.2.2, ifMode: 128 mac: 0000.1111.2222
75) Event:E_DEBUG_DSF, length:152, at 316420 usecs after Wed May  1 08:31:53 2019
log_collect_arp_pkt; sip = 172.16.2.2; dip = 172.16.2.254; interface = Vlan104;info = Garp Check adj:(nil)
77) Event:E_DEBUG_DSF, length:142, at 131918 usecs after Wed May  1 08:28:36 2019
log_collect_arp_pkt; dip = 172.16.2.2; interface = Vlan104;iod = 138; Info = Internal Request Done
78) Event:E_DEBUG_DSF, length:136, at 131757 usecs after Wed May  1 08:28:36 2019
log_collect_arp_glean;dip = 172.16.2.2;interface = Vlan104;info = Received pkt Fabric-Glean: 1
79) Event:E_DEBUG_DSF, length:174, at 131748 usecs after Wed May  1 08:28:36 2019
log_collect_arp_glean; dip = 172.16.2.2; interface = Vlan104; vrf = CiscoLive2020:vrf1; info = Address in PSVI subnet or special VIP
```

Endpoint Learn Installed

Response Received

ARP Request is generated by leaf

Glean Received, Dst IP is in BD Subnet

How ACI Builds Forwarding Tables

Building Adjacency Tables

ACI combines ARP and MAC Tables into the Endpoint Table

Legacy Behavior

- ARP/ND tables map Layer 3 to Layer 2
- ARP/ND tables are updated by control-plane messages
- MAC Address Table used for switching decisions
- Mac Address Table updated by dataplane

ACI Behavior

- Endpoint table contains endpoints, which are Layer 2 addresses OR Layer 3 addresses OR a combination of Layer 2 and Layer 3 addresses
- By default, both Layer 2 and Layer 3 information is updated by dataplane
- Used for security and forwarding policy

Building Endpoint Tables

Endpoints can be programmed via software process or by hardware dataplane learns (HAL)

Resource	Table Info	Commands to Verify
Supervisor	EPM – Endpoint Manager Sup process for managing endpoints.	show system internal epm endpoint mac <addr> show system internal epm endpoint ip <addr>
Line Card	EPMC – Endpoint Manager Client Line card process that sits between hardware layer (HAL) and EPM	vsh_lc –c “show system internal epmc endpoint mac <addr>” vsh_lc –c “show system internal epmc endpoint ip <addr>”
Asic	HAL – Hardware Abstraction Layer View of what is programmed into the ASIC.	vsh_lc -c "show plat internal hal ep l2 mac <addr>" vsh_lc -c "show plat internal hal ep l3 ip <ip/pfx len>“ ! !L3 Endpoints are put into HW Routing Table vsh_lc -c "show plat internal hal l3 routes | grep EP"

What about ARP?

ARP Tables are still used in ACI for…

- L3outs
- Overlay adjacencies
  - VXLAN Endpoints (AVE, K8s, Openstack, etc)
  - APIC / Fabric node adjacencies

Resource	Table Info	Commands to Verify
Supervisor	Adjacency Manager. Programmed by ARP process.	show ip arp vrf <name>
Line Card	UFIB	vsh_lc –c “show forwarding adjacency <ip>”
Asic	HAL – Hardware Abstraction Layer View of what is programmed into the ASIC.	vsh_lc –c "show plat internal hal l3 routes”

Building Routing Tables

Resource	Table Info	Commands to Verify
Supervisor	URIB / MRIB – the unicast and multicast routing tables. Programmed by route protocol	show ip route x.x.x.x/y vrf <name> show ip mroute x.x.x.x/y vrf <name>
Line Card	UFIB / MFIB – the unicast and multicast forwarding tables on the Line Card	vsh_lc -c "show forwarding route <ip/pfx len> vrf <name>“ vsh_lc -c "show forwarding multicast route vrf <name>"
Asic	HAL – Hardware Abstraction Layer View of what is programmed into the ASIC.	vsh_lc -c "show platform internal hal l3 routes vrf <name>” vsh_lc -c "show platform internal hal l3 mcast routes vrf <name>” vsh_lc -c "show plat internal hal l3 routes vrf <name>" | grep MC

Troubleshooting TIP

Check Endpoint Table before Routing Table

When Troubleshooting Layer 3 Flows Always…

1) Check if there is an Endpoint Learn

```
show endpoint ip <addr>
show system internal epm endpoint ip <addr>
```

If not then…

2) Check if there is a BD (pervasive) static route

If not then…

3) Check if there is an External Route

```
show ip route x.x.x.x/y vrf <name>
```

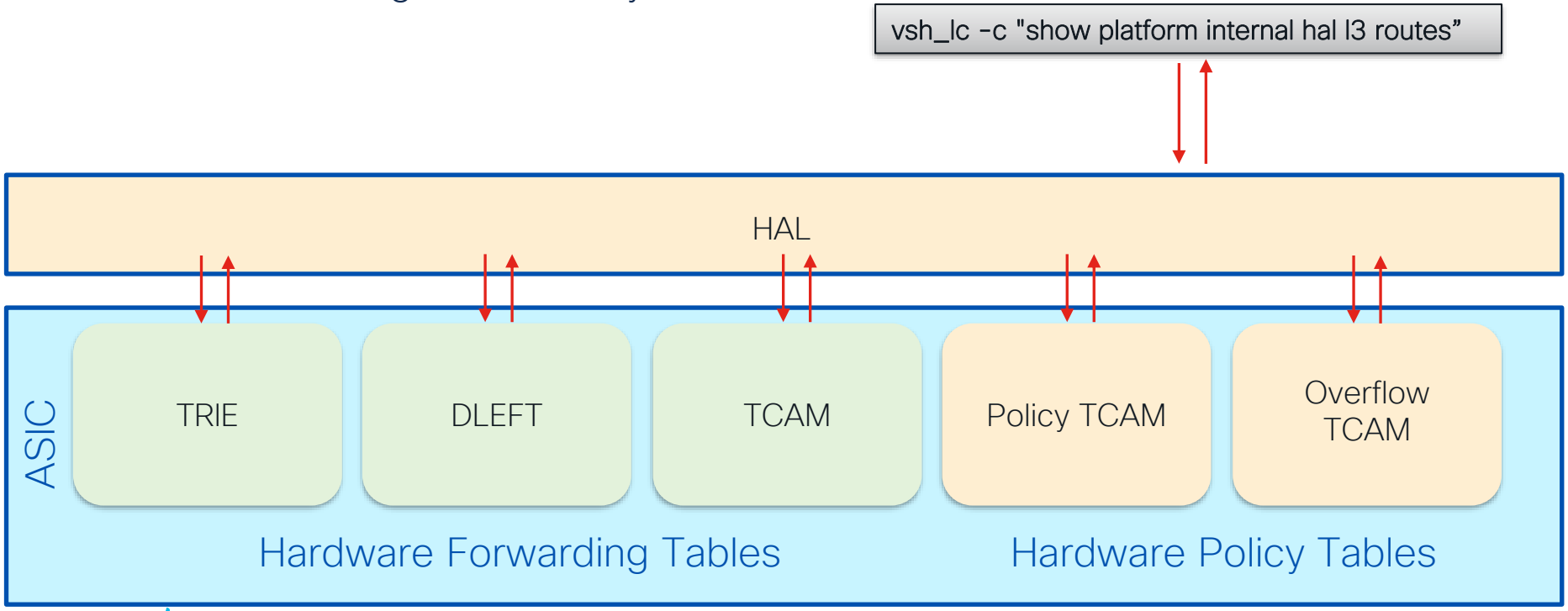
Programming Contracts

Resource	Table Info	Commands to Verify
Supervisor	Policy Manager. Programmed by leaf policy-element process	show zoning-rules
Line Card	ACLQOS	vsh_lc -c “show system internal aclqos zoning-rules”
Asic	HAL – Hardware Abstraction Layer View of what is programmed into the ASIC.	vsh_lc -c “show plat internal hal objects policy zoningrule”

HAL – Hardware Abstraction Layer

Applicable to EX and Later Hardware

Wouldn't it be great if there was a single point to validate forwarding and security classification?



HAL – Hardware Abstraction Layer

Applicable to EX and Later Hardware

L3 Lookup of Hardware Tables

```
module-1# show plat internal hal l3 routes vrf CL2022:vrf1
-----------------------------------------!!-------------------------
|     |                       |    | LID |!!                        |
| VRF | Prefix/Len            | RT| Type|!!!|CLSS| Flags             |
|-----|-----------------------|---|-----|!!!|----|------------------|
|-----|-----------------------|---|-----|!!!|----|------------------|
| 4626|    192.168.100.10/ 32| EP  TRIE|!! c002| le,bne,sne, dl    |
| 4626|        10.99.99.0/ 24| UC  TCAM|!! 8004| sc,spi,dpi        |
| 4626|     192.168.255.0/ 24| UC  TCAM|!!   24| sc,spi,dpi, dr    |
| 4626|    192.168.200.11/ 32| EP  TRIE|!! 8003| sc, le,sne        |
 -----------------------------------------!!-------------------------|
```

Much more info available in full output!

Consolidated view of routes for Endpoints, Shared Services, and External routes

PcTag from destination EPG…used for contract lookup

HAL – Hardware Abstraction Layer

L2 Lookup of Hardware Tables

Applicable to EX and Later Hardware

```
module-1# show platform internal hal ep l2 all
==============================================================

     BD            EP                  L2        L2          S
BdId Name       T  Mac                 IfId      Ifname      Class
==============================================================
b    BD-11      Pl 00:00:11:11:22:22 1a010000 Eth1/17     c003
1a   BD-26      Xr 00:00:22:22:33:33 18010004 Tunnel4     400f
21   BD-33      Pl 00:00:22:22:33:33 16000002 Po3         4002
```

Much more info available in full output!

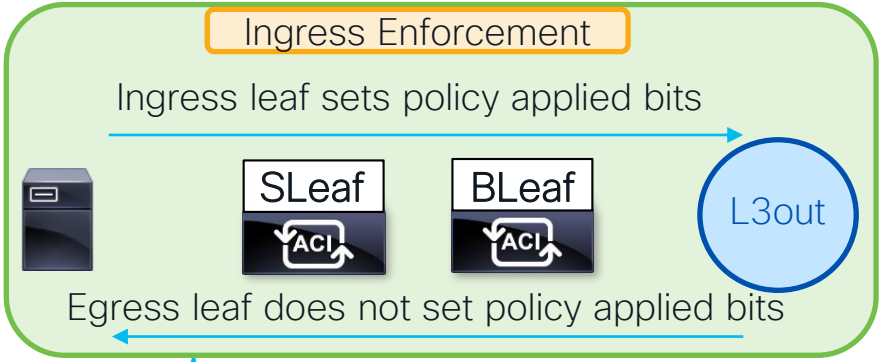
Consolidated view of all learned Mac Addresses

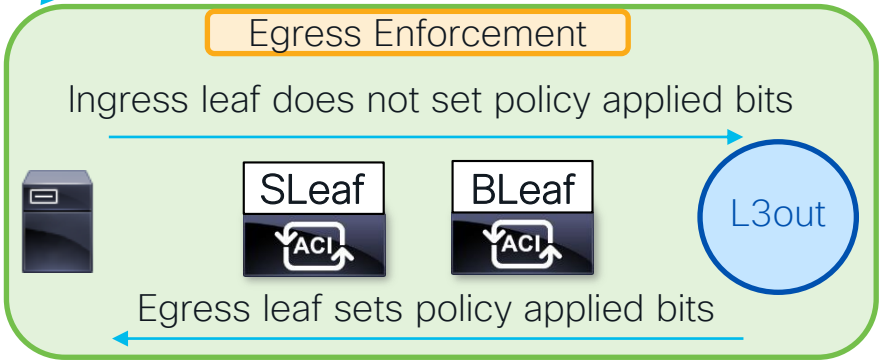
PcTag from destination EPG…used for contract lookup

Understanding the Configuration Options

VRF Level Forwarding Options

Feature	What Does it Do?
Policy Control Enforcement Preference	If disabled, policy is never applied between EPGs. If enabled, contracts are enforced.
IP Dataplane Learning	If Disabled, ACI uses legacy behavior for learning endpoints. Layer 3 endpoints are learned by ARP/GARP/ND and Layer 2 endpoints are learned by dataplane.
Policy Control Enforcement Direction	If set to Ingress, contract enforcement for l3out flows is done on service leaf. Egress enables enforcement on Border Leaf (requires remote learning to be enabled)





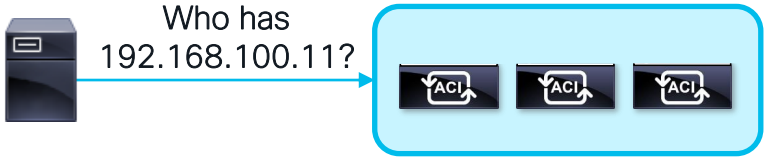
Bridge-Domain Level Forwarding Options

Feature	What Does it Do?
L3 Unknown Multicast Flooding	For non-link-local L3 multicast traffic in a PIM-disabled BD, should a leaf with no snooping entries flood in BD (flood) or wait for joins (OMF)?
Multidestination Flooding	For L2 mcast and broadcast, flood, drop, or flood within epg encap? If flooding with EPG encap, proxy-arp is required for cross-epg L2 communication
L2 Unknown Unicast	If destination mac is unicast and unknown, flood or proxy to spines?

Proxied, L2 Unknown Unicast is dropped if the Destination MAC isn't known in COOP

Bridge-Domain Level Forwarding Options

Feature	What Does it Do?
Limit IP Learning to Subnet	Only learn IP's if they are within the configured BD subnet for local learns.
Unicast Routing	Enable IP learning as well as routing (if a BD subnet is configured)
Disable IP Dataplane Learning	Only for PBR! Only local MAC's are learned via DP. IP's and remote macs learned via ARP.
ARP Flooding	When disabled, ARP is unicast routed based on the Target IP (if known)




```
leaf# show endpoint ip 192.168.100.11
leaf# show ip route 192.168.100.11 vrf CL2022:vrf1

192.168.100.0/24, ubest/mbest: 1/0, direct, pervasive
    *via 10.0.176.66%overlay-1, [1/0], 01w00d, static
        recursive next hop: 10.0.176.66/32%overlay-1
```

Proxy!

EPG Level Forwarding Options

Feature	What Does it Do?
Flood in Encapsulation	Feature is enabled for just the EPG (rather than all epg's in the BD). Requires proxy arp for L2 traffic between encaps.
L4-L7 Virtual IP's	Designed for Direct Server Return flows. This disables dataplane learning per IP. IP is learned by ARP/ND.
Disable DP Learning Per-IP/Prefix	Disables dataplane learning for non DSR scenarios. More specific than VRF-level option

New in 5.2

Global Forwarding Options

Feature	What Does it Do?
Enforce Subnet Check	Don’t learn an IP (both local and remote) if it is not within a configured BD subnet in the VRF.
Disable Remote EP Learning on BL’s	Remote IP learning is disabled for Unicast flows on a leaf in a specific VRF if an l3out exists in the same VRF

Multicast sources are still learned

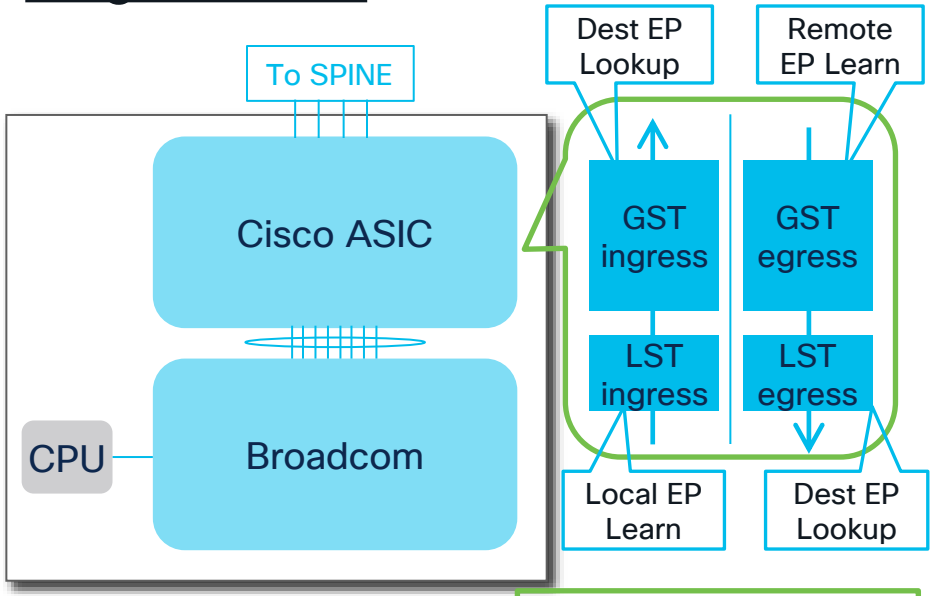
Also implicitly disabled when intersite l3out is configured

The Anatomy of an ACI Switch

LEAF ASIC Generations

※ LST: Local Station Table, GST: Global Station Table
※ FP Tile: Forwarding and Policy Tile

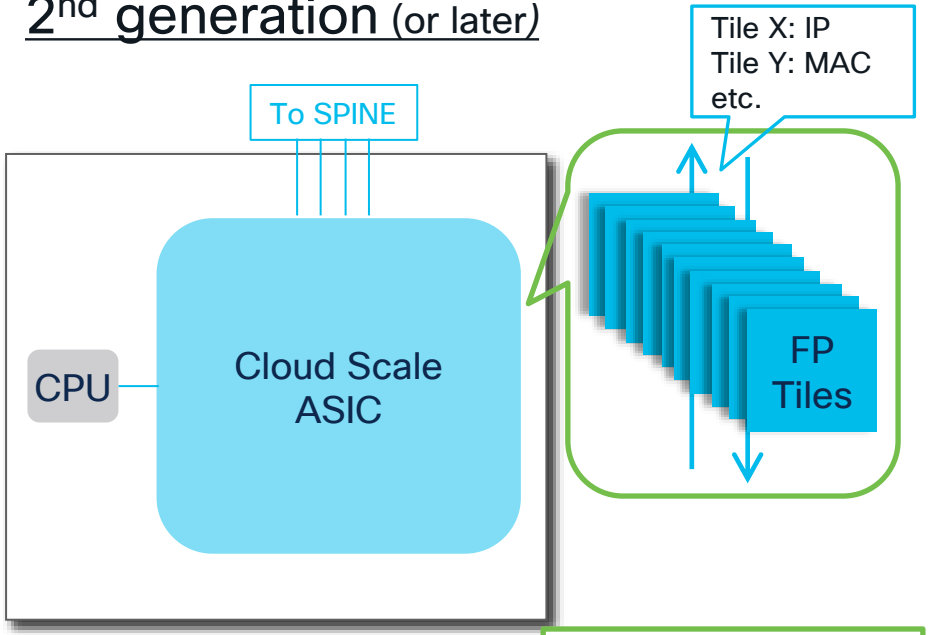
1st generation



N9K-C9332PQ
N9K-C9372PX
N9K-C9372PX-E
N9K-C9372TX
N9K-C9372TX-E
N9K-C9396PX
N9K-C9396TX
N9K-C93120TX
N9K-C93128TX

- Complete separation of
  + Ingress and Egress
  + Source Learn and Destination Lookup
- Separate GST/LST for IP and MAC

2nd generation (or later)



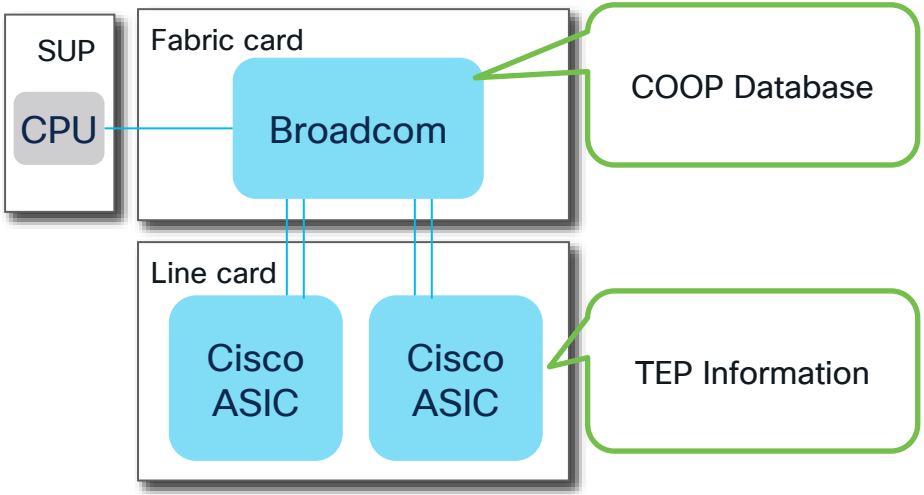
N9K-C*-EX
N9K-C*-FX
N9K-C*-FX2
N9K-C*-FX3
N9K-C*-FXP
N9K-C*-GX
N9K-C*-GX2

- More flexible/scalable with configurable tiles
- Abstracted with HAL
- Tile X for both source learn and destination lookup

SPINE ASIC Generations

※ number of ASIC per card depends on model

1st generation

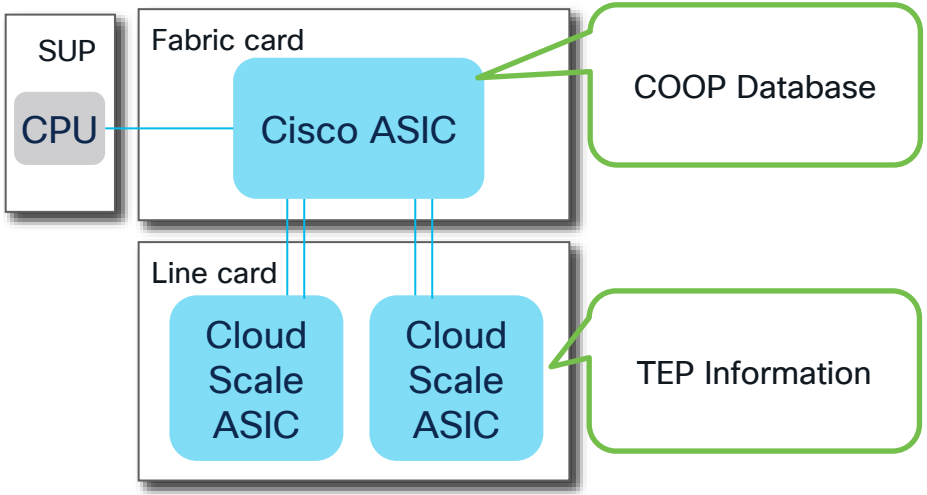


Line card
N9K-X9736PQ

Box spine
N9K-C9336PQ

Fabric card
N9K-C9504-FM
N9K-C9508-FM
N9K-C9516-FM

2nd generation (or later)

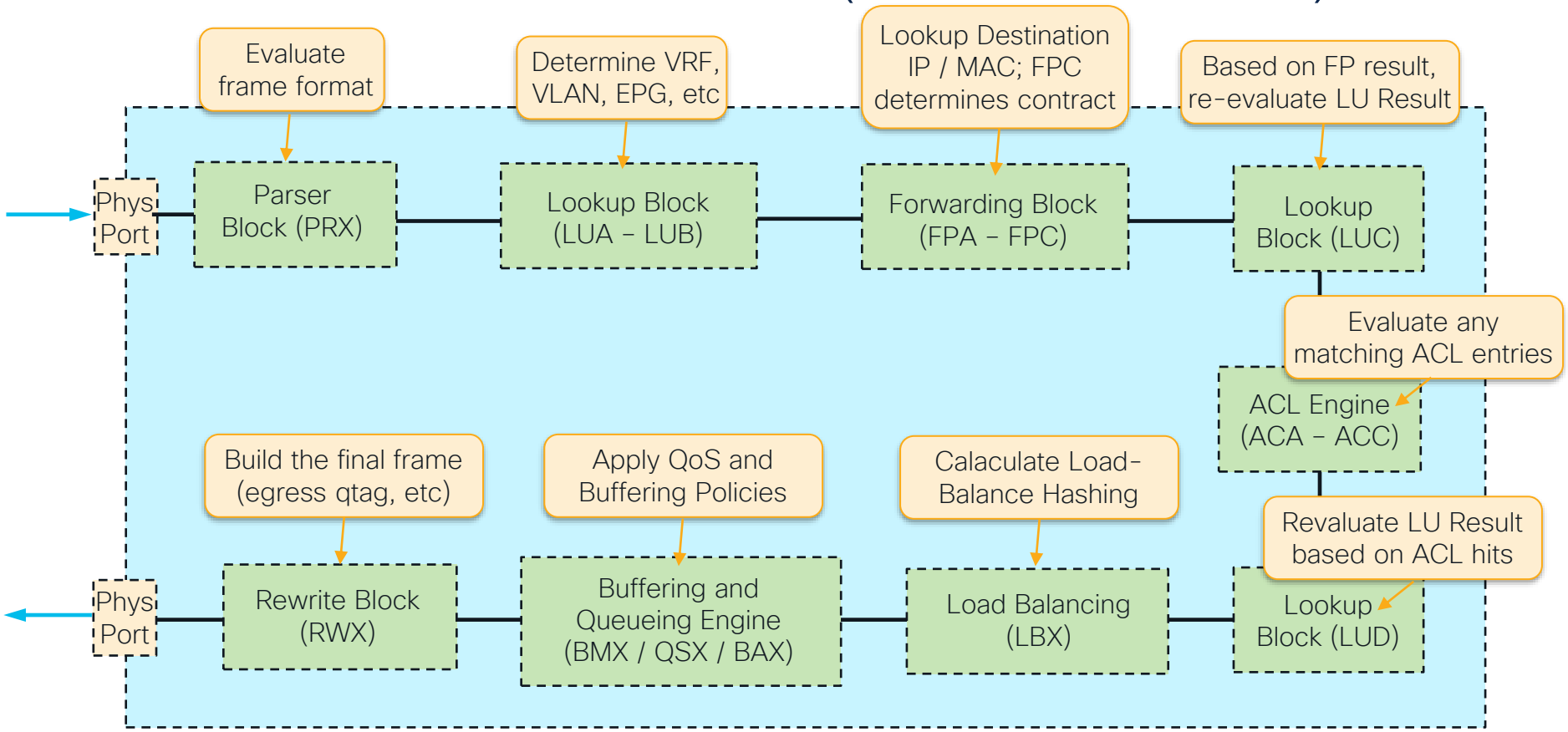


Line card
N9K-*X

Box spine
N9K-*C
N9K-*X

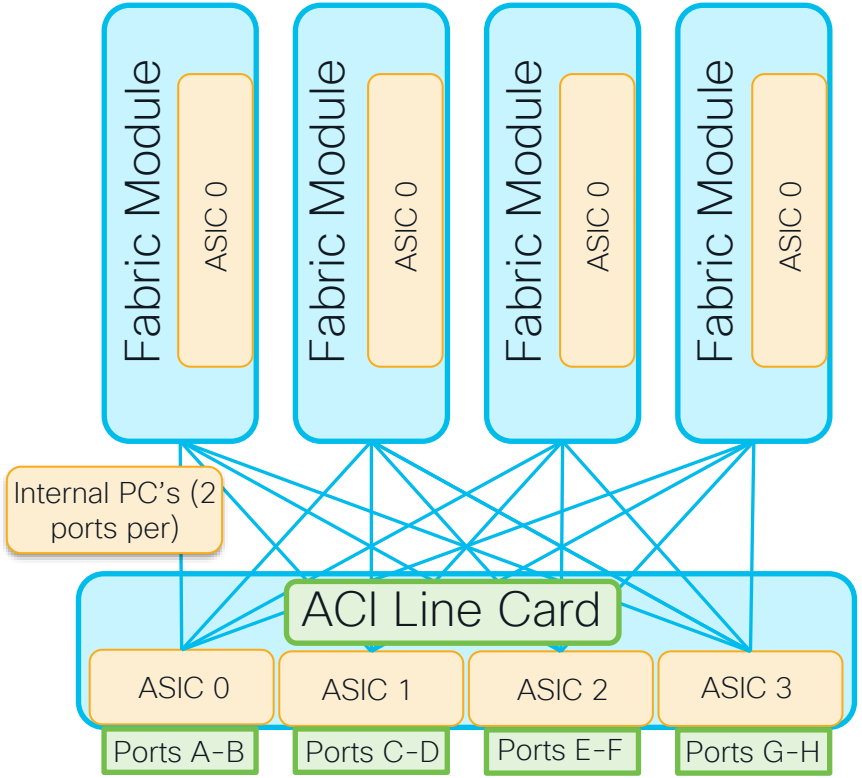
Fabric card
N9K-C*FM-E
N9K-C*FM-E2
N9K-C*FM-G

Inside an ACI Switch ASIC (Gen 2 and Later)



※ number of ASIC per card depends on model

Inside an ACI Modular Spine



What are the strange IP's on the Fabric Modules?

```
sp# vsh -c "slot 26 show plat internal hal l3 routes"
40.0.99.139/ 32
3.124.199.13/ 32
0.156.151.177/ 32
```

Where are the linecard forwarding tables?

```
sp# vsh -c "slot 2 show plat internal hal l3 routes"
<no output>
```

Inside an ACI Modular Spine

How is traffic forwarded?

For Proxied Traffic

- Depending on if the dest IP is the L2 or L3 Proxy TEP the VRF VNID + Dest IP OR BD VNID + Dest MAC is used to hash a synthetic Dest IP and VRF ID
- Synthetic information is used on LC to hash the uplink port to FM
- FM routing lookup is based on Synthetic IP
- Each Synthetic IP is owned by two FM's
- FM uses vnTag to tell egress LC which front panel port to use

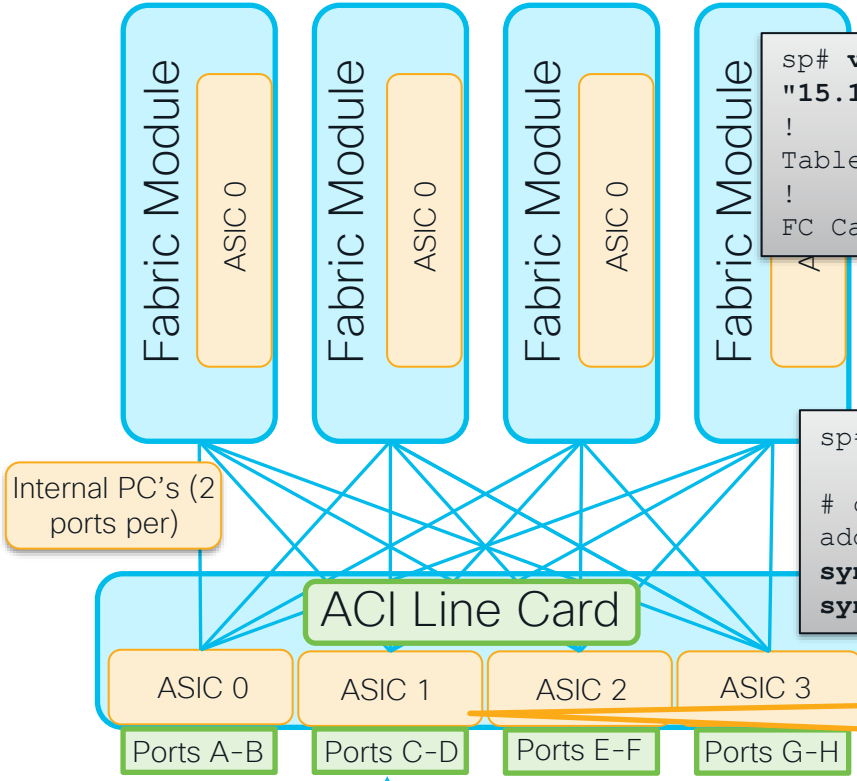
Inside an ACI Modular Spine

How is traffic forwarded?

For Transit Traffic

- Line card hashes across ALL FM uplinks
- ALL FM's have overlay TEP routes
- FM uses vnTag to tell egress LC which front panel port to use

Inside an ACI Modular Spine




```
sp# vsh -c "slot 26 show forwarding route platform" | grep -A 10
"15.180.164.253"
!
Table: (IN-HW) Type=100 Vrf=750 Synth=0
!
FC Cards/ASICs : FC22/ASIC-0     FC26/ASIC-0
```

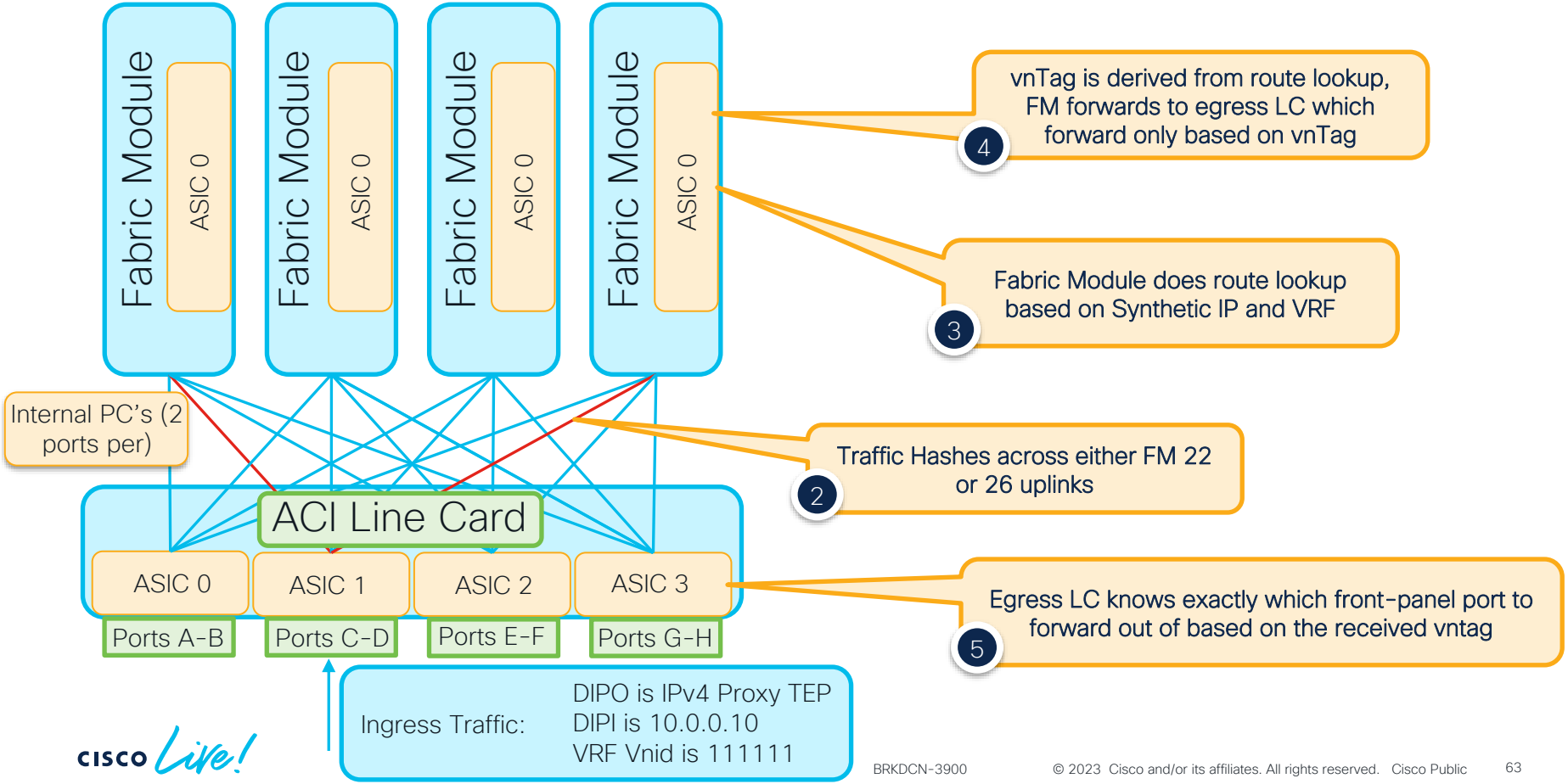
Synthetic info programmed on FM's 22 and 26

```
sp# moquery -c coopIpv4Rec -f 'coop.Ipv4Rec.addr=="10.0.0.10"'

# coop.Ipv4Rec
addr          : 10.0.0.10
synthIp       : 15.180.164.253
synthVrf      : 250
```

1 Ingress LC hashes Synth IP and VRF based on Real Tenant IP + VRF

Ingress Traffic:
DIPO is IPv4 Proxy TEP
DIPI is 10.0.0.10
VRF Vnid is 111111

Inside an ACI Modular Spine



Understanding the Tools

Start with High-level Tools

Use Endpoint Tracker for Building a Topology

System | Tenants | Fabric | Virtual Networking | Admin | **Operations** | Apps | Integrations

Visibility & Troubleshooting | Capacity Dashboard | **EP Tracker** | Visualization

EP Tracker

EP Locally Learned on pod 2, nodes 401-402

End Point Search

172.16.31.100

Search

Learned At	Tenant	Application	EPG	IP
2/401-2/402, vPC: vpc-esxi-10.2.10.19 (learned,vmm)	CiscoLive	Database	DB	172.16.31.100

No EP Learn, is this an L3out?

End Point Search

10.255.255.100

Search

Learned At	Tenant	IP
	No items have been found.	

Start with High-level Tools

Use Atomic Counters to Check for Overlay Drops and Latency (PTP)

Add EP to EP Policy

Name: CL-AC

Description: optional

Administrative State: Disabled | **Enabled**

Features: [x] Atomic Counter
[ ] Latency Statistics

Source Type: EP | **IP**

Source IP: Database (Application Profile) | EPG - DB (EPG/ESG) | 00:50:56:9A:65:DB (Client Endpoint) | 172.16.31.100 (Internet Protocol)

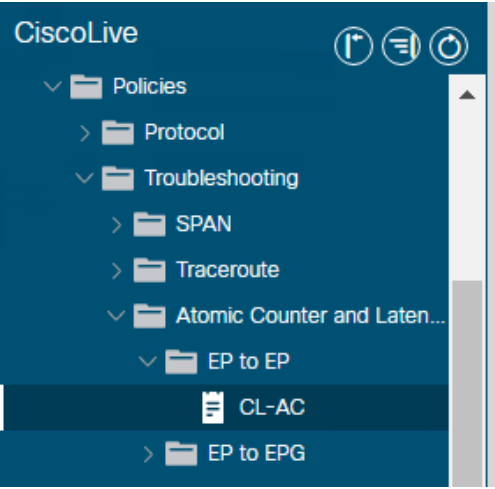
Destination IP: APP (Application Profile) | EPG - WEB (EPG/ESG) | 00:50:56:9A:66:6E (Client Endpoint) | 172.16.32.200 (Internet Protocol)

Filters:

Name	Protocol	Source port	Destination port	Description
ip	Unspecified	Unspecified	Unspecified	

Start with High-level Tools

Use Atomic Counters to Check for Overlay Drops and Latency (PTP)



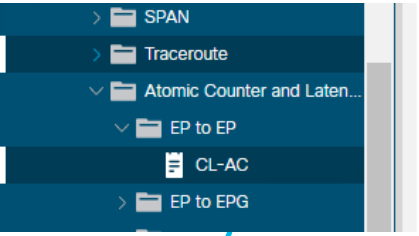
EP to EP CL-AC

EP-to-EP Atomic Counter - CL-AC

Source	Destination	Last Collection (30 seconds) Pkt			
		Transmit	Admitted	Dropped	Excess
uni/tn-CiscoLive/ap-Databas...	uni/tn-CiscoLive/ap-APP/epg...	29	29	0	0

104 Microseconds of delay in overlay

No overlay drops!



EP-to-EP Latency Average - CL-AC

Last 30 Seconds Collection 04/25/2022 16:06:05			Cumulative (04/25/2022 15:04:45 - 04/25/2022 16:06:05)		
Average(µs)	Standard Deviation(µs)	Packet Count	Average(µs)	Max(µs)	Packet Count
104.8575	0.0000	29	104.8575	104.8575	3768

Start with High-level Tools

Use Tenant Visibility tools to check for Contract Drops

CiscoLive

- Quick Start
- CiscoLive
  - Application Profiles
  - Networking
  - Contracts
  - Policies
  - Services
  - Security

Tenant - CiscoLive

Summary | Dashboard | Policy | Operational | Stats | Health | Faults | History

Endpoints | Flows | Packets | Policy Tags | Resource IDs

L2 Permit | L3 Permit | L2 Drop | L3 Drop

This flow is being contract dropped

Timestamp	VRF	Src IP	Dest IP	Protocol	Src Port	Dest Port	Node
2022-04-25T17:19:44.070+00:00	CustA	172.16.31.100	10.255.255.100	icmp	unspecified	unspecified	node-402
2022-04-25T17:19:39.430+00:00	CustA	172.16.31.100	10.255.255.100	icmp	unspecified	unspecified	node-402
2022-04-25T17:18:53.350+00:00	CustA	172.16.31.100	10.255.255.100	icmp	unspecified	unspecified	node-402
2022-04-25T17:11:12.545+00:00	CustA	172.16.31.100	10.255.255.100	icmp	unspecified	unspecified	node-402
2022-04-25T17:18:52.870+00:00	CustA	172.16.31.100	10.255.255.100	icmp	unspecified	unspecified	node-402
2022-04-25T17:18:52.326+00:00	CustA	172.16.31.100	10.255.255.100	icmp	unspecified	unspecified	node-402

```
apic4# show acllog deny l3 pkt tenant common  vrf  CORE
srcIp dstIp protocol srcPort dstPort node srcIntf vrfEncap
----- ----- -------- ------- ------- ---- ------- --------
<EMPTY>
```

Start with High-level Tools

Port Counters are as Useful as Ever

```
leaf1# show interface eth1/8
Ethernet1/8 is up
admin state is up, Dedicated Interface
  Last link flapped 03:07:41
  RX
    3527922 unicast packets  !ommitted
    4041582 input packets  609518993 bytes
    12 jumbo packets  0 storm suppression bytes
    0 runts  0 giants  0 CRC  0 Stomped CRC  0 no buffer
    0 input error  0 short frame  0 overrun !ommitted
    0 watchdog  0 bad etype drop  0 bad proto drop !ommitted
    0 input with dribble  0 input discard
    0 input buffer drop  0 input total drop
  TX
    32262479565 unicast packets  !ommitted
    32395063346 output packets  49034781261
    32249687943 jumbo packets
    0 output error  0 collision  0 deferred
    0 lost carrier  0 no carrier  0 babble  0 output discard
    0 output buffer drops  0 output total drops
```

Frames received with bad FCS

Frame transmitted with stomped CRC

Buffer drops, sign of congestion

Indicates a previously stomped frame was received

What is a Stomp?

- When a frame is received with a bad FCS and/or is malformed

AND

- The frame is cut-through switched

The switch will invert the new CRC to tell the first store-and-forward device to drop it

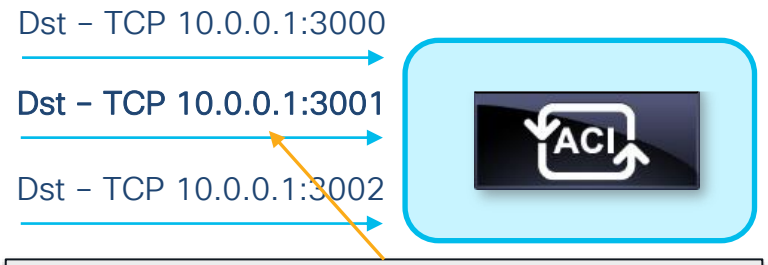
Start with High-level Tools

Using moquery to check port counters fabric-wide

```
#Check Fabric-wide for FCS Errors
moquery -c rmonDot3Stats -f 'rmon.Dot3Stats.fCSErrors>="1"' | egrep "dn|fCSErrors”

#Check Fabric-wide for total CRC Stomp + FCS Errors
moquery -c rmonEtherStats -f 'rmon.EtherStats.cRCAlignErrors>="1"' | egrep "dn|cRCAlignErrors”

#Check Fabric-wide for Output Buffer Drops
moquery -c rmonEgrCounters -f 'rmon.EgrCounters.bufferdroppkts>="1"' | egrep "dn|bufferdroppkts”

#Check Fabric-wide Output Errors
moquery -c rmonIfOut -f 'rmon.IfOut.errors>="1"' | egrep "dn|errors”
```

ELAM – Embedded Logic Analyzer Module

- It is a tripwire in hardware
- The first frame to match a specified condition 'trips' it
- Report is created with vast amount of data regarding asic decisions




```
vsh_lc
debug platform internal tah elam asic 0
 trigger reset
 trigger init in-select 6 out-select 1
  set outer ipv4 dst_ip 10.0.0.1
  set outer l4 dst-port 3001
  start
```

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

module-1(DBG-elam-insel6)# ereport | grep "drop reason"
RW drop reason                          : no drop
LU drop reason                          : no drop
```

Matching frame was caught!

Frame was not dropped in lookups!

What ASIC should be set in the ELAM?

```
vsh_lc
debug platform internal <asic> elam asic 0
```

Model	Role	Asic for Elam
N9K-C*C	Fixed Spine	roc
N9K-C*GX	Fixed Spine	app
N9K-C*-EX	Leaf	tah
N9K-C*-FX/FXP/FX2	Leaf	roc
N9K-C*-GX	Leaf	app
N9K-C*-GX2	Leaf	cho
N9K-X97*-EX	Spine LC	tah
N9K-X97*-FX	Spine LC	roc
N9K-X97*-GX	Spine LC	app
N9K-C95*-FM-E	Spine FM	tah
N9K-C950*-FM-E2	Spine FM	roc
N9K-C95*-FM-G	Spine FM	app

Steps to Using Elam on Gen2+ Leaf or Fixed Spine

Elams are run from the line card shell

Refer to "What ASIC should be set in the ELAM" slide

Leafs and fixed spines are single asic switches. Always use asic 0

```
vsh_lc
debug platform internal tah elam asic 0
 trigger reset
 trigger init in-select 6 out-select 0
  set outer ipv4 dst_ip 10.0.0.1
  set outer l4 dst-port 3001
  start
```

Failing to reset the trigger can cause past elam configurations to take effect. Always reset the trigger!

Use 0 or 1

```
module-1(DBG-elam)# trigger init in-select ?
  !ommitted
  14   Outer(l2(vntag)|l3|l4)-inner(l2|l3|l4)-ieth
  6    Outerl2-outerl3-outerl4
  7    Innerl2-innerl3-innerl4
  !ommitted
```

Determines which headers conditions can be matched in. Use 14 or 7 when matching vxlan encapsulated headers.

Steps to Using Elam on Gen2+ Leaf or Fixed Spine

Use "`set outer`" or "`set inner`" depending on in-select and if matching outer or inner headers in vxlan packet

Which headers to match conditions for?

```
vsh_lc
debug platform   ternal tah elam asic 0
 trigger reset
 trigger init in-select 6 out-select 0
  set outer ipv4 dst_ip 10.0.0.1
  set outer l4 dst-port 3001
  start
```

What to match in the header?

Finally enable the elam!

When running `stat` if `Triggered` is seen, this means a matching packet was received

ereport available since 4.2

Reading an Elam

At a high-level…

```
module-1(DBG-elam-insel6)# ereport
!ommitted
-----------------------------------------------------------------
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     : set
TTL                    : 64
IP Protocol Number     : ICMP
Destination IP         : 192.168.200.11
Source IP              : 192.168.100.10
!omitted
Contract Result
Contract Drop          : no
Contract Logging       : no
Contract Applied       : yes
Contract Hit           : yes
```

- ereport provides a simple, human-readable report output
- ereport requires >= 5.2 code for modular spines
- Groups data into outer/inner, headers, and lookup results

ereport available since 4.2

Reading an Elam

At a low-level…

```
report detail | grep -F "---------" | grep -v VECTOR | grep -v end
LU BEGIN ------------------------------
LUA ------------------------------
LUB ------------------------------
LUC ------------------------------
LUD ------------------------------
LU END ------------------------------
*** FP latch results ------------------------------
*** LBX latch results  ------------------------------
*** ACX latch results  ------------------------------
RW BEGIN ------------------------------
RW END ------------------------------
```

- An elam report provides a walkthrough of each ASIC block
- Each decision in each block is recorded
- Refer to “Inside an ACI Switch ASIC” from part 1 for more details
- All output is in HEX

What if Elam Shows a Drop?

ereport available since 4.2

```
ereport
Lookup Drop
-------------------------------------
LU drop reason    : SECURITY_GROUP_DENY
```

Common Drop Reasons

Drop Code	What Does it Mean?	What to Do?
ACL_DROP	For traffic destined to the CPU on an FX switch it is expected and cosmetic. Also seen when traffic was received from a fabric port and the leaf has a remote EP learn with no bounce flag.	Ignore if its an FX switch and destined to local switch IP/process. Otherwise, check for incorrect EP learn.
DCI_*_XLATE_MISS	For multisite / remote-leaf, there was no matching vnid or pctag translation found.	Check contracts between local and remote resources.
INFRA_ENCAP_SRC_TEP_MISS	No route and/or tunnel found back to the outer source IP	Check for a tunnel pointing back to the outer source IP. Also, check for a route in overlay.
SECURITY_GROUP_DENY	**Frame was contract dropped**	**Make sure a contract is configured to allow the flow.**
SRC_VLAN_MBR	Received vlan not programmed on ingress port.	Check if the frame was correct tagged/untagged. Make sure no invalid-path faults exist for the epg.
UC_PC_CFG_TABLE_DROP	No route was found for the destination.	Check the routing table for the destination.
VLAN_XLATE_MISS	Received vlan doesn't exist on the switch.	Check if the frame is tagged with correct vlan. Check for invalid-path faults on the epg.

Steps to Using Elam on Gen2+ Modular Spine

Challenges of Modular Spines

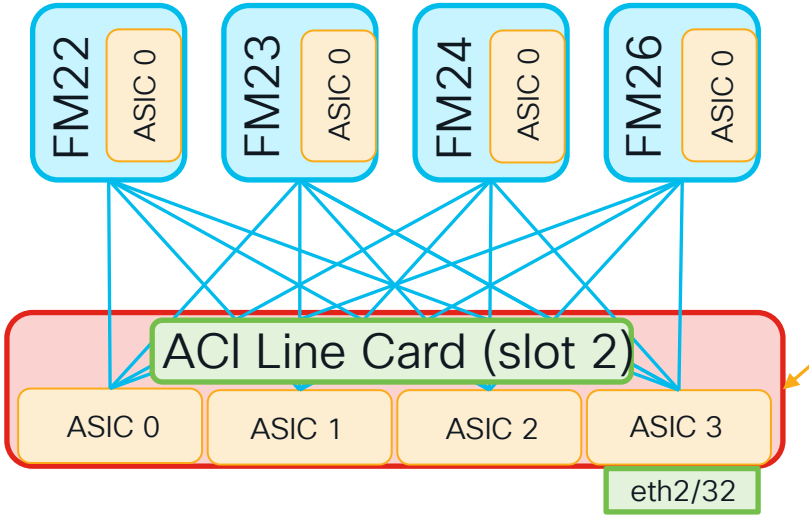
- Line cards (and potentially FM's) have multiple asics
- Elam must specify asic number
- Ingress/Egress ports may be internal LC – FM connections
- ereport only available in 5.2 and later

Fortunately, spine elams aren't needed as commonly as leaf elams!

Steps to Using Elam on Gen2+ Modular Spine

Determine the Asic, Slice, and Srcid of the ingress port

Ingress LC



1

```
sp# vsh
sp# attach mod 2
module-2# show plat internal hal l2 port gpd
=================================================================
                          Uc    Uc
                        I PC    Pc
IfId      Ifname        P Cfg   MbrID  As  AP  Sl  Sp  Ss  Ovec
=================================================================
!ommitted
1a09f000 Eth2/32        0 b9    38     3   31  1   8   10  90
```

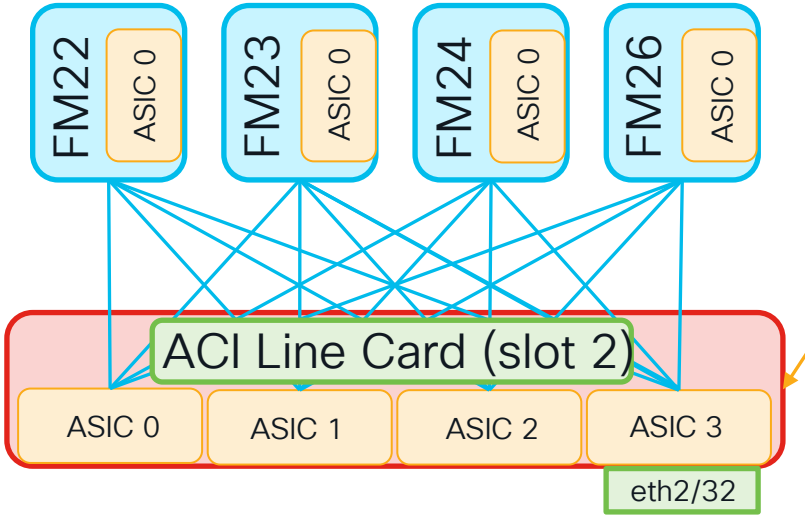
Ingress Traffic:

Inner Headers -
Src – 10.10.10.10
Dst – 10.10.11.11

Eth2/32 is on Asic 3, Slice 1, with srcid 0x10.
Use for Elam!

Steps to Using Elam on Gen2+ Modular Spine

Ingress LC



Ingress Traffic:
Inner Headers -
Src - 10.10.10.10
Dst - 10.10.11.11

Asic and slice of eth1/32 (see last slide)

2

```
sp# vsh
sp# attach mod 2
debug plat internal tah elam asic 3 slice 1
 trigger reset
 trigger init in-select 14 out-select 1
  set srcid 0x10
  set inner ipv4 sr ip 10.10.10.10 dst_ip 10.10.11.11
  start
```

Source ID value of eth1/32 (see last slide)

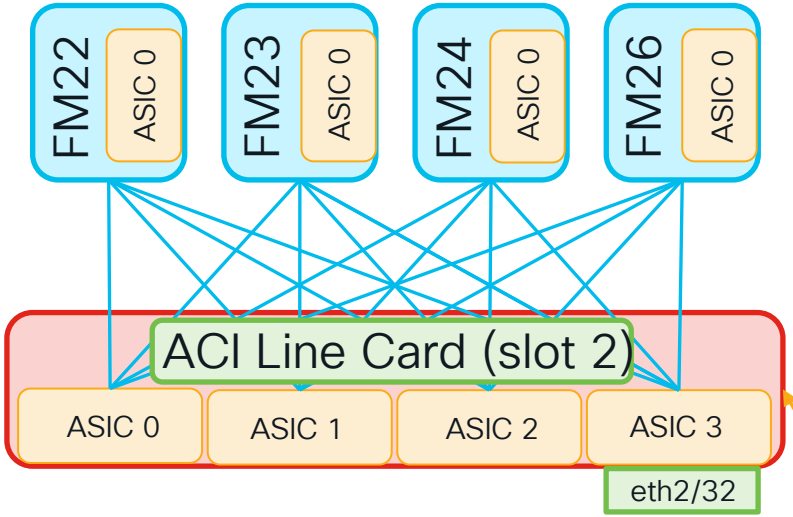
3

```
module-2(DBG-elam-insel14)# stat
 ELAM STATUS
===========
Asic 3 Slice 1 Status Triggered
```

Packet was matched!

Steps to Using Elam on Gen2+ Modular Spine

Ingress LC



Ingress Traffic:
Inner Headers -
Src – 10.10.10.10
Dst – 10.10.11.11

5

```
module-2# show plat internal hal l2 internal-port pi
==========================================
IfId      IfName                          As Ovec
==========================================
96        lc(0)-fc(0):22:pc2:p1           0  b8
98        lc(1)-fc(0):22:pc2:p1           1  b8
9a        lc(2)-fc(0):22:pc2:p1           2  b8
9c        lc(3)-fc(0):22:pc2:p1           3  b8
```

Packet forwarded to FM 23! (output is zero-based)

Ovector indicates the egress port to FM

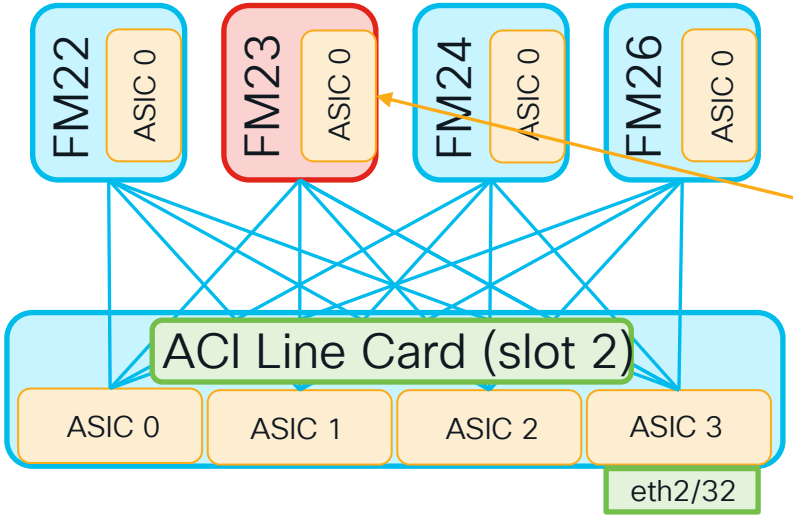
4

```
report | egrep "drop\_vec|ovec|asic"
Dumping report for asic inst 3 slice 1 insel 14 outsel 1
*_sidebnd_no_spare_vec.ovector_idx: 0xB8
*_vec.pbx_header_sidebnd_drop_vec.lux_drop_vec: 0x0000000
```

Packet wasn't dropped in lookups!

Steps to Using Elam on Gen2+ Modular Spine

Fabric Mod



Ingress Traffic:
Inner Headers -
Src – 10.10.10.10
Dst – 10.10.11.11

9508 and 9516 FM's have 2 asics; if no trigger on 0, try 1.

6

```
sp# vsh
sp# attach mod 23
debug plat internal tah elam asic 0
 trigger reset
 trigger init in-select 14 out-select 1
  set inner ipv4 src_ip 10.10.10.10 dst_ip 10.10.11.11
  start
```

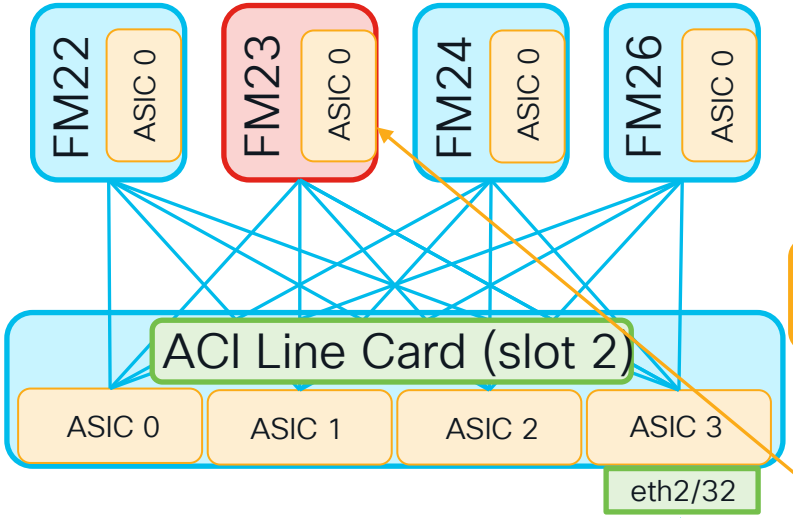
7

```
module-23(DBG-elam-insel14)#
stat
 ELAM STATUS
===========
Asic 0 Slice 0 Status Armed
Asic 0 Slice 1 Status Triggered
Asic 0 Slice 2 Status Armed
Asic 0 Slice 3 Status Armed
Asic 0 Slice 4 Status Armed
Asic 0 Slice 5 Status Armed
```

Packet was matched!

Steps to Using Elam on Gen2+ Modular Spine

Fabric Mod



9

```
module-23# show plat internal hal l2 port gpd
=========================================
IfId      Ifname       As Ovec
============================
f5        fc0-lc1:3-1  0  58
```

Packet forwarded to LC 2
(zero based – Asic 3, Slice 1)

Ovector indicates the egress port to LC

8

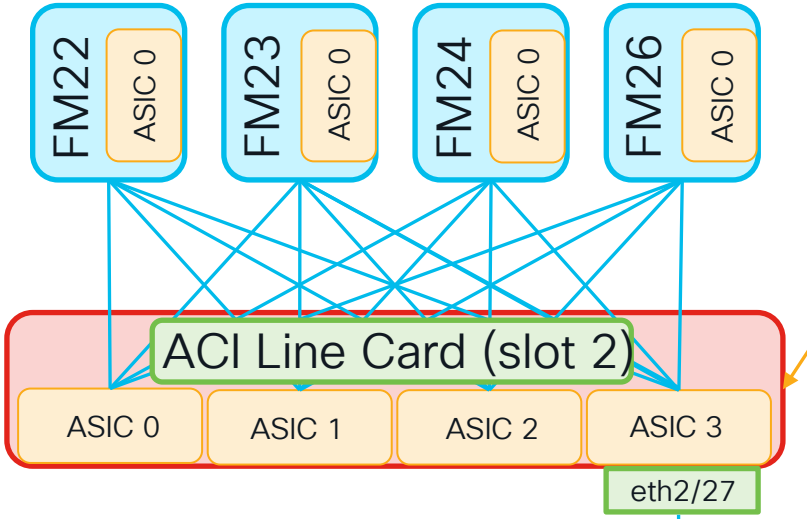
```
report | egrep "drop\_vec|ovec|asic"
Dumping report for asic inst 0 slice 1 insel 14 outsel 1
  *_sidebnd_no_spare_vec ovector_idx: 0x58
*_vec.pbx_header_sidebnd_drop_vec.lux_drop_vec: 0x00000000000
```

Packet wasn't dropped in lookups!

Ingress Traffic:
Inner Headers –
Src – 10.10.10.10
Dst – 10.10.11.11

Steps to Using Elam on Gen2+ Modular Spine

Egress LC



Egress Traffic: Inner Headers – Src – 10.10.10.10 Dst – 10.10.11.11

Asic 3 / slice 1 as seen on last slide

10

```
sp# vsh
sp# attach mod 2
debug plat internal tah elam asic 3 slice 1
 trigger reset
 trigger init in-select 14 out-select 1
  set outer l2 vntag_vld 1
  set inner ipv4 src_ip 10.10.10.10 dst_ip 10.10.11.11
  start
```

Vntag present only coming from FM

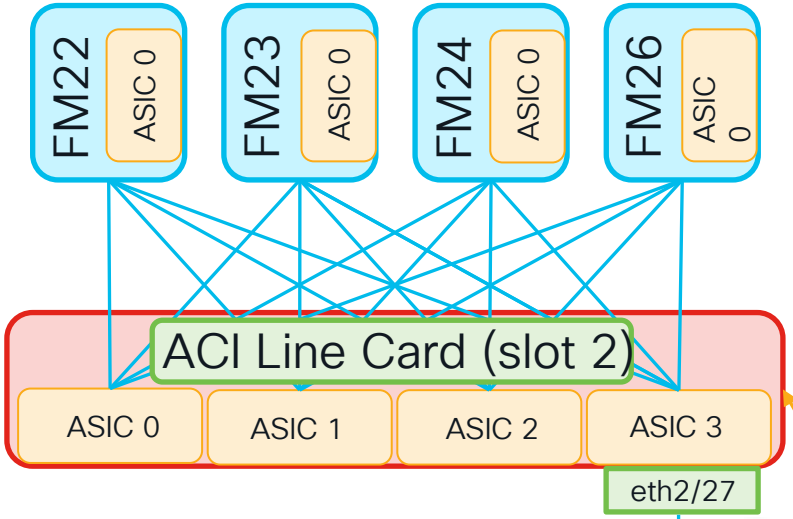
11

```
module-2(DBG-elam-insel14)# stat
 ELAM STATUS
===========
Asic 3 Slice 1 Status Triggered
```

Packet was matched!

Steps to Using Elam on Gen2+ Modular Spine

Egress LC



Egress Traffic: Inner Headers -
Src – 10.10.10.10
Dst – 10.10.11.11

13

```
module-2# show plat internal hal l2 port gpd
=====================================
IfId      Ifname      As AP Sl Sp Ss Ovec
=====================================
1a08a000 Eth2/11      1  5  0  4  8  8
1a09a000 Eth2/27      3  5  0  4  8  8
```

Spine forwards out front-panel Eth2/27!

Ovector indicates the egress port to Leaf

12

```
report | egrep "drop\_vec|ovec|asic"
Dumping report for asic inst 3 slice 1 insel 14 outsel 1
*_sidebnd_no_spare_vec.ovector_idx: 0x8
*_vec.pbx_header_sidebnd_drop_vec.lux_drop_vec: 0x00000000
```

Packet wasn't dropped in lookups!

Automating Modular Spine ELAMs

CLI-based Modular Spine Elam tool available at – EasySpineElam

Easily Set Conditions on All or Some Modules

```
spine1#./easy-spine-elam.sh -m all -d ingress
Final module list is:
2 23 26 3
2022-06-08T14:55:57 In-select - 14 and out-select - 0 are being used.
!ommitted
70. inner ipv4 destination ip       > Format : d.d.d.d
71. inner ipv4 protocol             > Format : 0-255
73. inner ipv4 source ip            > Format : d.d.d.d
91. inner l4 dest port              > Format : 0-65535

    Select corresponding numbers of conditions to set. Separate numbers with commas.
    Ex: 1,2,3,4,5
Enter selections: 70,73,71,91
Enter inner ipv4 destination ip > Format : d.d.d.d: 80.0.0.1
Enter inner ipv4 source ip > Format : d.d.d.d: 150.0.0.100
Enter inner ipv4 protocol > Format : 0-255: 6
Enter inner l4 dest port > Format : 0-65535: 8989
```

Which conditions to match?

Set conditions

Automating Modular Spine ELAMs

CLI-based Modular Spine Elam tool available at – EasySpineElam

```
2022-06-08T14:56:28 Checking elam status for module 2
2022-06-08T14:56:28 Checking elam status for module 23
2022-06-08T14:56:28 Checking elam status for module 26
2022-06-08T14:56:28 Checking elam status for module 3

ELAM TRIGGERED on module 26:
ASIC: 0 SLICE: 1

ELAM TRIGGERED on module 2:
ASIC: 3 SLICE: 1

Type "status" to check elam status again. Type "ereport", "report" or "report detail"
to collect all reports: ereport
2022-06-08T14:57:36 Collecting report for module 26 asic 0...
2022-06-08T14:57:36 Collecting report for module 2 asic 3...
2022-06-08T14:57:46 Converting reports to ereport format!
The following decoded elams are available -
/data/techsupport/mod26-asic0-elamreport-2022-06-08T14-57-36-EREPORT
/data/techsupport/mod2-asic3-elamreport-2022-06-08T14-57-36-EREPORT
2022-06-08T14:57:49 FINISHED!
```

Generate and view ereport from all Triggered Modules!

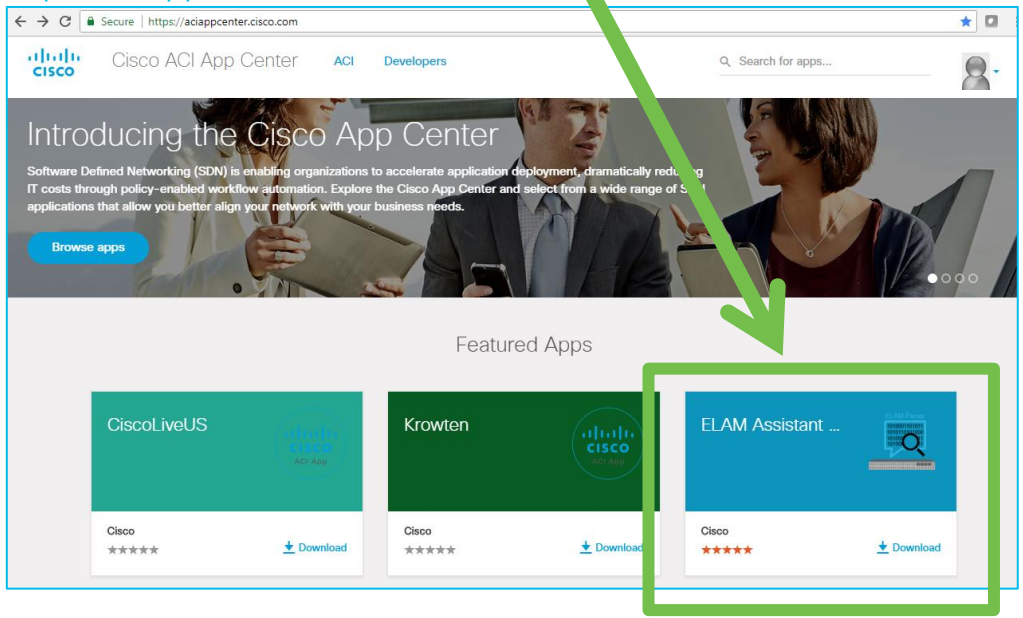
ELAM triggered on LC and FM!

Locally view or copy off the final ereports

Shouldn’t ELAM be More Simple?

Elam Assistant in DCAppCenter

https://dcappcenter.cisco.com



ELAM (Embedded Logic Analyzer Module)

- Perform an ASIC level packet capture

ELAM Assistant

- You can perform ELAM like a TAC engineer!
- With a nicely formatted result report

Detail Explanations:

- https://dcappcenter.cisco.com/elam-assistant.html
- How to use video, pictures
  - ➢ A download link for ELAM Assistant

ELAM Assistant in ACI AppCenter (example)

1. Perform an Elam

System | Tenants | Fabric | Virtual Networking | Admin | Operations | **Apps** | Integrations

Installed Apps | Faults | Downloads

Apps | ELAM Assistant

ELAM Assistant — Capture a packet with ELAM (Embedded Logic Analyzer Module)

- Capture (Perform ELAM)
- node-101 (site2-pod1-leaf1)
- node-102 (site2-pod1-leaf2)
- node-203 (site2-pod1-spine3)
- node-303 (site2-pod2-spine3)
- node-401 (site2-pod2-leaf1)
- node-402 (site2-pod2-leaf2)
- Unsupported Nodes

ELAM Parameters — Quick Add | Add Node

Name your capture

Status	Node	Direction	Source I/F	Parameters		VxLAN (outer) header
Set	node-401	from downlink	any	dst ip	10.255.255.100	
Report Ready	node-402	from downlink	any	dst ip	10.255.255.100	
Report Ready	node-303	from LEAF/IPN	any	dst ip	10.255.255.100	

▶ Set ELAM(s) | Check Trigger

ELAM Report Parse Result (report name:)

Express | Detail | Raw

Select a report.

Triggered!! and Report is Ready

Set Parameters

ELAM Assistant in ACI AppCenter (example)

2. Read a Report

Capture (Perform ELAM)
node-101 (site2-pod1-leaf1)
node-102 (site2-pod1-leaf2)
node-203 (site2-pod1-spine3)
Unsupported Nodes

ELAM Parameters
Quick Add
Add Node
Name your capture

Status	Node	Direction	Source I/F	Paramet
Set	node-401	from downlink	any	
Report Ready	node-402	from downlink	any	
Report Ready	node-303	from LEAF/IPN	any	

Set ELAM(s)

ELAM Report Parse Result (report name: node-402_slot1_

Express Detail Raw

Captured Packet Information

Basic In	
Device Type	L
Packet Direction	ingress (from
Incoming I/F	eth1/4

L2 Header	
Destination MAC	0022.BDF8.19FF
Source MAC	0050.569A.65DB
Access Encap VLAN	844

Packet Forwarding Information

Forward Result	
Destination Type	To another ACI node (LEAF, AVS/AVE etc.)
Destination TEP	10.1.240.33 (MAC Spine-Proxy)
Destination Physical Port	eth1/49

Contract	
Destination EPG pcTag (dclass)	0x4002 / 16386 (L3OUT CiscoLive:L3out-CUST:EEPG2)
Source EPG pcTag (sclass)	0x8005 / 32773 (CiscoLive:Database:DB)
Contract was applied	1 (Contract was applied on this node)

Drop	
Drop Code	no drop

Click to see report

Report shows up here

Scroll Down

FTRIAGE – Automating Elams

Orchestrate End-to-End ELAMs from the APIC!

```
apic1# ftriage route -ii LEAF:101,102 -dip 10.99.99.100 -sip 192.168.100.10
20:19:54 INFO main:1295 L3 packet Seen on  leaf102 Ingress: Eth1/34 (Po5) Egress: Eth1/54  Vnid: 2523136
20:19:55 INFO main:1364 leaf102: Packet's egress outer [SIP:10.0.176.67, DIP:10.0.64.70]
20:19:55 INFO main:1371 leaf102: Outgoing packet's Vnid: 2523136
20:19:56 INFO main:353  Computed ingress encap string vlan-3501
20:20:03 INFO main:464  Ingress BD(s) CL2022:bd1
20:20:03 INFO main:476  Ingress Ctx: CL2022:vrf1 Vnid: 2523136
!
20:21:46 INFO main:1295 L3 packet Seen on  spine1005 Ingress: Eth1/1 Egress: Eth1/3  Vnid: 2523136
20:22:38 INFO  fib:737  spine1005: Transit in spine
20:23:32 INFO main:1295 L3 packet Seen on  leaf103 Ingress: Eth1/29 Egress: Eth1/27/4  Vnid: NULL
!
20:24:02 INFO  fib:219  leaf103:  L3 out interface Ethernet1/27/4
20:24:10 INFO main:781  Computed egress encap string vlan-1055
20:24:17 INFO main:1796 Packet is Exiting fabric with peer-device: N3K-1 and peer-port: Ethernet1/31
```

SPAN / ERSPAN

Don’t neglect old friends!

- Both local span and erspan supported
- ERSPAN requires an l3 endpoint learned anywhere in the fabric
- Still the best tool for checking –
  - Packet contents
  - Frame format
  - Retransmissions
  - …and anything else that can be seen in a pcap

Other Tools Requiring External Resources

Netflow

- Captures flow information based on specified criteria
- Useful for troubleshooting packet loss and latency

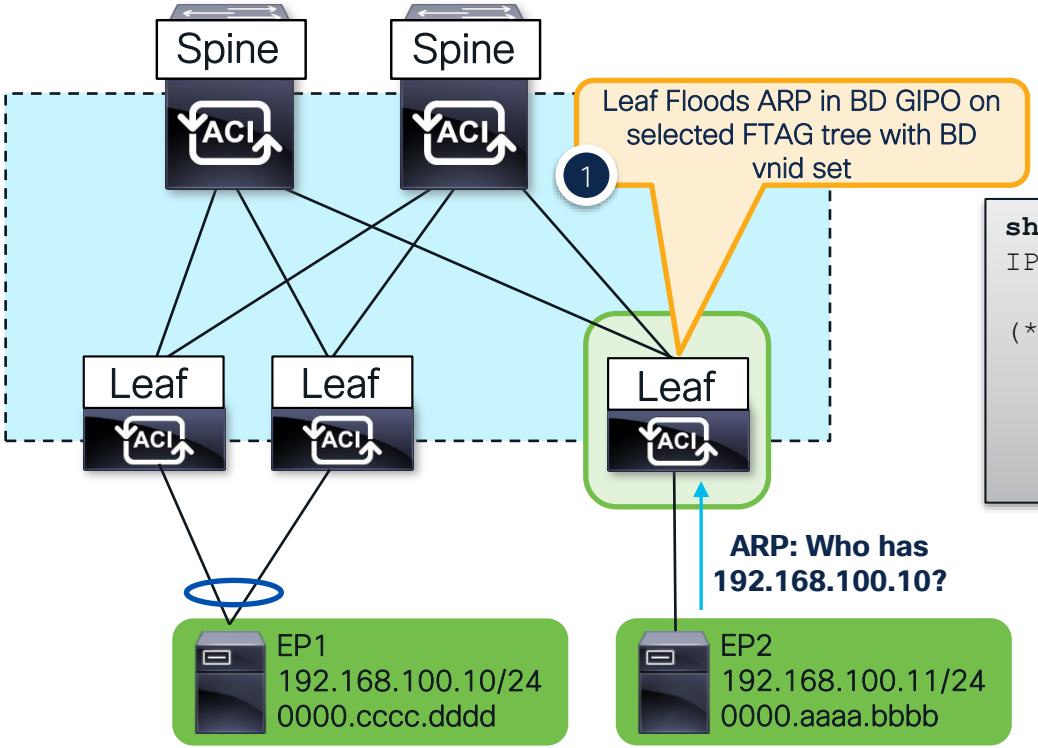
Flow Telemetry

- Hardware directly streams flow data to Nexus Dashboard Insights
- Useful for troubleshooting packet loss and latency
- Latency measurements leverage PTP for additional accuracy
- NDI can perform additional flow analytics

Debugging ACI BUM Flows

ARP – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Disable
ARP Flooding Enabled

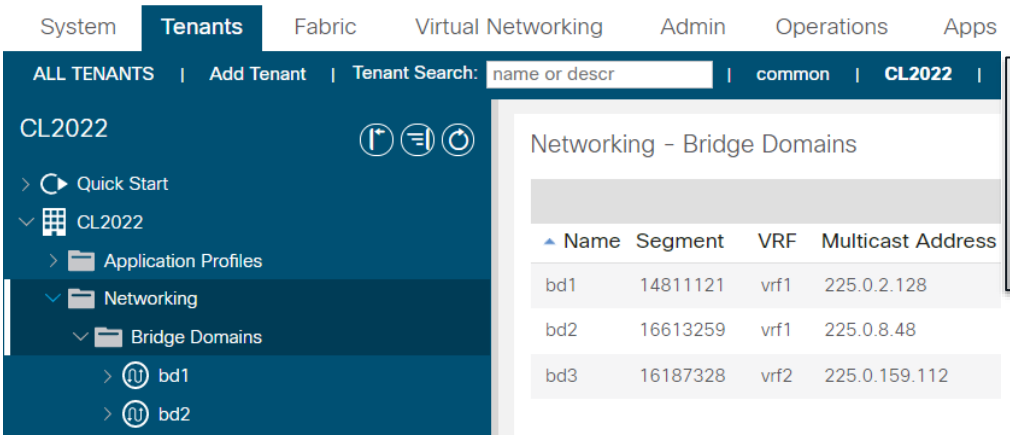


Check GIPO Route

```
show ip mroute 225.0.2.128 vrf overlay-1
IP Multicast Routing Table for VRF "overlay-1"

(*, 225.0.2.128/32), uptime: 22w2d, isis
  Incoming interface: Null, RPF nbr: 0.0.0.0
  Outgoing interface list: (count: 2)
    Ethernet1/29.9, uptime: 8w2d
    Ethernet1/30.10, uptime: 22w2d
```

ARP – How to Find the GiPo

From the GUI…



From the APIC CLI…

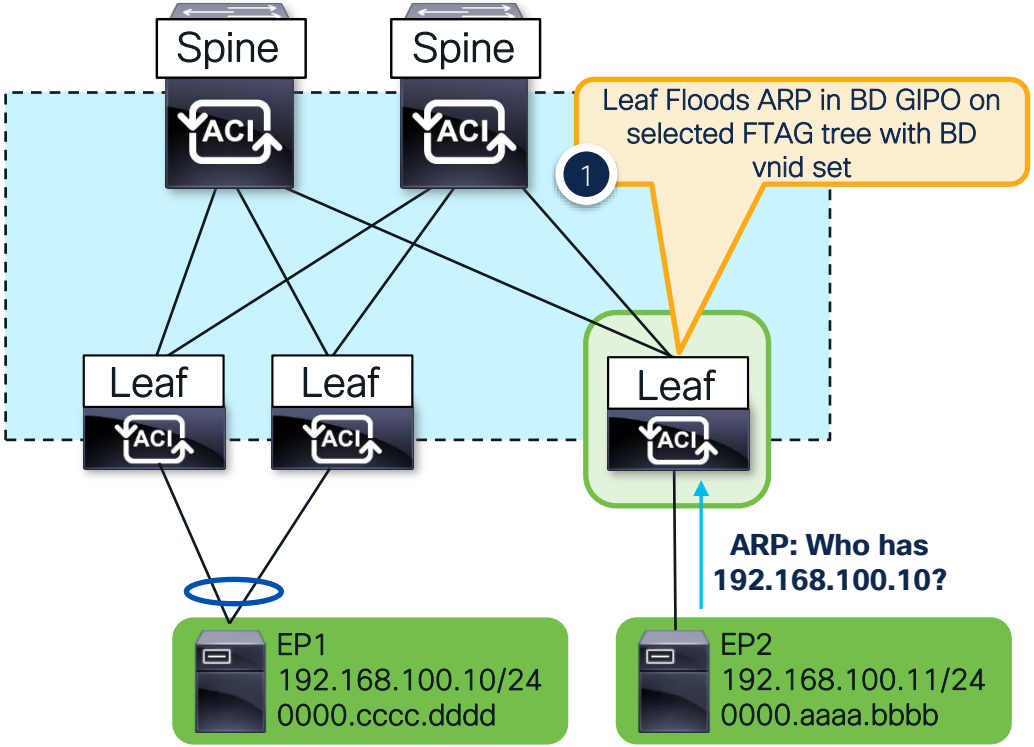
```
moquery -c fvBD -f 'fv.BD.dn*"tn-CL2022/BD-bd1"'

# fv.BD
arpFlood                    : yes
bcastP                      : 225.0.2.128
dn                          : uni/tn-CL2022/BD-bd1
```

From the Switch CLI…

```
moquery -c l2BD -f 'l2.BD.name=="CL2022:bd1"' -x rsp-subtree=full rsp-subtree-class=fmcastGrp
# fmcast.Grp
  addr          : 225.0.2.128
  dn            : sys/ctx-[vxlan-2523136]/bd-[vxlan-14811121]/fmgrp-[225.0.2.128]
  rn            : fmgrp-[225.0.2.128]
```

ARP – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Disable
ARP Flooding Enabled



ELAM the ARP request!

```
vsh_lc
debug plat internal app elam asic 0
 trigger reset
  trigger init in-select 6 out-select 0
  set outer arp source-ip 192.168.100.11
  set outer arp target-ip 192.168.100.10
  start
!
stat
 ELAM STATUS
===========
Asic 0 Slice 0 Status Armed
Asic 0 Slice 1 Status Armed
Asic 0 Slice 2 Status Triggered
Asic 0 Slice 3 Status Armed
```

ARP – Ingress Leaf Elam Results (ereport)

Bridge Domain Settings:
Unicast Routing Disable
ARP Flooding Enabled

```
Outer L2 Header
-----------------------------------
Access Encap VLAN    : 3502( 0xDAE )

Outer L3 Header
-----------------------------------
ARP Opcode           : Request( 0x1 )
ARP Sender IP        : 192.168.100.11
ARP Target IP        : 192.168.100.10

Contract Result
------------------------
Contract Drop        : no
Contract Applied     : no

FINAL FORWARDING LOOKUP
-------------------------------------------------------------------------------
Bits set in Final Forwarding Block:   : IFABRIC_IG MC TENANT MYTEP BRIDGE MISS FLOOD

Lookup Drop
----------------------------
LU drop reason       : no drop
```

Make sure this matches what is expected

Frame is flooded in the Bridge Domain!

Not Dropped in lookups!

ARP – How to Find the FTAG

No other way than Elam…

```
module-1(DBG-elam-insel6)# ereport | grep "nopad.ftag"
    wol_lu2ba_sb_info.mc_info.mc_info_nopad.ftag: 0x8
```

Selected ftag is 0x8

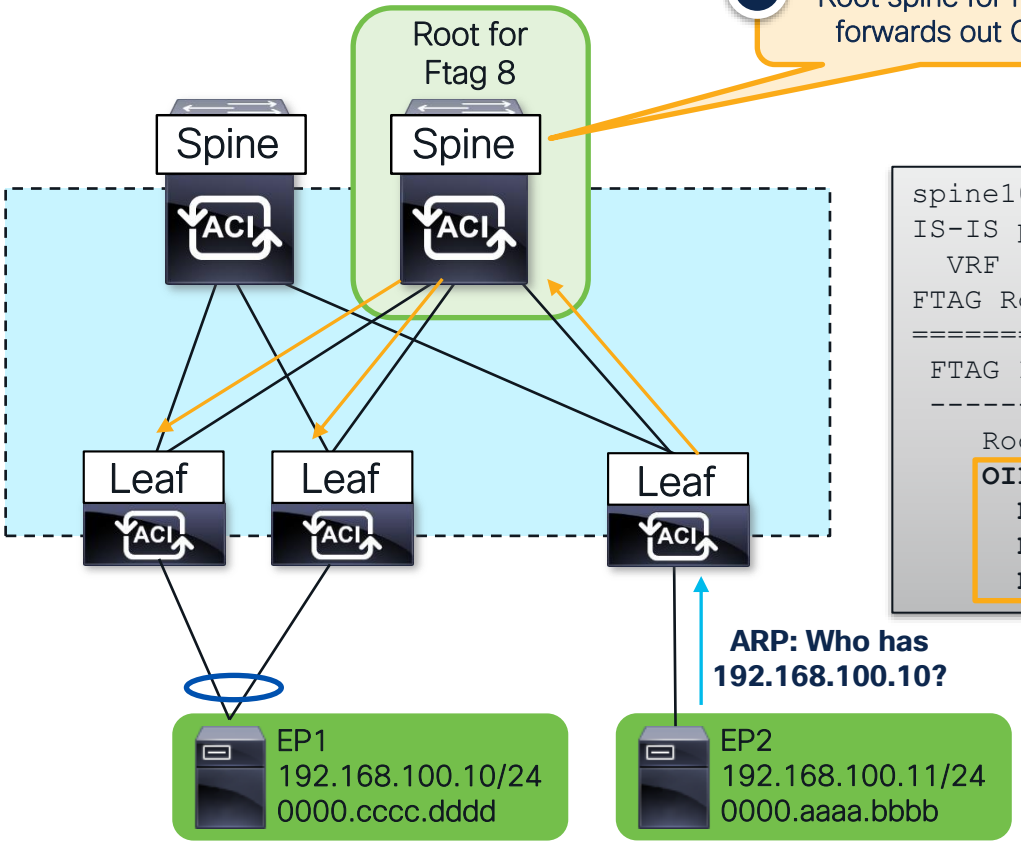
- Leaf forwards to root port and OIF's for ftag 8
- Since GIPO is 225.0.2.128, Dest multicast address is 225.0.2.136 (gipo + ftag)
- Check ftag topology with **`show isis internal mcast routes ftag`**

```
leaf103# show isis internal mcast routes ftag
IS-IS process: isis_infra
  VRF : default
FTAG Routes
====================================
 FTAG ID:   8  [Enabled] Cost:(   1/   6/   0)
 ----------------------------------
    Root port: Ethernet1/29.9
    OIF List:
```

Leaf appends ftag to gipo and forwards out Eth1/29 to spine

ARP - Spine

Bridge Domain Settings:
Unicast Routing Disable
ARP Flooding Enabled



2 Root spine for ftag 8 forwards out OIFs

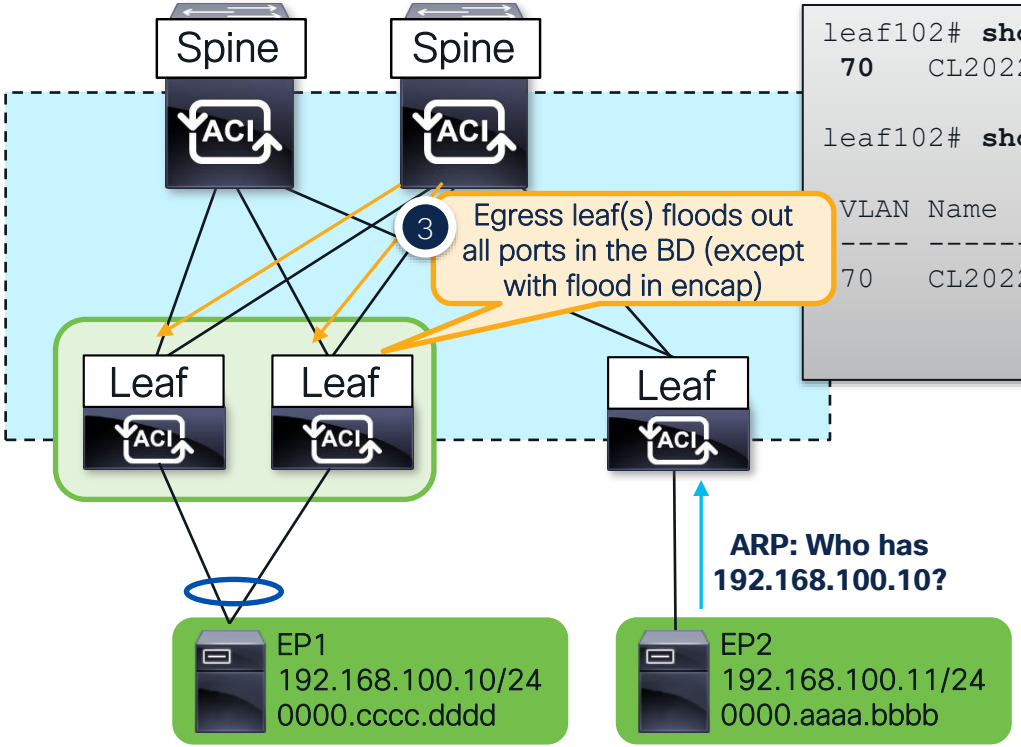
This spine is the root!

```
spine1005# show isis internal mcast routes ftag
IS-IS process: isis_infra
  VRF : default
FTAG Routes
==================================
 FTAG ID:    8 [Root] [Enabled] Cost:(   0/   0/   0)
 ---------------------------------
    Root port: -
    OIF List:
      Ethernet1/1.20
      Ethernet1/2.21
      Ethernet1/3.19
```

Spine forwards out OIFs

ARP – Egress Leaf

Bridge Domain Settings:
Unicast Routing Disable
ARP Flooding Enabled

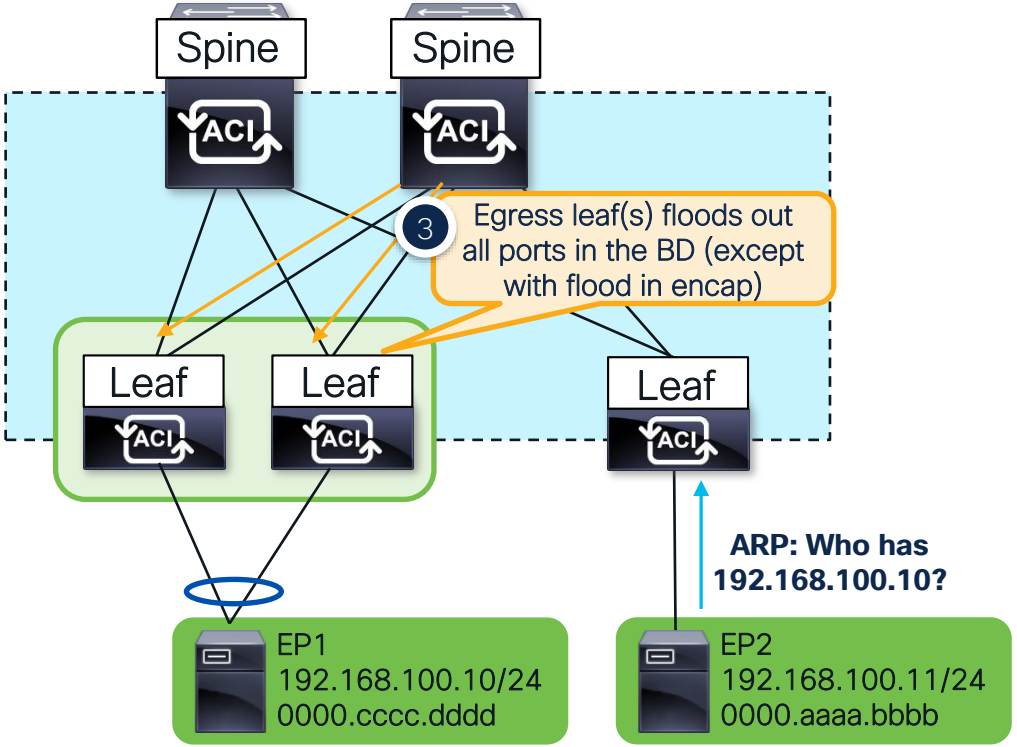



```
leaf102# show vlan | grep CL2022:bd1
 70   CL2022:bd1                          active

leaf102# show vlan id 70 extended

VLAN Name            Encap              Ports
---- ------------    ----------------   ----------------------
70   CL2022:bd1      vxlan-14811121     Eth1/33, Eth1/34,
                                        Eth1/43, Eth1/48, Po5,
                                        Po6, Po8, Po9
```

Flood out these ports and encaps in this BD

ARP – Egress Leaf

Bridge Domain Settings:
Unicast Routing Disable
ARP Flooding Enabled



ELAM the ARP request!

```
vsh_lc
debug plat internal tah elam asic 0
 trigger reset
 trigger init in-select 14 out-select 1
  set inner arp source-ip 192.168.100.11
  set inner arp target-ip 192.168.100.10
  set inner l2 dst_mac ffff.ffff.ffff
  start

stat
 ELAM STATUS
===========
Asic 0 Slice 0 Status Triggered
Asic 0 Slice 1 Status Armed
```

ARP – Egress Leaf Elam Results (ereport)

Bridge Domain Settings:
Unicast Routing Disable
ARP Flooding Enabled

```
Outer L3 Header
-------------------------------
Destination IP     : 225.0.2.136

Inner L3 Header
----------------------------------
ARP Sender IP      : 192.168.100.11
ARP Target IP      : 192.168.100.10

Outer L4 Header
----------------------------------------
VRF or BD VNID     : 14811121( 0xE1FFF1 )

Contract Result
----------------------
Contract Drop      : no

FINAL FORWARDING LOOKUP
-------------------------------------------------------------------------------------
Bits set in Final Forwarding Block:  : IFABRIC_EG MC INFRA ENCAP MYTEP BRIDGE MISS FLOOD

Lookup Drop
---------------------------
LU drop reason     : no drop
```

Destination is GIPO (225.0.2.128) + FTAG (0x8)

Frame is flooded in the Bridge Domain!

Not Dropped in lookups!

ARP – Egress Leaf Port is VPC

Bridge Domain Settings:
Unicast Routing Disable
ARP Flooding Enabled

- Both VPC members receive a flooded copy
- One VPC member is the Designated Forwarder (DF) for the flow
- DF is hashed per flow
- **Only DF floods out VPC interfaces**

Non-DF Leaf

```
module-1(DBG-elam-insel14)# ereport | grep df | grep vpc
  sug_lub_latch_results_vec.lub4_1.vpc_df: 0x0
        sug_fpx_lookup_vec.lkup.dciptvec.pt.vpc_df: 0x0
          sug_fpc_lookup_vec.fplu_vec.lkup.dciptvec.pt.vpc_df: 0x0
          sug_fpc_lookup_vec.fplu_vec.lkup.dciptvec.pt.vpc_df: 0x0
```

DF Leaf

```
module-1(DBG-elam-insel14)# ereport | grep df | grep vpc
  sug_lub_latch_results_vec.lub4_1.vpc_df: 0x1
        sug_fpx_lookup_vec.lkup.dciptvec.pt.vpc_df: 0x1
          sug_fpc_lookup_vec.fplu_vec.lkup.dciptvec.pt.vpc_df: 0x1
          sug_fpc_lookup_vec.fplu_vec.lkup.dciptvec.pt.vpc_df: 0x1
```

Debugging ACI Bridged Flows

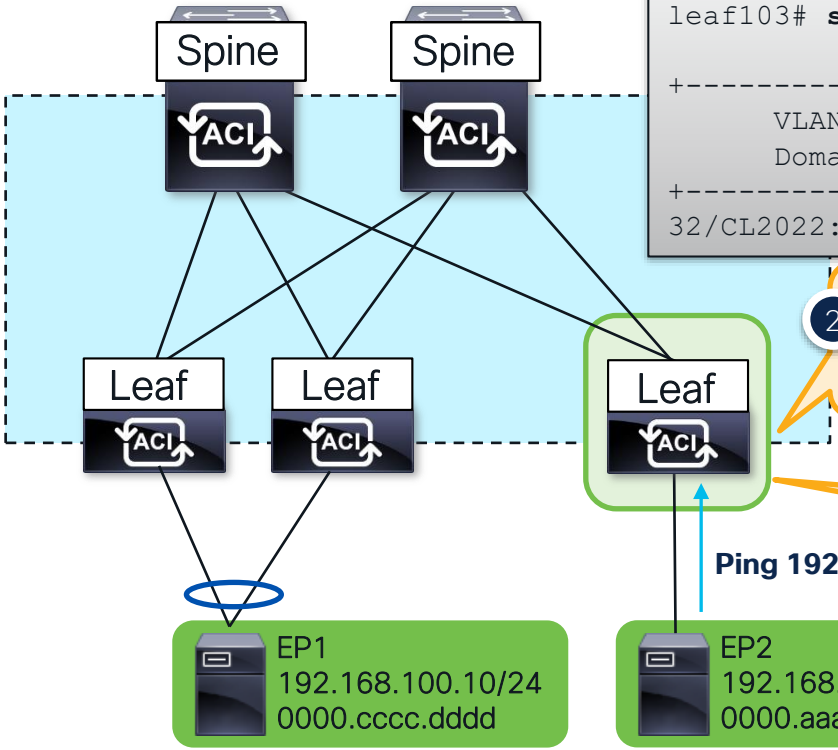
Known Unicast – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Disable
Unknown Unicast Flood

Lookup dst mac in ingress BD

```
leaf103# show endpoint mac 0000.cccc.dddd

+---------------+---------------+-----------------+-------------+
      VLAN/         Encap            MAC Address        Interface
      Domain        VLAN             IP Address
+---------------+---------------+-----------------+-------------+
32/CL2022:vrf1    vxlan-14811121    0000.cccc.dddd        tunnel1
```

```
leaf103# show int tun1
Tunnel destination 10.0.176.67
```

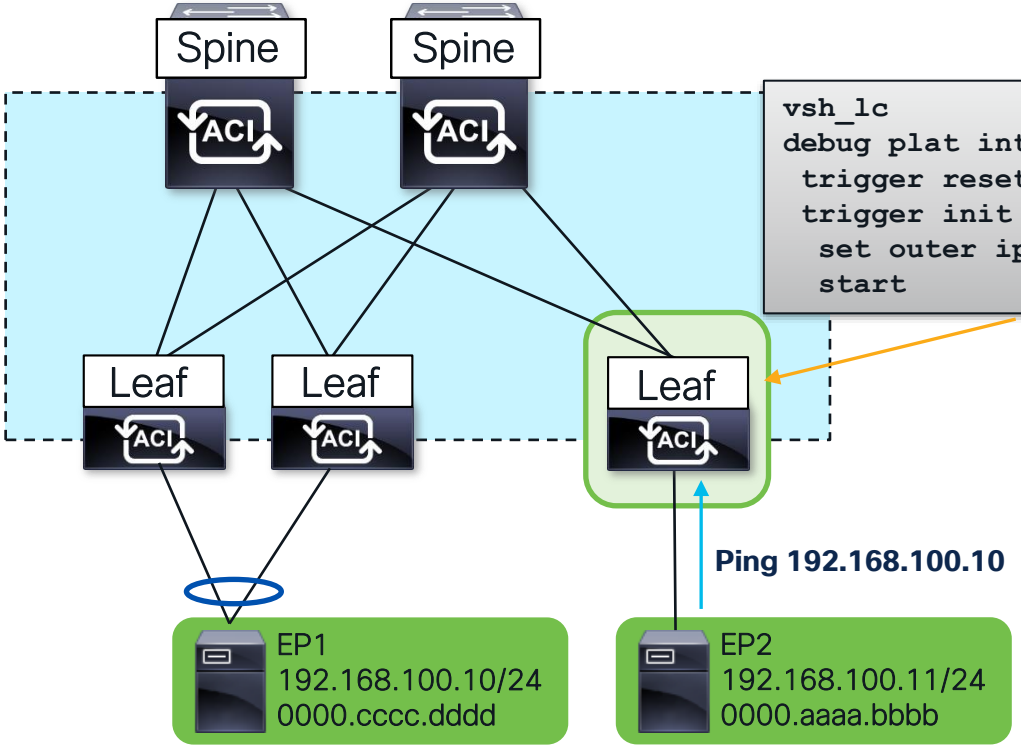


2 Since dst mac is not the router (GW) mac, leaf does mac lookup in BD (usually determined by ingress vlan)

1 Leaf looks at the dst mac to determine if it should route or switch

Known Unicast – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Disable
Unknown Unicast Flood



ELAM

```
vsh_lc
debug plat internal app elam asic 0
 trigger reset
 trigger init in-select 6 out-select 0
  set outer ipv4 src_ip 192.168.100.11 dst_ip 192.168.100.10
  start
```

Known Unicast – Ingress Leaf

Forwarding Verifications

Bridge Domain Settings:
Unicast Routing Disable
Unknown Unicast Flood

```
Outer L2 Header
-----------------------------------
Destination MAC     : 0000.cccc.dddd
Source MAC          : 0000.aaaa.bbbb
Access Encap VLAN   : 3502( 0xDAE )

Outer L3 Header
-----------------------------------
IP Protocol Number : ICMP
Destination IP     : 192.168.100.10
Source IP          : 192.168.100.11

Other Forwarding Information
-------------------------------------
Encap Index is valid       : yes
Encap Index                : 34( 0x22 )

FINAL FORWARDING LOOKUP
-----------------------------------------------------------------
Bits set in Final Forwarding Block: IFABRIC_IG UC TENANT MYTEP BRIDGE HIT

Lookup Drop
-------------------------
LU drop reason    : no drop
```

Dest mac that is looked up within BD

Make sure this is the expected vlan

Dest is tunnel

```
show plat internal hal tunnel rtep apd
=================================
ifId      IP            RwEncapIdx
=================================
18010001  10.0.176.67   22
```

Forward to this overlay TEP

Not Dropped in lookups!

Unicast + Bridge (L2 lookup) + Destination Known

Known Unicast – Ingress Leaf

Forwarding Verifications

Bridge Domain Settings:
Unicast Routing Disable
Unknown Unicast Flood

```
ereport | grep "ovector "
ovector    : 152( 0x98 )
```

```
show platform internal hal l2 port gpd

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

IfId      Ifname      As AP  Sl Sp Ss Ovec
==========================================
1a01c000 Eth1/29      0  59  2  18 18 98
```

Traffic is forwarded out Eth1/29!

Known Unicast – Ingress Leaf

Contract Verification

Bridge Domain Settings:
Unicast Routing Disable
Unknown Unicast Flood

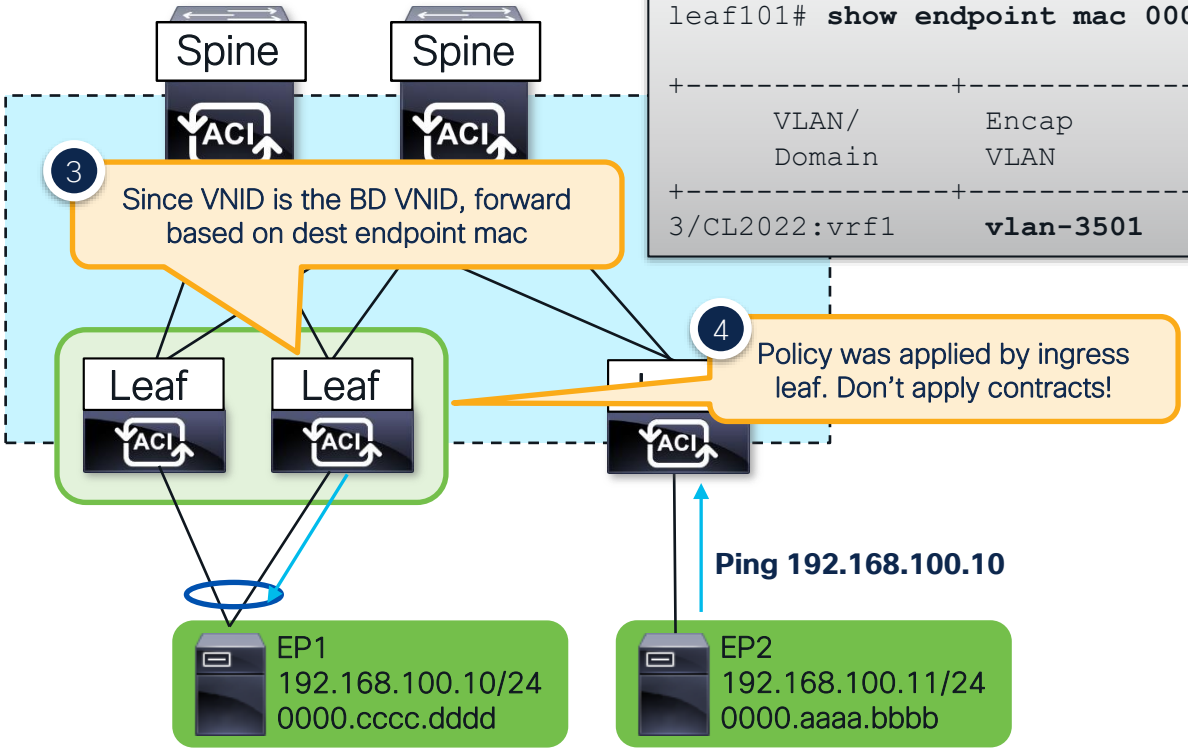
```
Contract Lookup Key
-----------------------------------------------
IP Protocol                     : ICMP( 0x1 )
L4 Src Port                     : 2048( 0x800 )
L4 Dst Port                     : 35914( 0x8C4A )
sclass (src pcTag)              : 49154( 0xC002 )
dclass (dst pcTag)              : 49154( 0xC002 )
src pcTag is from local table   : yes
Unknown Unicast / Flood Packet  : no

Contract Result
-----------------------------------------------
Contract Drop                                : no
Contract Applied                             : yes
Contract Hit                                 : yes
Contract Aclqos Stats Index                  : 131025
( show sys int aclqos zoning-rules | grep -B 9 "Idx: 131025" )
```

Source and Dest EPG is the same. Implicitly permit! (unless isolation enabled)

Contract Applied and no Drop!

Known Unicast – Egress Leaf

Bridge Domain Settings:
Unicast Routing Disable
Unknown Unicast Flood

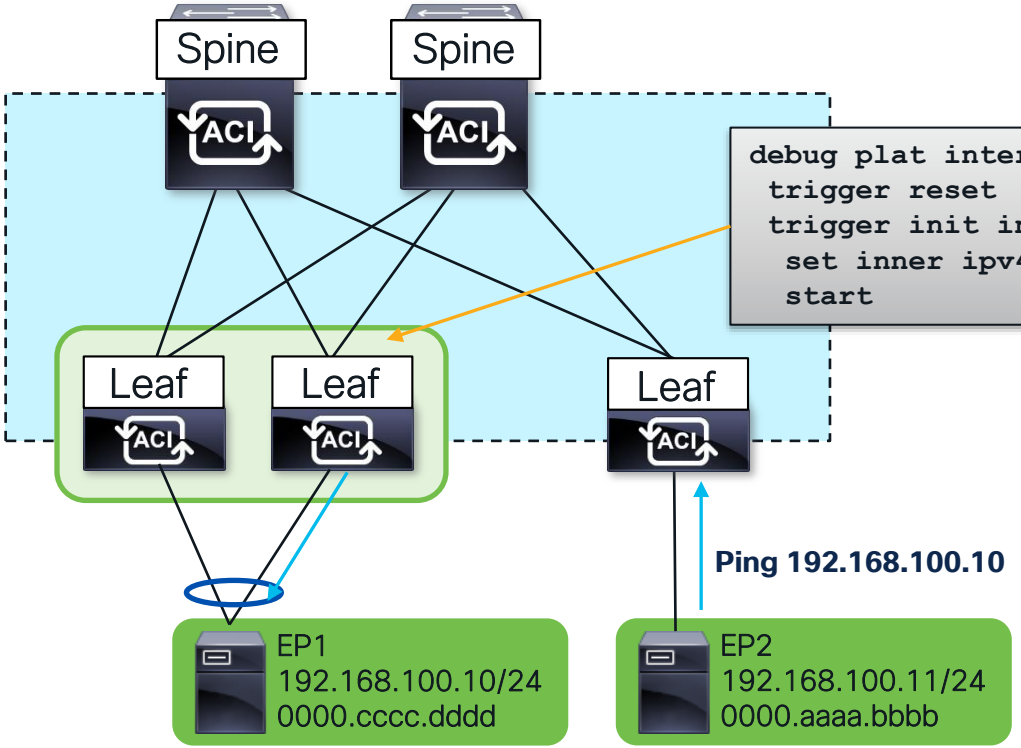



```
leaf101# show endpoint mac 0000.cccc.dddd

+---------------+---------------+-----------------+-------------+
      VLAN/        Encap           MAC Address        Interface
      Domain       VLAN            IP Address
+---------------+---------------+-----------------+-------------+
3/CL2022:vrf1      vlan-3501       0000.cccc.dddd            po5
```

Forward out portchannel 5 in vlan 3501!

Known Unicast – Egress Leaf

Bridge Domain Settings:
Unicast Routing Disable
Unknown Unicast Flood



Elam

```
debug plat internal tah elam asic 0
 trigger reset
 trigger init in-select 14 out-select 0
  set inner ipv4 src_ip 192.168.100.11 dst_ip 192.168.100.10
  start
```

Known Unicast – Egress Leaf

Bridge Domain Settings:
Unicast Routing Disable
Unknown Unicast Flood

```
Inner L2 Header
-----------------------------------
Inner Destination MAC  : 0000.cccc.

Inner L3 Header
-----------------------------------
Destination IP         : 192.168.100

Outer L4 Header
-----------------------------------
L4 Type                : iVxLAN
Src Policy Applied Bit : 1
Dst Policy Applied Bit : 1
VRF or BD VNID         : 14811121( 0xE1FFF1 )

Sideband Information
----------------------------------
ovector                : 146( 0x92 )

FINAL FORWARDING LOOKUP
-----------------------------------------------------------------------
Bits set in Final Forwarding Block: IFABRIC_EG UC INFRA ENCAP MYTEP BRIDGE HIT

Lookup Drop
------------------------
LU drop reason    : no drop
```

Contracts have already been applied. No need to check.

Mac lookup done in bridge domain with this VNID

```
show platform internal hal l2 port gpd
==========================================
IfId      Ifname      As AP  Sl Sp Ss Ovec
==========================================
1a021000  Eth1/34     0  32  1  9  12 92
```

Forward out Eth1/34!

Unicast + Bridge (L2 lookup) + Destination Known

Debugging ACI Routed Flows

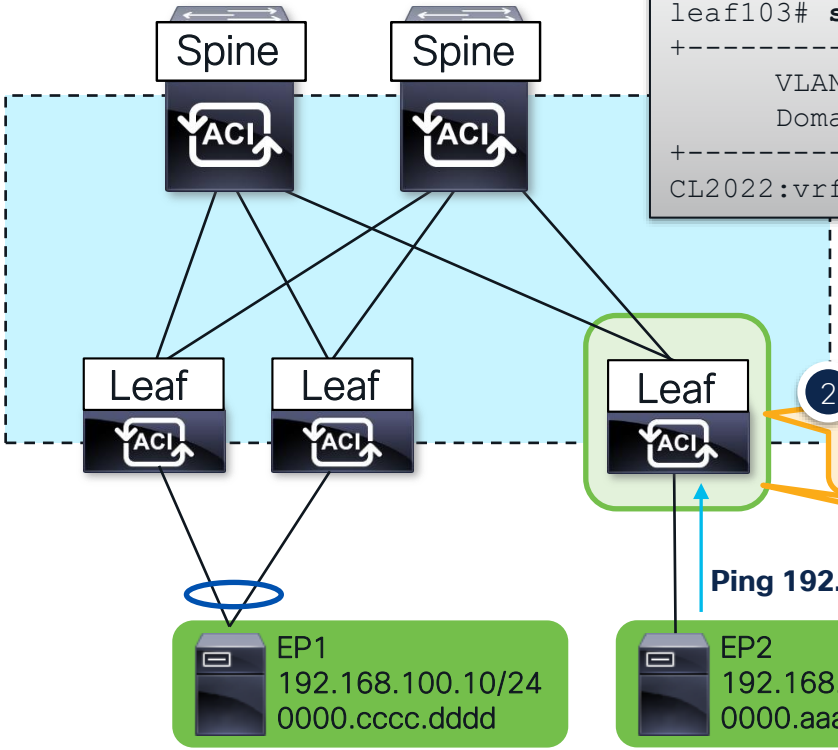
Known Unicast – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Enabled

Lookup dst IP in ingress VRF

```
leaf103# show endpoint ip 192.168.100.10
+---------------+-----------------+-------------+
      VLAN/        MAC Address        Interface
      Domain       IP Address
+---------------+-----------------+-------------+
CL2022:vrf1        192.168.100.10      tunnel1
```

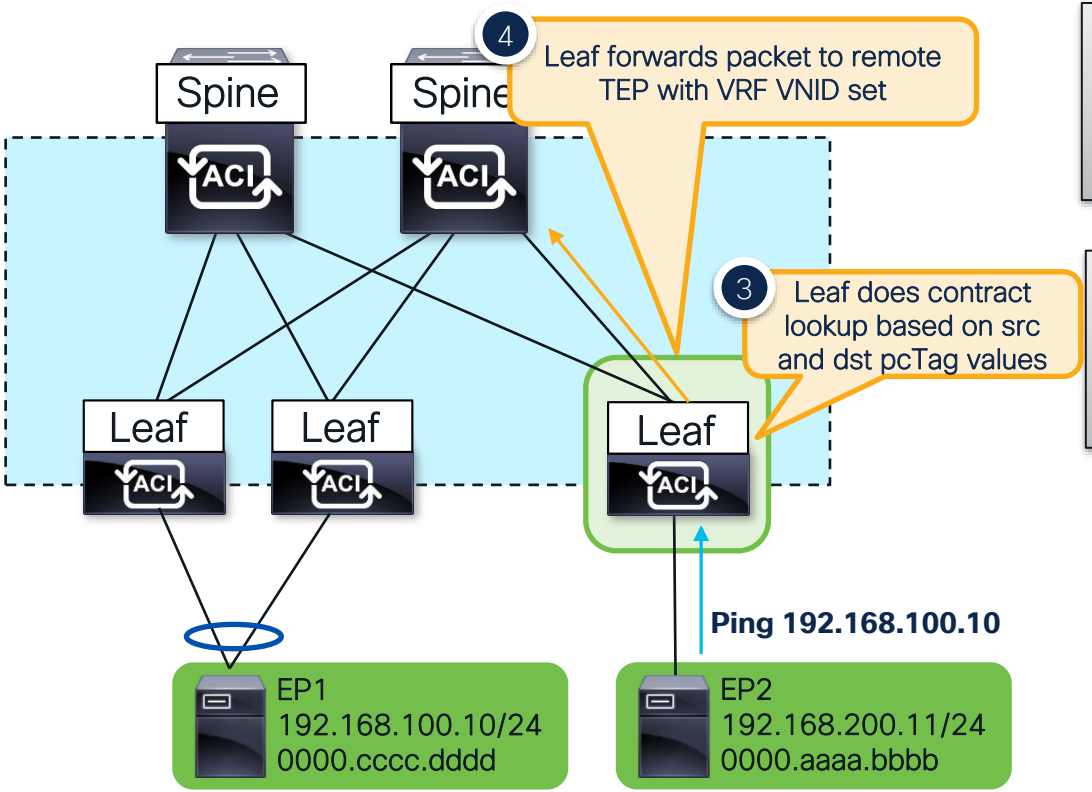
```
leaf103# show int tun1
Tunnel destination 10.0.176.67
```



2 Since dst mac is the router (GW) mac, leaf does IP lookup in VRF of source IP

1 Leaf looks at the dst mac to determine if it should route or switch

Known Unicast – Ingress Leaf



Bridge Domain Settings: Unicast Routing Enabled

Get Sclass

```
103# show sys internal epm endpoint ip
192.168.200.11
!omitted
BD vnid : 16613259 ::: VRF vnid : 2523136
sclass : 32771
```

Get Dclass

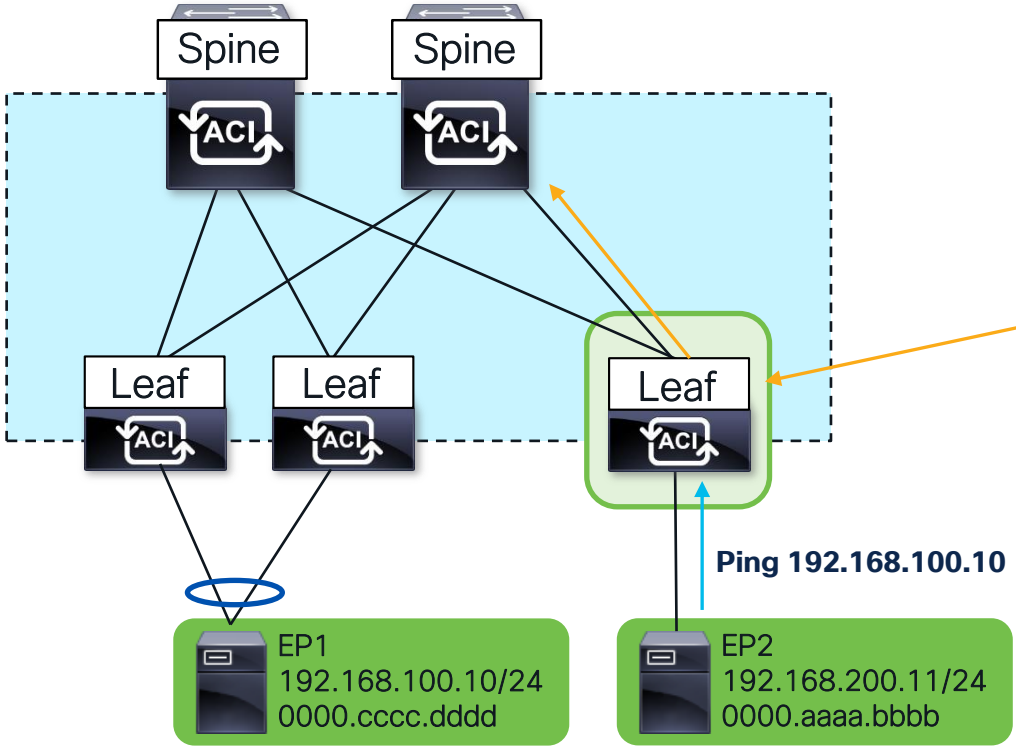
```
103# show sys internal epm endpoint ip
192.168.100.10
!omitted
BD vnid : 0 ::: VRF vnid : 2523136
sclass : 49154
```

Check Contract

```
103# show zoning-rule src-epg 32771
dst-epg 49154 scope 2523136
+---------+------------------+--------+
| RuleID  |       Name       | Action |
+---------+------------------+--------+
|   4209  | CL2022:allow-all | permit |
+---------+------------------+--------+
```

Known Unicast – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Enabled



ELAM

```
vsh_lc
debug plat internal app elam asic 0
 trigger reset
 trigger init in-select 6 out-select 0
  set outer ipv4 src_ip 192.168.200.11
  set outer ipv4 dst_ip 192.168.100.10
  start
stat
 ELAM STATUS
===========
Asic 0 Slice 0 Status Triggered
Asic 0 Slice 1 Status Armed
```

Known Unicast – Ingress Leaf

Forwarding Verifications

Bridge Domain Settings: Unicast Routing Enabled

```
Outer L2 Header
-----------------------------------
Destination MAC     : 0022.BDF8.19FF
Access Encap VLAN   : 3769( 0xEB9 )

Outer L3 Header
-----------------------------------
Destination IP                : 192.168.100.10
Source IP                     : 192.168.200.11

Other Forwarding Information
-----------------------------------
Encap Index is valid        : yes
Encap Index                 : 34( 0x22 )

FINAL FORWARDING LOOKUP
-----------------------------------------------------------------------
Bits set in Final Forwarding Block: IFABRIC_IG UC TENANT MYTEP ROUTE HIT

Lookup Drop
-------------------------
LU drop reason    : no drop
```

ACI Router Mac. Route this packet!

Make sure this is the expected vlan

Dest is tunnel

```
show plat internal hal tunnel rtep apd
=================================
ifId      IP            RwEncapIdx
=================================
18010001  10.0.176.67   22
```

Forward to this overlay TEP

Unicast + Route (L3 lookup) + L3 Route Found

Not Dropped in lookups!

Known Unicast – Ingress Leaf

Forwarding Verifications

Bridge Domain Settings:
Unicast Routing Enabled

```
ereport | grep "ovector "
ovector    : 152( 0x98 )
```

```
show platform internal hal l2 port gpd

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

IfId      Ifname      As AP  Sl Sp Ss Ovec
=========================================
1a01c000 Eth1/29      0  59  2  18 18 98
```

Traffic is forwarded out Eth1/29!

Known Unicast – Ingress Leaf

Contract Verification

Bridge Domain Settings: Unicast Routing Enabled

```
Contract Lookup Key
-----------------------------------------------
IP Protocol                     : ICMP( 0x1 )
L4 Src Port                     : 2048( 0x800 )
L4 Dst Port                     : 31219( 0x79F3 )
sclass (src pcTag)              : 32771( 0x8003 )
dclass (dst pcTag)              : 49154( 0xC002 )
src pcTag is from local table   : yes
Unknown Unicast / Flood Packet  : no

Contract Result
-----------------------------------------------
Contract Drop                                : no
Contract Applied                             : yes
Contract Hit                                 : yes
Contract Aclqos Stats Index                  : 131025
```

Source and Dest EPG used for contract lookup

Contract Applied and no Drop!

But how do I know which contract this is actually hitting?

Known Unicast – Ingress Leaf

Contract Verification

Bridge Domain Settings: Unicast Routing Enabled

```
Contract Result
--------------------------------
Contract Drop                  : no
Contract Applied               : yes
Contract Hit                   : yes
Contract Aclqos Stats Index    : 131025
```

Hardware Index of matching contract

Run this from vsh_lc

```
show sys int aclqos zoning-rules | grep -B 9 "Idx: 130974"
==========================================
Rule ID: 4163 Scope 8 Src EPG: 32771 Dst EPG: 49154 Filter 532
Curr TCAM resource:
  ============================
    === SDK Info ===
        Result/Stats Idx: 130974
```

Zoning-rule ID

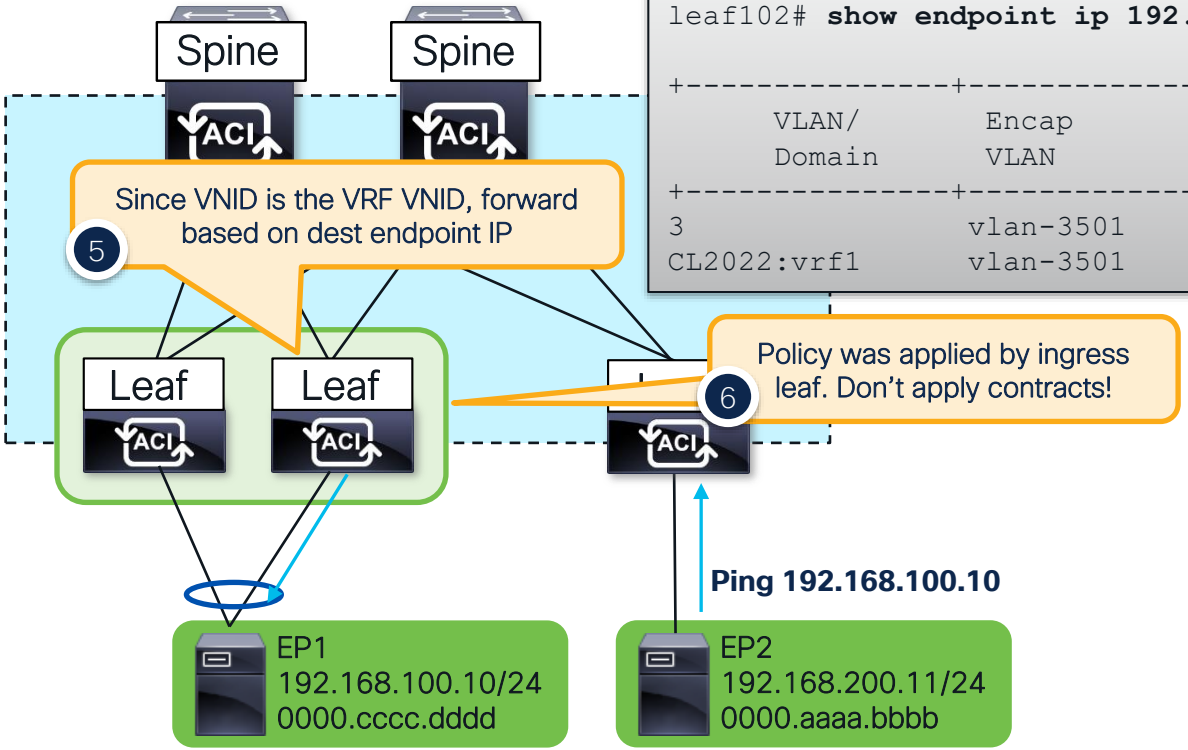
Run this from normal shell

```
show zoning-rule rule-id 4163
+---------+--------+--------+----------+---------+------------------+--------+
| Rule ID | SrcEPG | DstEPG | FilterID |  Scope  |       Name       | Action |
+---------+--------+--------+----------+---------+------------------+--------+
|   4163  | 32771  | 49154  |   532    | 2523136 | CL2022:allow-all | permit |
+---------+--------+--------+----------+---------+------------------+--------+
```

Traffic hit this contract!

Known Unicast – Egress Leaf

Bridge Domain Settings:
Unicast Routing Enabled

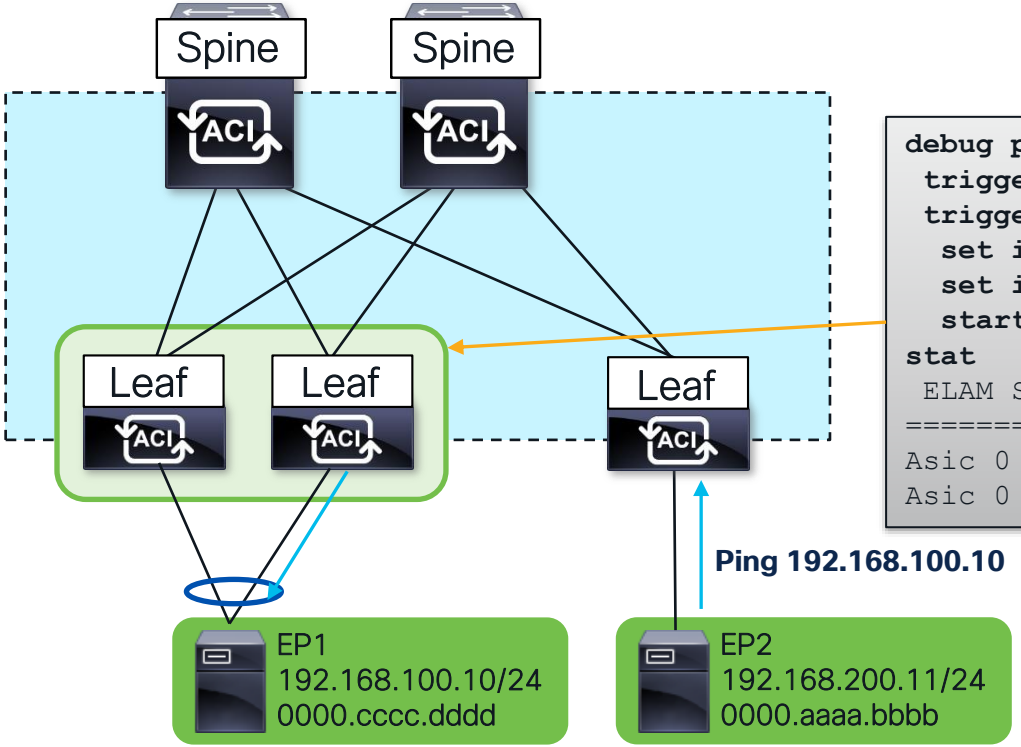



```
leaf102# show endpoint ip 192.168.100.10

+---------------+---------------+-----------------+-------------+
      VLAN/        Encap            MAC Address        Interface
      Domain       VLAN             IP Address
+---------------+---------------+-----------------+-------------+
3                 vlan-3501       0000.cccc.dddd     po5
CL2022:vrf1       vlan-3501       192.168.100.10     po5
```

Forward out portchannel 5 in vlan 3501!

Known Unicast – Egress Leaf

Bridge Domain Settings:
Unicast Routing Enabled



Elam

```
debug plat internal tah elam asic 0
 trigger reset
 trigger init in-select 14 out-select 0
  set inner ipv4 src_ip 192.168.200.11
  set inner ipv4 dst_ip 192.168.100.10
  start
stat
 ELAM STATUS
===========
Asic 0 Slice 0 Status Triggered
Asic 0 Slice 1 Status Armed
```

Known Unicast – Egress Leaf

Bridge Domain Settings: Unicast Routing Enabled

```
Inner L2 Header
-----------------------------------
Inner Destination MAC  : 000C.0C0C

Inner L3 Header
---------------------------------------
Destination IP         : 192.168.100.

Outer L4 Header
-----------------------------------------
L4 Type                : iVxLA
Src Policy Applied Bit : 1
Dst Policy Applied Bit : 1
VRF or BD VNID         : 2523136( 0x268000 )

Sideband Information
-----------------------------------
ovector                : 146( 0x92 )

FINAL FORWARDING LOOKUP
-------------------------------------------------------------------------
Bits set in Final Forwarding Block: IFABRIC_EG UC INFRA ENCAP MYTEP ROUTE HIT

Lookup Drop
------------------------
LU drop reason    : no drop
```

Contracts have already been applied. No need to check.

IP lookup done in VRF with this VNID

```
show platform internal hal l2 port gpd
=========================================
IfId      Ifname      As AP  Sl Sp Ss Ovec
=========================================
1a021000  Eth1/34     0  32  1  9  12 92
```

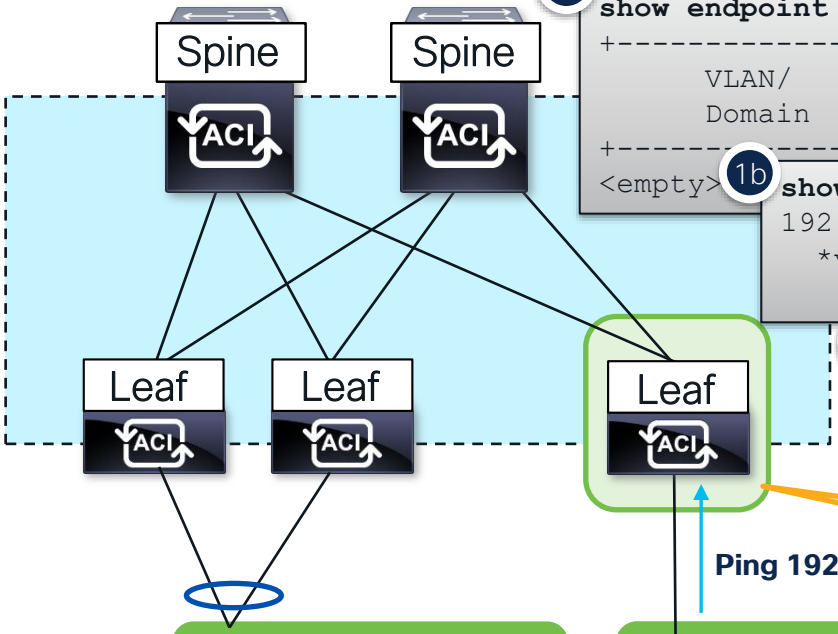
Forward out Eth1/34!

Unicast + Route (L3 lookup) + L3 Route Found

Proxied Unicast – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Enabled

Lookup dst IP in ingress VRF



1a

```
show endpoint ip 192.168.100.10
+---------------+-----------------+-------------+
      VLAN/        MAC Address        Interface
      Domain       IP Address
+---------------+-----------------+-------------+
<empty>
```

No endpoint learn, check route table!

1b

```
show ip route 192.168.100.10 vrf CL2022:vrf1
192.168.100.0/24, attached, direct, pervasive
  *via 10.0.176.66%overlay-1, [1/0], static
     recursive next hop: 10.0.176.66/32%overlay-1
```

1c

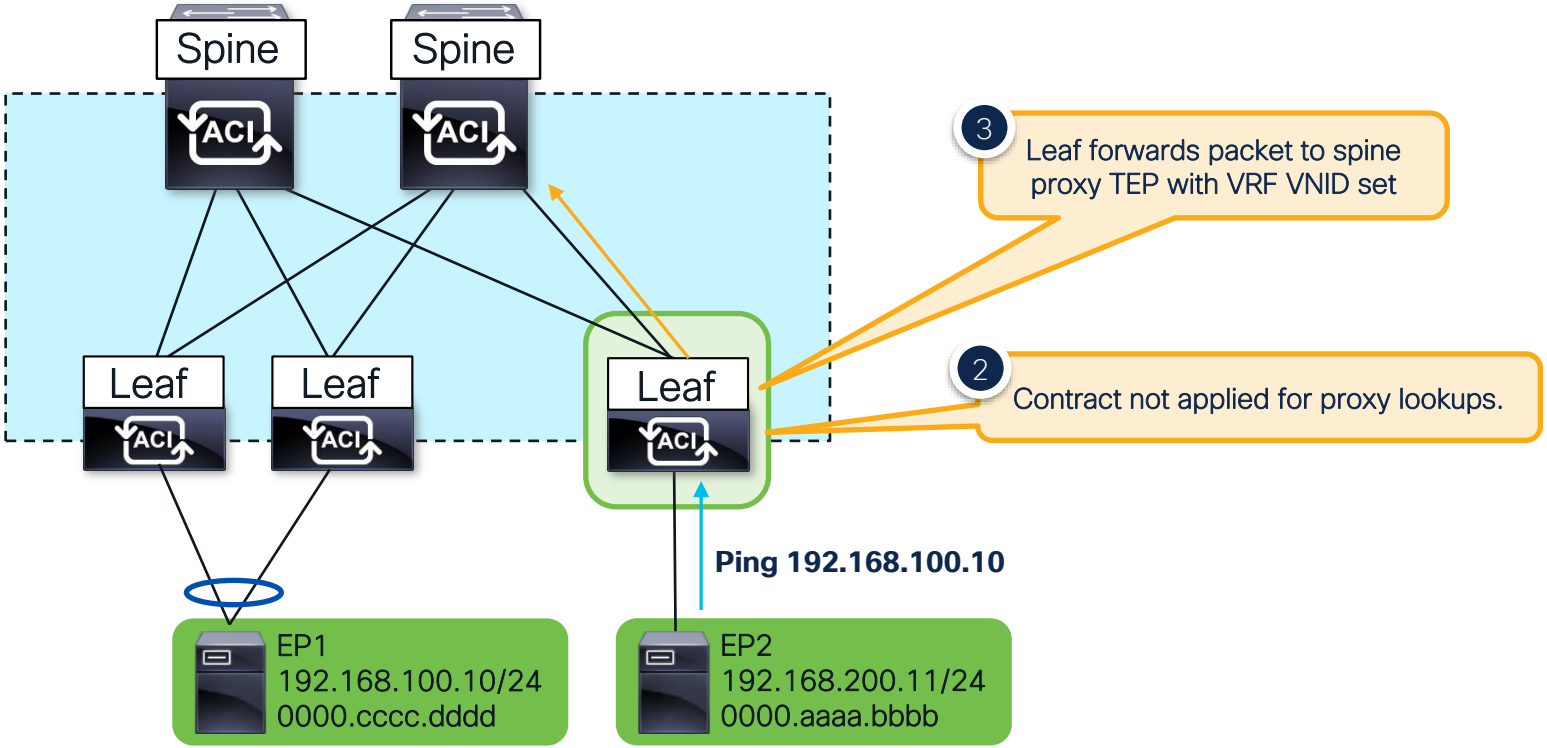
```
show isis dtep vrf overlay-1
DTEP-Address Role   Type
10.0.176.66  SPINE  PHYSICAL,PROXY-ACAST-V4
```

Send to spine proxy!

1 Dst mac is router mac so Route!
Lookup dest IP in vrf of source IP.

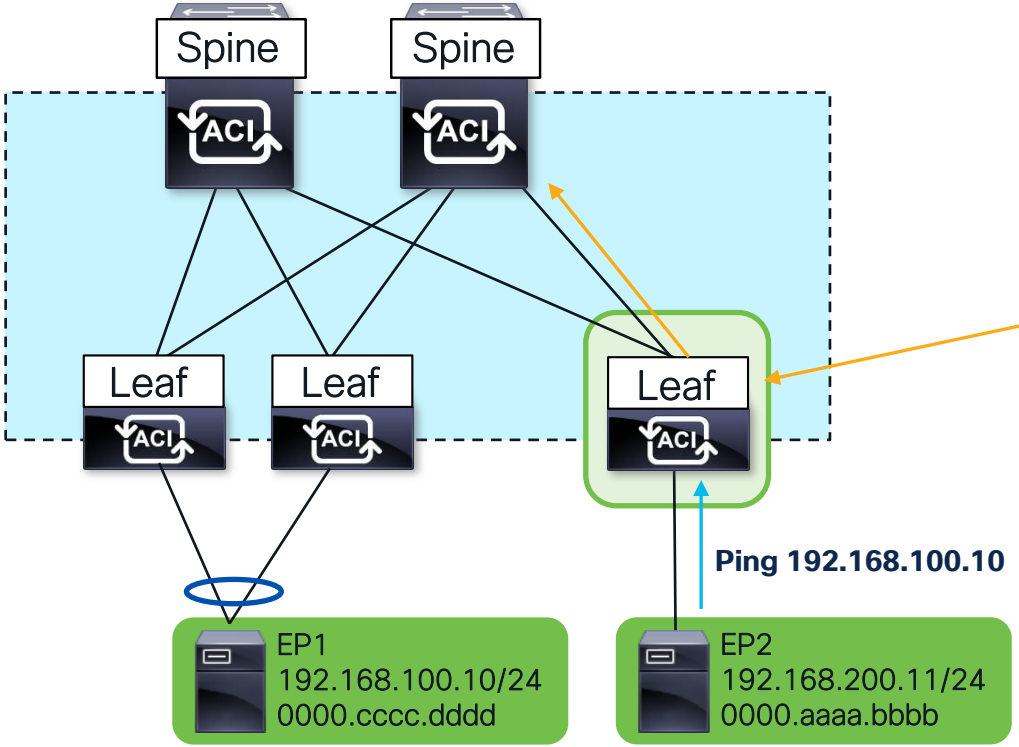
Proxied Unicast – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Enabled



Proxied Unicast – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Enabled



ELAM

```
vsh_lc
debug plat internal app elam asic 0
 trigger reset
 trigger init in-select 6 out-select 0
  set outer ipv4 src_ip 192.168.200.11
  set outer ipv4 dst_ip 192.168.100.10
  start
stat
 ELAM STATUS
===========
Asic 0 Slice 0 Status Triggered
Asic 0 Slice 1 Status Armed
```

Proxied Unicast – Ingress Leaf

Forwarding Verifications

Bridge Domain Settings: Unicast Routing Enabled

```
Outer L2 Header
-----------------------------------
Destination MAC     : 0022.BDF8.19FF
Access Encap VLAN   : 3769( 0xEB9 )

Outer L3 Header
-----------------------------------
Destination IP                : 192.168.100.10
Source IP                     : 192.168.200.11

Other Forwarding Information
-----------------------------------
Encap Index is valid       : yes
Encap Index                : 1( 0x1 )

FINAL FORWARDING LOOKUP
-----------------------------------------------------------------------
Bits set in Final Forwarding Block: IFABRIC_IG UC TENANT MYTEP ROUTE HIT

Lookup Drop
-------------------------
LU drop reason    : no drop
```

ACI Router Mac. Route this packet!

Make sure this is the expected vlan

Dest is tunnel

```
show plat internal hal tunnel rtep apd
=================================
ifId      IP            RwEncapIdx
=================================
18010007  10.0.176.66   1
```

Forward to this overlay TEP

Unicast + Route (L3 lookup) + L3 Route Found

Not Dropped in lookups!

Proxied Unicast – Ingress Leaf

Forwarding Verifications

Bridge Domain Settings:
Unicast Routing Enabled

```
ereport | grep "ovector "
ovector    : 152( 0x98 )
```

```
show platform internal hal l2 port gpd

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

IfId      Ifname      As AP  Sl Sp Ss Ovec
==========================================
1a01c000 Eth1/29      0  59  2  18 18 98
```

Traffic is forwarded out Eth1/29!

Proxied Unicast – Ingress Leaf

Contract Verification

Bridge Domain Settings: Unicast Routing Enabled

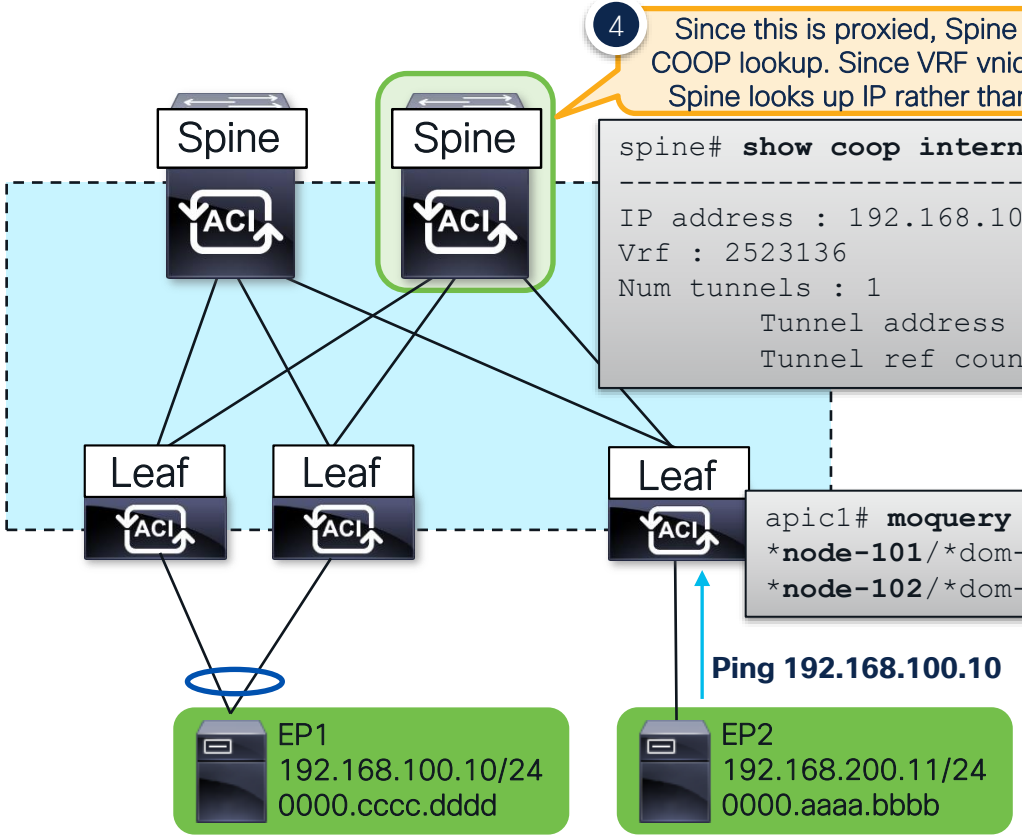
```
Contract Lookup Key
-----------------------------------------------
IP Protocol                    : ICMP( 0x1 )
L4 Src Port                    : 2048( 0x800 )
L4 Dst Port                    : 31219( 0x79F3 )
sclass (src pcTag)             : 32771( 0x8003 )
dclass (dst pcTag)             : 1( 0x1 )
src pcTag is from local table  : yes
Unknown Unicast / Flood Packet : no

Contract Result
---------------------------------------------
Contract Drop                                : no
Contract Applied                             : no
Contract Hit                                 : yes
Contract Aclqos Stats Index                  : 131025
```

Dest EPG is 1 for fabric owned subnets

Contract not applied since this is proxied!

Proxied Unicast – Spine

Bridge Domain Settings:
Unicast Routing Enabled



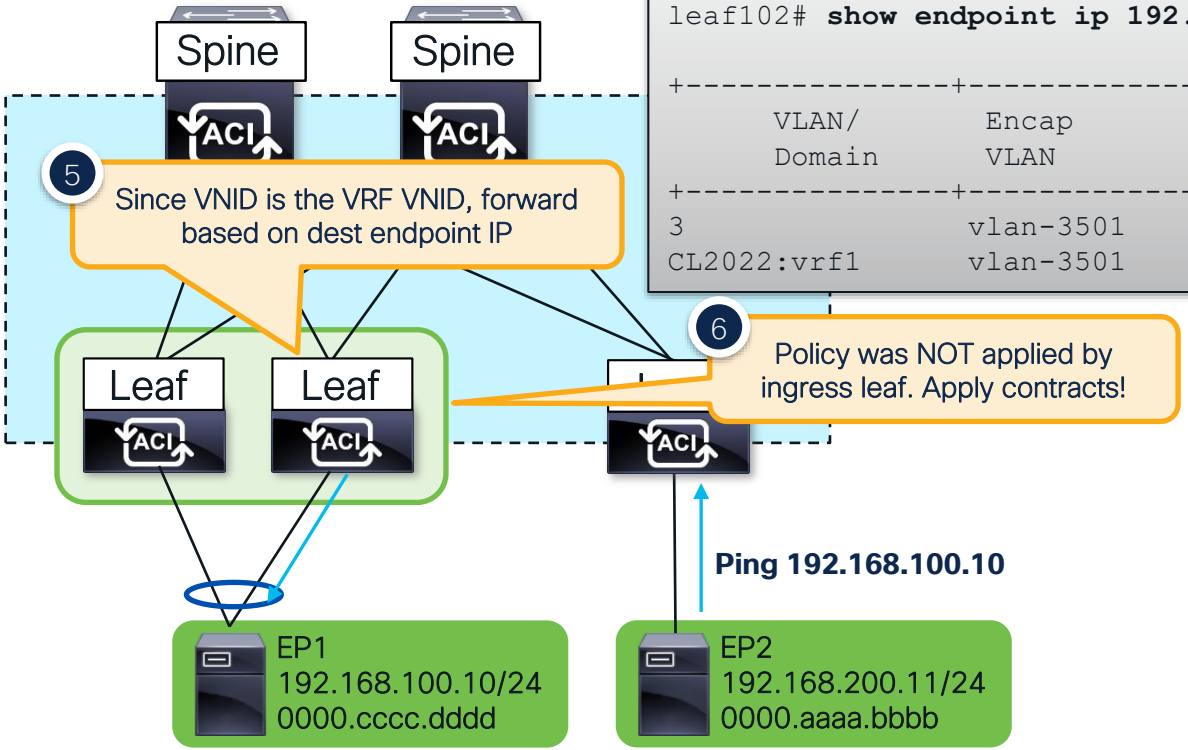

```
spine# show coop internal info ip-db | grep -B 1 -A 15 192.168.100.10
------------------------------
IP address : 192.168.100.10
Vrf : 2523136
Num tunnels : 1
        Tunnel address : 10.0.176.67
        Tunnel ref count : 1
```

Dest TEP of Leaf(s) that own this Endpoint

```
apic1# moquery -c ipv4Addr -f 'ipv4.Addr.addr=="10.0.176.67"'
*node-101/*dom-overlay-1/if-[lo1]/addr-[10.0.176.67/32]
*node-102/*dom-overlay-1/if-[lo1]/addr-[10.0.176.67/32]
```

Proxied Unicast – Egress Leaf

Bridge Domain Settings:
Unicast Routing Enabled

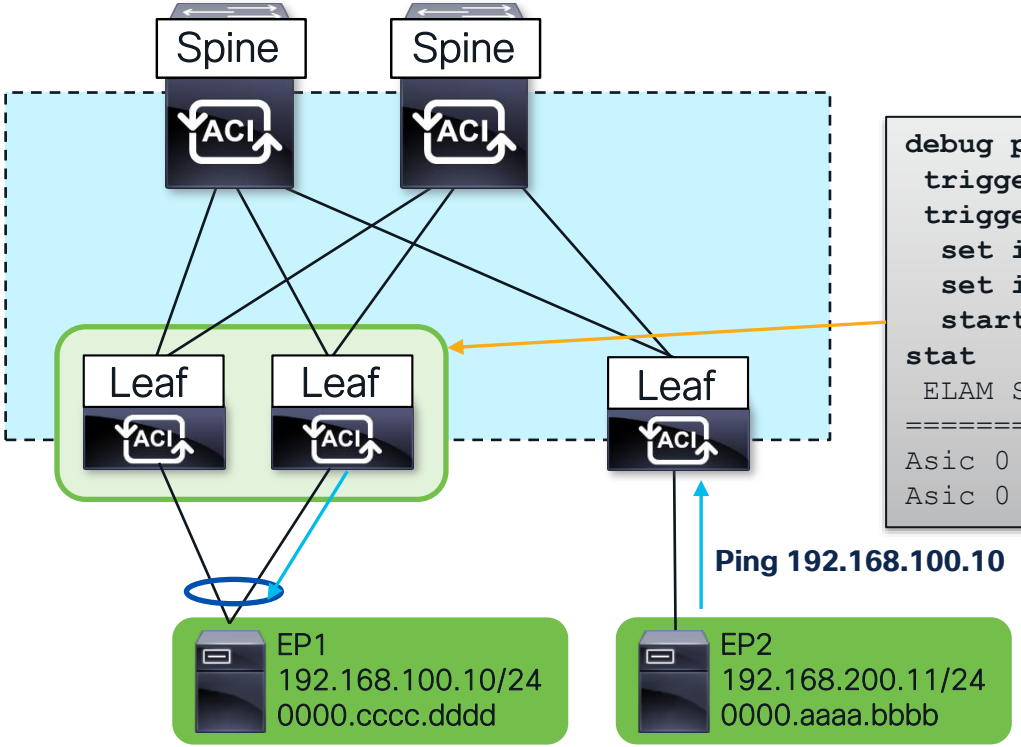



```
leaf102# show endpoint ip 192.168.100.10

+---------------+---------------+-----------------+-------------+
      VLAN/        Encap            MAC Address        Interface
      Domain       VLAN             IP Address
+---------------+---------------+-----------------+-------------+
3                 vlan-3501       0000.cccc.dddd     po5
CL2022:vrf1       vlan-3501       192.168.100.10     po5
```

Forward out portchannel 5 in vlan 3501!

Proxied Unicast – Egress Leaf

Bridge Domain Settings:
Unicast Routing Enabled



Elam

```
debug plat internal tah elam asic 0
 trigger reset
 trigger init in-select 14 out-select 0
  set inner ipv4 src_ip 192.168.200.11
  set inner ipv4 dst_ip 192.168.100.10
  start
stat
 ELAM STATUS
===========
Asic 0 Slice 0 Status Triggered
Asic 0 Slice 1 Status Armed
```

Proxied Unicast – Egress Leaf

Forwarding Verifications

Bridge Domain Settings: Unicast Routing Enabled

```
Inner L3 Header
-----------------------------------
Destination IP          : 192.168.100

Outer L4 Header
-----------------------------------
L4 Type                 : iVxLAN
Src Policy Applied Bit  : 0
Dst Policy Applied Bit  : 0
VRF or BD VNID          : 2523136( 0x268000 )

Sideband Information
---------------------------------
ovector                 : 146( 0x92 )

FINAL FORWARDING LOOKUP
-----------------------------------------------------------------------------
Bits set in Final Forwarding Block: IFABRIC_EG UC INFRA ENCAP MYTEP ROUTE HIT

Lookup Drop
------------------------
LU drop reason    : no drop
```

Contracts have not been applied yet!

IP lookup done in VRF with this VNID

```
show platform internal hal l2 port gpd
=========================================
IfId      Ifname      As AP  Sl Sp Ss Ovec
=========================================
1a021000  Eth1/34     0  32  1  9  12 92
```

Forward out Eth1/34!

Unicast + Route (L3 lookup) + L3 Route Found

Not Dropped in lookups!

Proxied Unicast – Egress Leaf

Contract Verification

Bridge Domain Settings:
Unicast Routing Enabled

```
Contract Lookup Key
-----------------------------------------------
IP Protocol                     : ICMP( 0x1 )
L4 Src Port                     : 2048( 0x800 )
L4 Dst Port                     : 33226( 0x81CA )
sclass (src pcTag)              : 32771( 0x8003 )
dclass (dst pcTag)              : 49154( 0xC002 )
src pcTag is from local table   : no
Unknown Unicast / Flood Packet  : no

Contract Result
-----------------------------------------------
Contract Drop                                : no
Contract Applied                             : yes
Contract Hit                                 : yes
Contract Aclqos Stats Index                  : 131025
```

Source and Dest EPG used for contract lookup.

Contract Applied and no Drop!

But how do I know which contract this is actually hitting?

Proxied Unicast – Egress Leaf

Contract Verification

Bridge Domain Settings: Unicast Routing Enabled

```
Contract Result
--------------------------------
Contract Drop                  : no
Contract Applied               : yes
Contract Hit                   : yes
Contract Aclqos Stats Index    : 81836
```

Hardware Index of matching contract

Run this from vsh_lc

```
show sys int aclqos zoning-rules | grep -B 9 "Idx: 81836"
==========================================
Rule ID: 4234 Scope 16 Src EPG: 32771 Dst EPG: 49154 Filter
532
============================
    === SDK Info ===
        Result/Stats Idx: 81836
```

Zoning-rule ID

Run this from normal shell

```
show zoning-rule rule-id 4234
+---------+--------+--------+----------+---------+------------------+--------+
| Rule ID | SrcEPG | DstEPG | FilterID |  Scope  |       Name       | Action |
+---------+--------+--------+----------+---------+------------------+--------+
|   4163  | 32771  | 49154  |   532    | 2523136 | CL2022:allow-all | permit |
+---------+--------+--------+----------+---------+------------------+--------+
```

Traffic hit this contract!

L3Out Destination – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Enabled

Lookup dst IP in ingress VRF

2a

```
show endpoint ip 10.99.99.100
+---------------+-----------------+-------------+
      VLAN/        MAC Address        Interface
      Domain       IP Address
+---------------+-----------------+-------------+
<empty>
```

No endpoint learn, check route table!

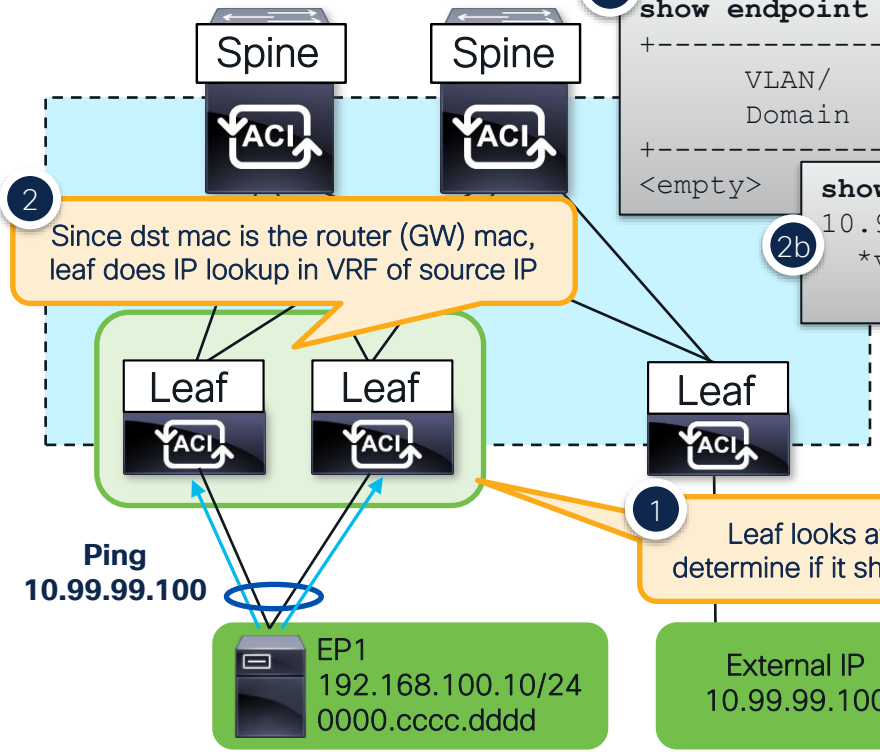
2b

```
show ip route 10.99.99.100 vrf CL2022:vrf1
10.99.99.0/24, ubest/mbest: 1/0
  *via 10.0.64.70%overlay-1, [200/20], bgp-65100
    recursive next hop: 10.0.64.70/32%overlay-1
```

Send to BL PTEP!

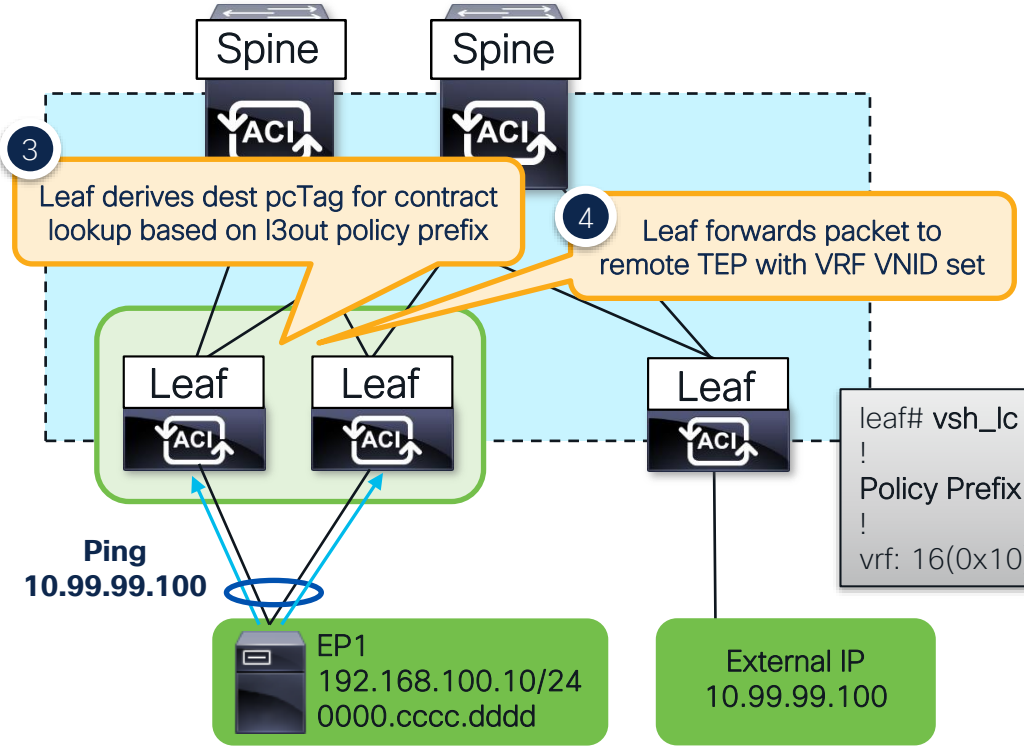
2c

```
acidiag fnvread | grep 10.0.64.70
Name        IP Address      Role
----------------------------------
leaf103     10.0.64.70/32   leaf
```



L3Out Destination – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Enabled



External EPGs

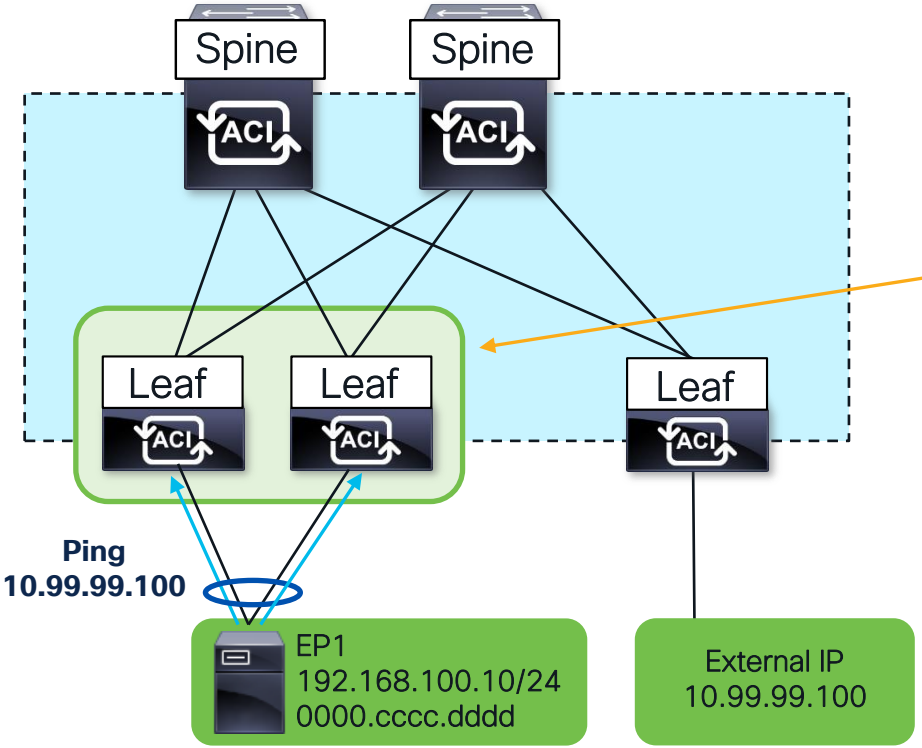
External EPGs

Name	Description	pcTag
all	10.99.99.0/24 Network	32772

```
leaf# vsh_lc -c "show forwarding route 10.99.99.100 platf vrf CL2022:vrf1"
!
Policy Prefix 10.99.99.0/24
!
vrf: 16(0x10), routed_if: 0x0 epc_class: 32772(0x8004)
```

L3Out Destination – Ingress Leaf

Bridge Domain Settings:
Unicast Routing Enabled



ELAM

```
vsh_lc
debug plat internal tah elam asic 0
 trigger reset
 trigger init in-select 6 out-select 0
  set outer ipv4 src_ip 192.168.100.10
  set outer ipv4 dst_ip 10.99.99.100
  start
stat
 ELAM STATUS
===========
Asic 0 Slice 0 Status Triggered
Asic 0 Slice 1 Status Armed
```

L3Out Destination – Ingress Leaf

Forwarding Verifications

Bridge Domain Settings: Unicast Routing Enabled

```
Outer L2 Header
-----------------------------------
Destination MAC     : 0022.BDF8.19FF
Access Encap VLAN   : 3501( 0xDAD )

Outer L3 Header
-----------------------------------
Destination IP                  : 10.99.99.100
Source IP                       : 192.168.100.10

Other Forwarding Information
-----------------------------------
Encap Index is valid        : yes
Encap Index                 : 37( 0x25 )

FINAL FORWARDING LOOKUP
-----------------------------------------------------------------------
Bits set in Final Forwarding Block: IFABRIC_IG UC TENANT MYTEP ROUTE HIT

Lookup Drop
-------------------------
LU drop reason    : no drop
```

ACI Router Mac. Route this packet!

Make sure this is the expected vlan

Dest is tunnel

```
show plat internal hal tunnel rtep apd
=================================
ifId      IP            RwEncapIdx
=================================
18010004  10.0.64.70    25
```

Forward to this overlay TEP

Unicast + Route (L3 lookup) + L3 Route Found

Not Dropped in lookups!

L3Out Destination – Ingress Leaf

Forwarding Verifications

Bridge Domain Settings:
Unicast Routing Enabled

```
ereport | grep "ovector "
ovector    : 48 ( 0x30 )
```

```
show platform internal hal l2 port gpd

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

IfId      Ifname      As AP  Sl Sp Ss Ovec
=========================================
1a035000  Eth1/54     0  19  0  18 30 30
```

Traffic is forwarded out Eth1/54!

L3Out Destination – Ingress Leaf

Contract Verification

Bridge Domain Settings: Unicast Routing Enabled

```
Contract Lookup Key
-----------------------------------------------
IP Protocol                     : ICMP( 0x1 )
L4 Src Port                     : 2048( 0x800 )
L4 Dst Port                     : 12063( 0x2F1F )
sclass (src pcTag)              : 49154( 0xC002 )
dclass (dst pcTag)              : 32772( 0x8004 )
src pcTag is from local table   : yes
Unknown Unicast / Flood Packet  : no

Contract Result
-----------------------------------------------
Contract Drop                                : no
Contract Applied                             : yes
Contract Hit                                 : yes
Contract Aclqos Stats Index                  : 81765
```

Source and Dest EPG used for contract lookup

Contract Applied and no Drop!

But how do I know which contract this is actually hitting?

L3Out Destination – Ingress Leaf

Contract Verification

Bridge Domain Settings: Unicast Routing Enabled

```
Contract Result
--------------------------------
Contract Drop                  : no
Contract Applied               : yes
Contract Hit                   : yes
Contract Aclqos Stats Index    : 81765
```

Hardware Index of matching contract

Run this from vsh_lc

```
show sys int aclqos zoning-rules | grep -B 9 "Idx: 81765"
==========================================
Rule ID: 4248 Scope 16 Src EPG: 0 Dst EPG: 32772 Filter 532
  Curr TCAM resource:
  ============================
    === SDK Info ===
        Result/Stats Idx: 81765
```

Zoning-rule ID

Run this from normal shell

```
show zoning-rule rule-id 4248
+---------+--------+--------+----------+---------+----------------------+---------+
| Rule ID | SrcEPG | DstEPG | FilterID |  Scope  |        Name          | Action  |
+---------+--------+--------+----------+---------+----------------------+---------+
|  4248   | 0      | 32772  |   532    | 2523136 | CL2022:l3out-allow-all | permit  |
+---------+--------+--------+----------+---------+----------------------+---------+
```

Traffic hit this contract!

L3Out Destination – Egress Leaf

Bridge Domain Settings:
Unicast Routing Enabled

Lookup dst IP in received VRF

5a

```
show endpoint ip 10.99.99.100
+---------------+-----------------+-------------+
      VLAN/        MAC Address        Interface
      Domain       IP Address
+---------------+-----------------+-------------+
<empty>
```

No endpoint learn, check route table!

5b

```
show ip route 10.99.99.100 vrf CL2022:vrf1
10.99.99.0/24, ubest/mbest: 1/0
  *via 10.55.0.100, vlan25, [110/20], ospf, type-2
```

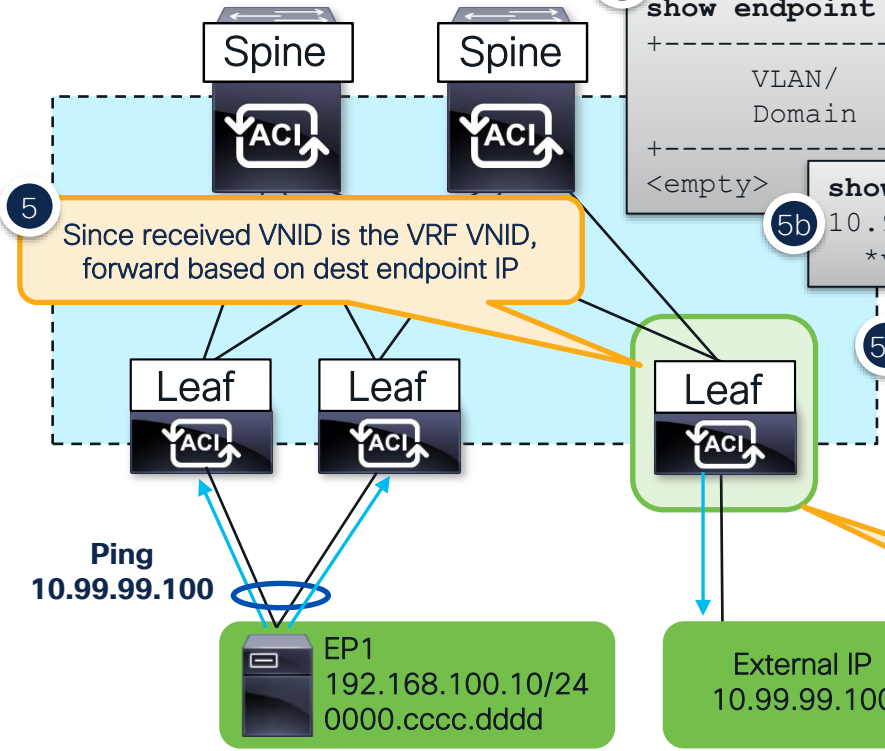
5c

```
show ip arp 10.55.0.100 vrf CL2022:vrf1
Address         MAC Address      Interface
10.55.0.100     0005.73ff.593c   vlan25
```

5d

```
show mac address addr 0005.73ff.593c vl 25
   VLAN      MAC Address      Ports
---------+----------------+----------
* 25       0005.73ff.593c   eth1/27/4
```

Forward based on ARP and MAC Adajcencies

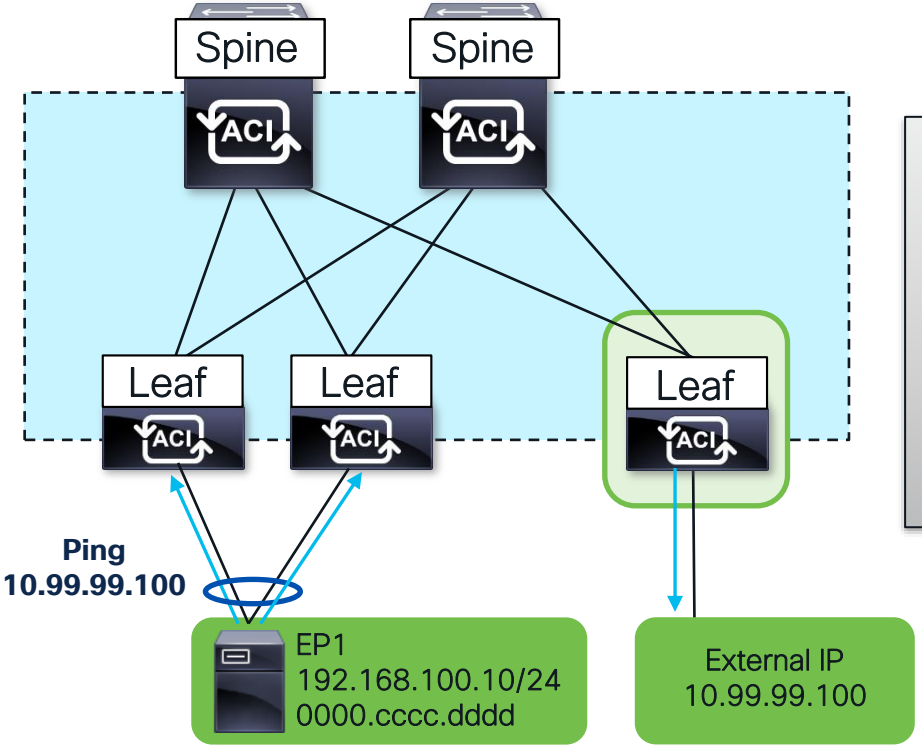


5 Since received VNID is the VRF VNID, forward based on dest endpoint IP

6 Policy was applied by ingress leaf. No need to apply contracts

L3Out Destination – Egress Leaf

Bridge Domain Settings:
Unicast Routing Enabled



Elam

```
debug plat internal app elam asic 0
 trigger reset
 trigger init in-select 14 out-select 0
  set inner ipv4 src_ip 192.168.100.10
  set inner ipv4 dst_ip 10.99.99.100
  start
stat
 ELAM STATUS
===========
Asic 0 Slice 0 Status Triggered
Asic 0 Slice 1 Status Armed
```

L3Out Destination – Egress Leaf

Bridge Domain Settings: Unicast Routing Enabled

```
Inner L2 Header
---------------------------------------
Inner Destination MAC   : 000C.0C0C.0C0C

Inner L3 Header
---------------------------------------
Destination IP          : 10.99.99.10

Outer L4 Header
---------------------------------------
L4 Type                 : iVxLAN
Src Policy Applied Bit  : 1
Dst Policy Applied Bit  : 1
VRF or BD VNID          : 2523136( 0x268000 )

Sideband Information
---------------------------------
ovector                 : 147( 0x93 )

FINAL FORWARDING LOOKUP
-----------------------------------------------------------------------
Bits set in Final Forwarding Block: IFABRIC_EG UC INFRA ENCAP MYTEP ROUTE HIT

Lookup Drop
------------------------
LU drop reason    : no drop
```

Contracts have already been applied. No need to check.

IP lookup done in VRF with this VNID

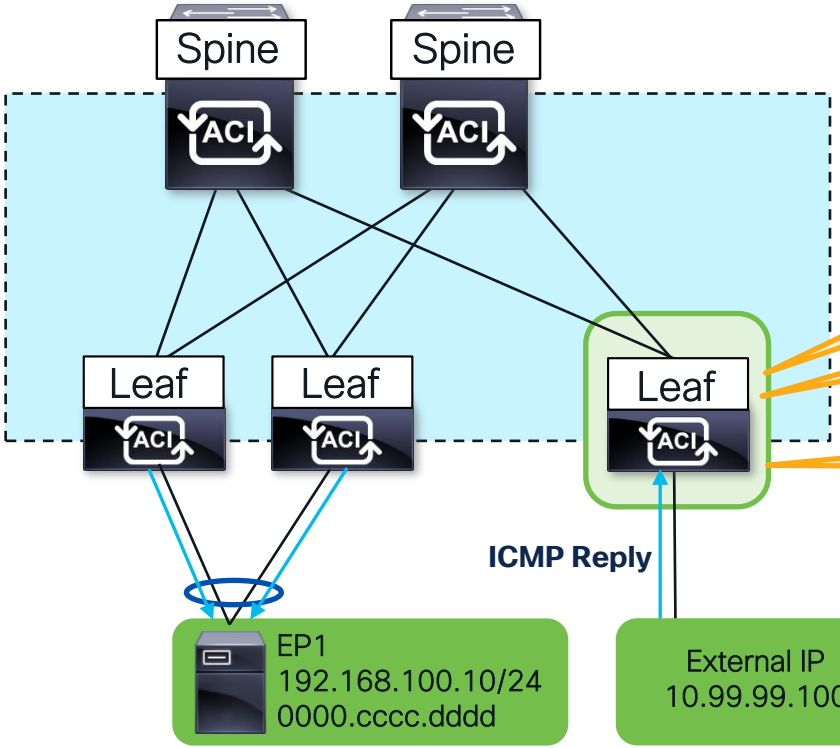
```
show platform internal hal l2 port gpd
=========================================
IfId      Ifname      As AP  Sl Sp Ss Ovec
=========================================
4301a000  Eth1/27/4   0  54  2  13 13 93
```

Forward out Eth1/27/4!

Unicast + Route (L3 lookup) + L3 Route Found

L3Out Source – Ingress Border Leaf

Bridge Domain Settings:
Unicast Routing Enabled



2b If dest IP is not learned endpoint and subnet is BD subnet, proxy!

2a Forward based on longest prefix-match within source VRF. EP learns are always longest.

1 If VRF is in ingress mode, BL doesn't apply policy

Refer back to the Routed Known Unicast and Proxied Unicast for more verifications

Complete your Session Survey

- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (open from Thursday) to receive your Cisco Live t-shirt.

- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Session Catalog and clicking the "Attendee Dashboard" at https://www.ciscolive.com/emea/learn/sessions/session-catalog.html

Continue Your Education

Visit the Cisco Showcase for related demos.

Book your one-on-one Meet the Engineer meeting.

Attend any of the related sessions at the DevNet, Capture the Flag, and Walk-in Labs zones.

Visit the On-Demand Library for more sessions at ciscolive.com/on-demand.

The bridge to possible

Thank you

cisco Live!

Early Access. Yes, please.

Cisco U.

Tech learning, shaped to you.

cisco

CISCO Live!

ALL IN