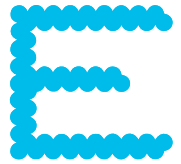
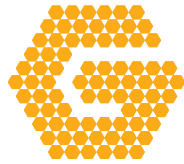
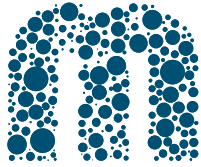


Cisco *live!*

January 28 - February 1, 2019 - Barcelona



INTUITIVE



BRKACI-3545

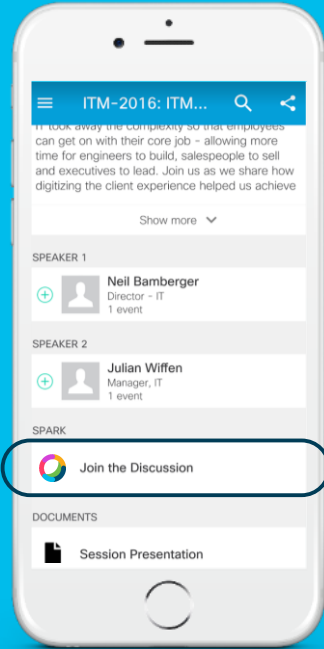
Mastering ACI Forwarding Behavior

- A day in the life of a packet -

Takuya Kishida, Technical Leader, Service



INTUITIVE



cs.co/cisolivebot#BRKACI-3545

Cisco Webex Teams

Questions?

Use Cisco Webex Teams (formerly Cisco Spark) to chat with the speaker after the session

How

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

Agenda

- Introduction
 - ACI Overlay VxLAN and TEP
- ACI Forwarding components
 - Endpoints, EPG, EP Learning, COOP and How it all works
 - BD, VRF forwarding scope and detailed options
 - Spine-Proxy and ARP Glean
 - Forwarding Software Architecture and ASIC Generation
- ACI Packet Walk
 - Walk through the life of a packet going through ACI

Basic Acronyms/Definitions

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
dXXXo	Outer Destination XXX (dIPo = Outer Destination IP)
sXXXo	Outer Source XXX (sIPo = Outer Source IP)
dXXXi	Inner Destination XXX (dIPi = Inner Destination IP)
sXXXi	Inner Source XXX (sIPi = Inner Source IP)
GIPO	Outer Multicast Group IP
VNID	Virtual Network Identifier

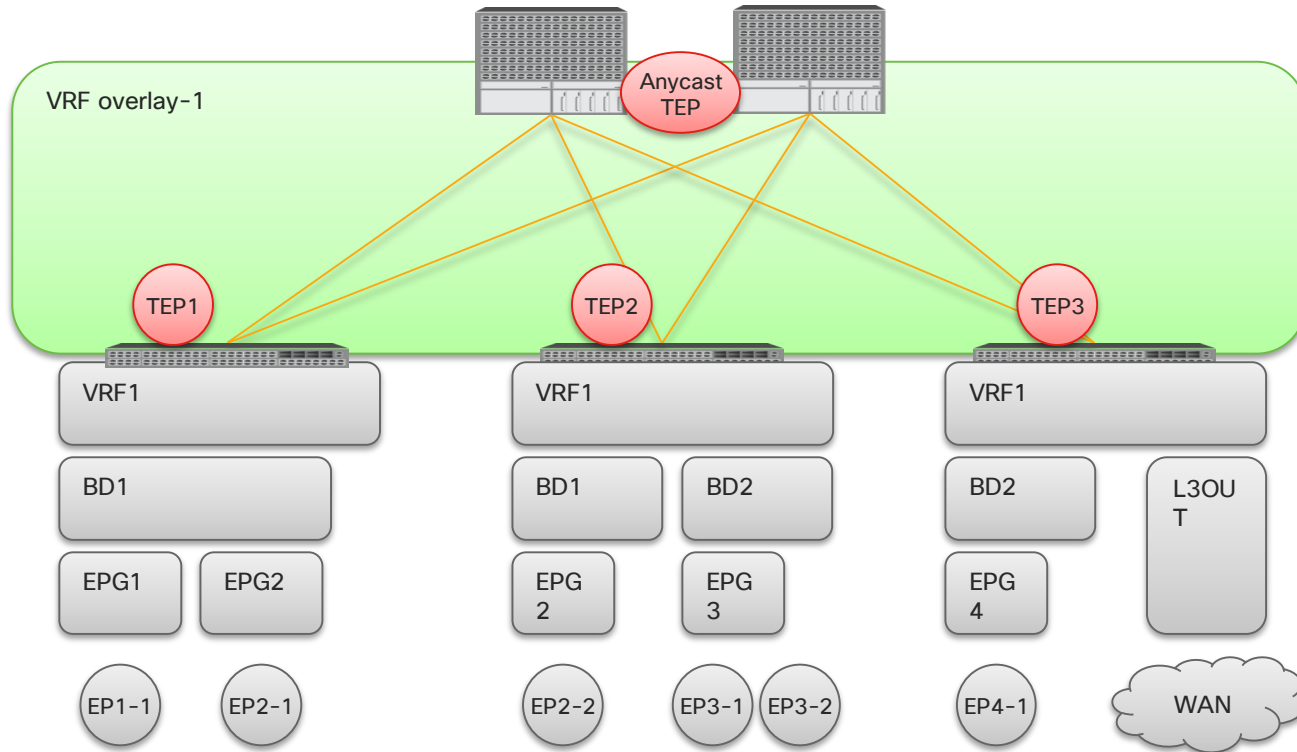


Agenda

- Introduction
 - ACI Overlay VxLAN and TEP
- ACI Forwarding components
 - Endpoints, EPG, EP Learning, COOP and How it all works
 - BD, VRF forwarding scope and detailed options
 - Spine-Proxy and ARP Glean
 - Forwarding Software Architecture and ASIC Generation
- ACI Packet Walk
 - Walk through the life of a packet going through ACI

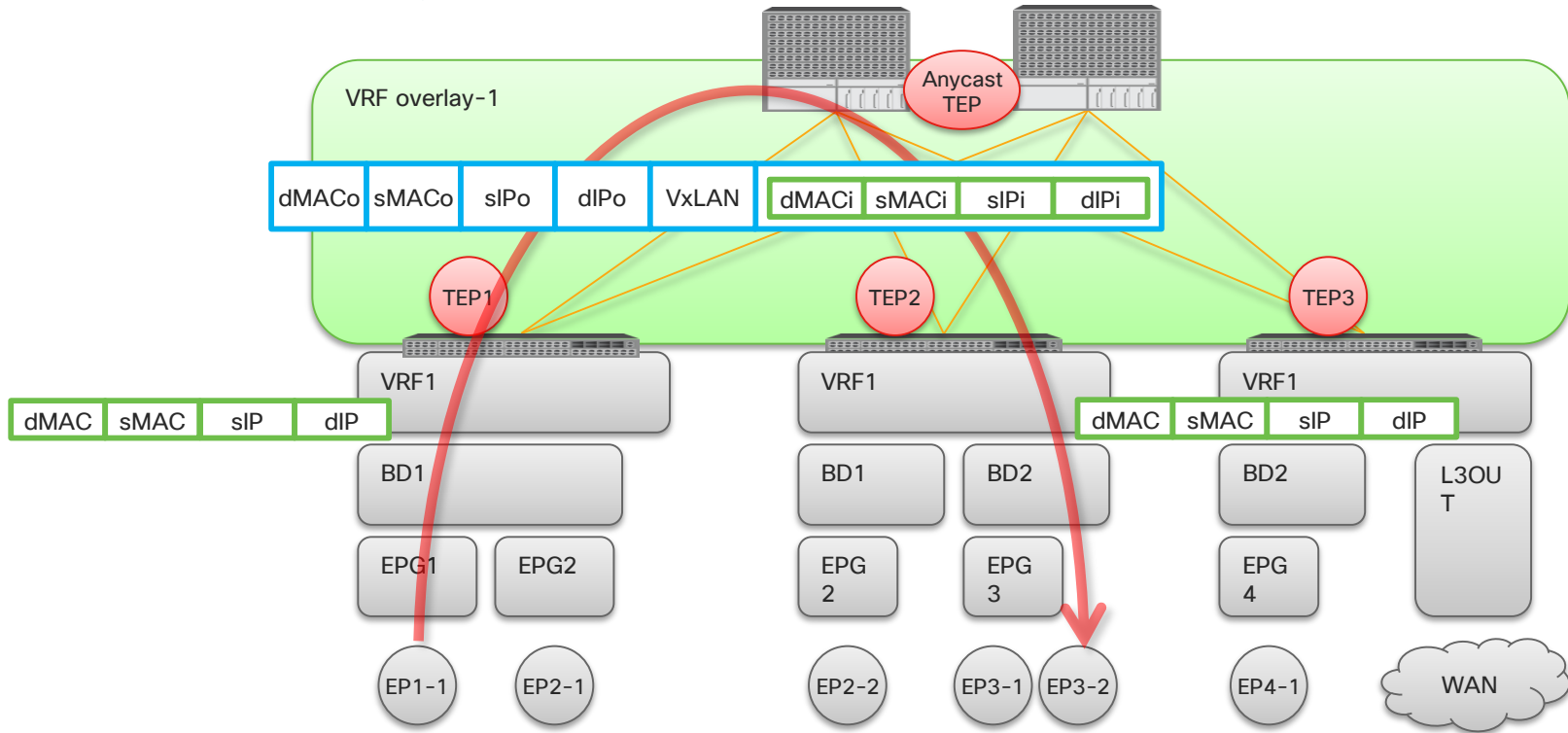
ACI Overlay VxLAN and TEP

✳️ TEP : Tunnel EndPoint



ACI Overlay VxLAN and TEP

✳️ TEP : Tunnel EndPoint



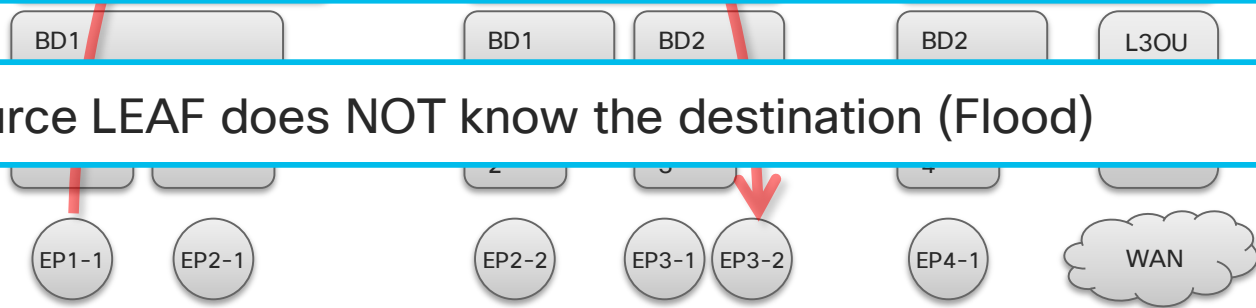
ACI Overlay VxLAN and TEP

Scenario 1 : source LEAF knows the destination (on the same LEAF)

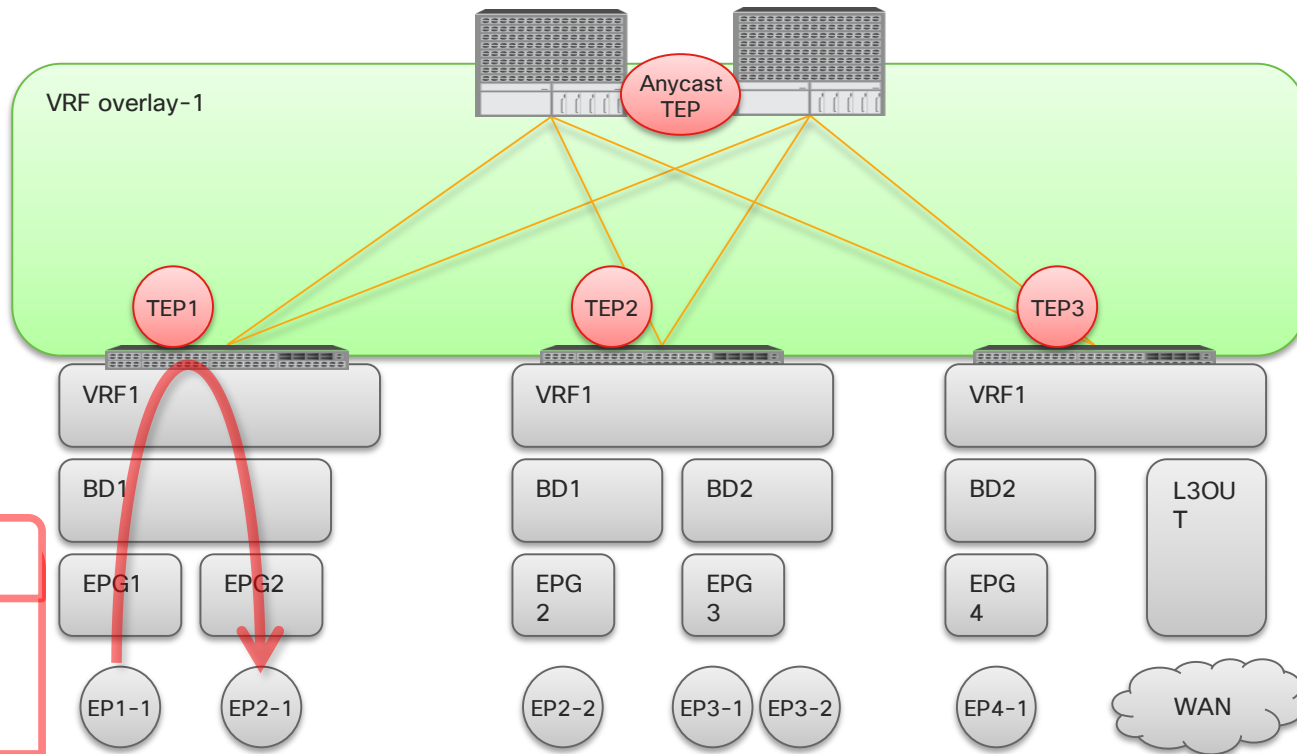
Scenario 2 : source LEAF knows the destination (on another LEAF X)

Scenario 3 : source LEAF does NOT know the destination (Spine-Proxy)

Scenario 4 : source LEAF does NOT know the destination (Flood)



Source LEAF knows the destination (on the same LEAF)



Local EndPoint

Quite Similar to
normal L2/L3
Forwarding

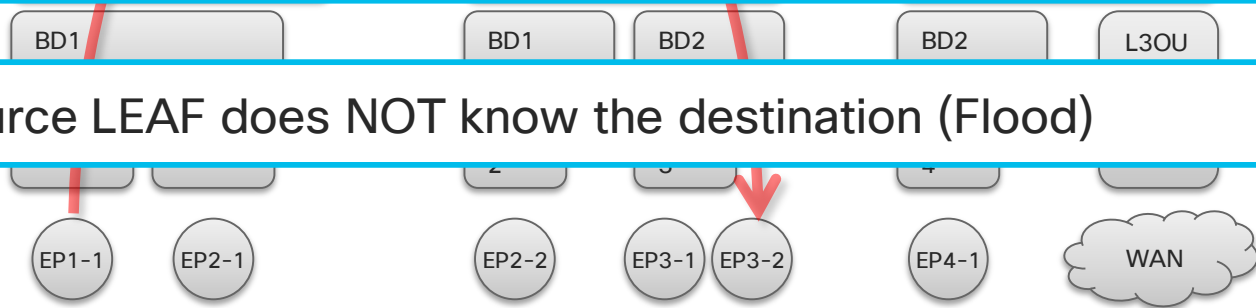
ACI Overlay VxLAN and TEP

Scenario 1 : source LEAF knows the destination (on the same LEAF)

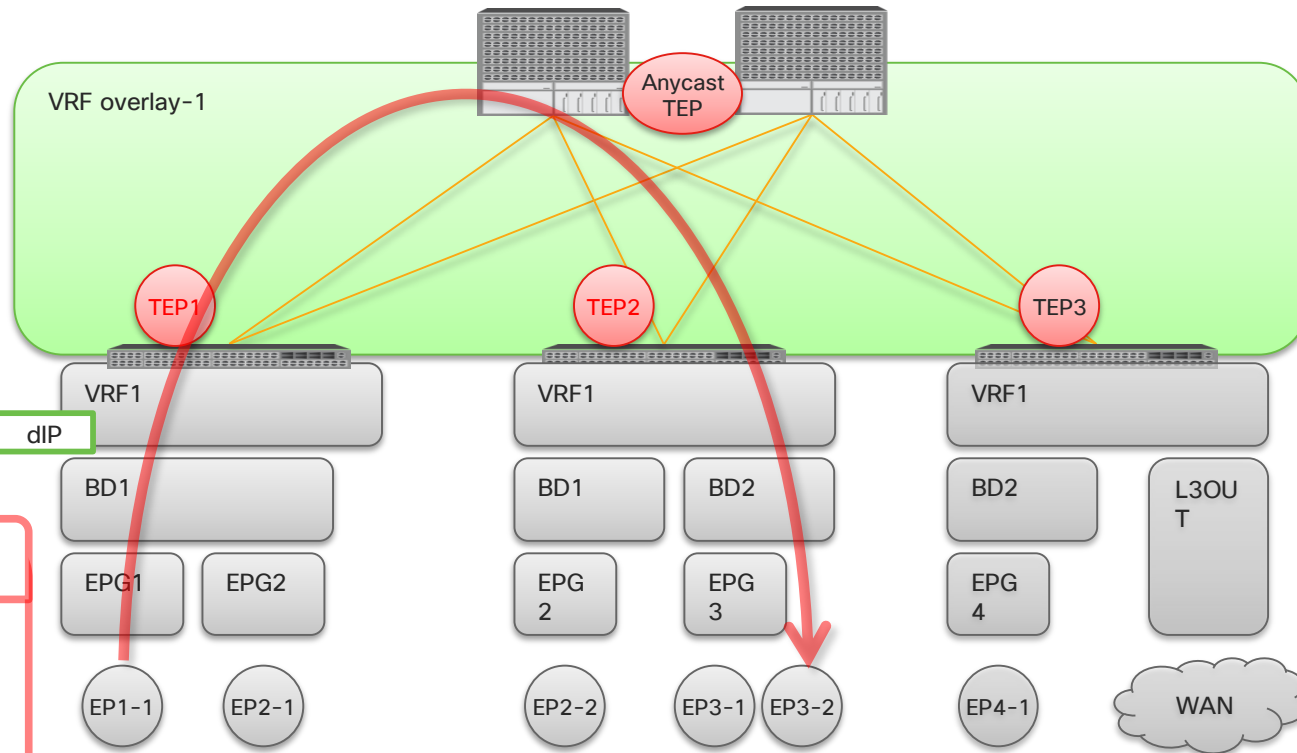
Scenario 2 : source LEAF knows the destination (on another LEAF X)

Scenario 3 : source LEAF does NOT know the destination (Spine-Proxy)

Scenario 4 : source LEAF does NOT know the destination (Flood)



Source LEAF knows the destination (on the remote LEAF)



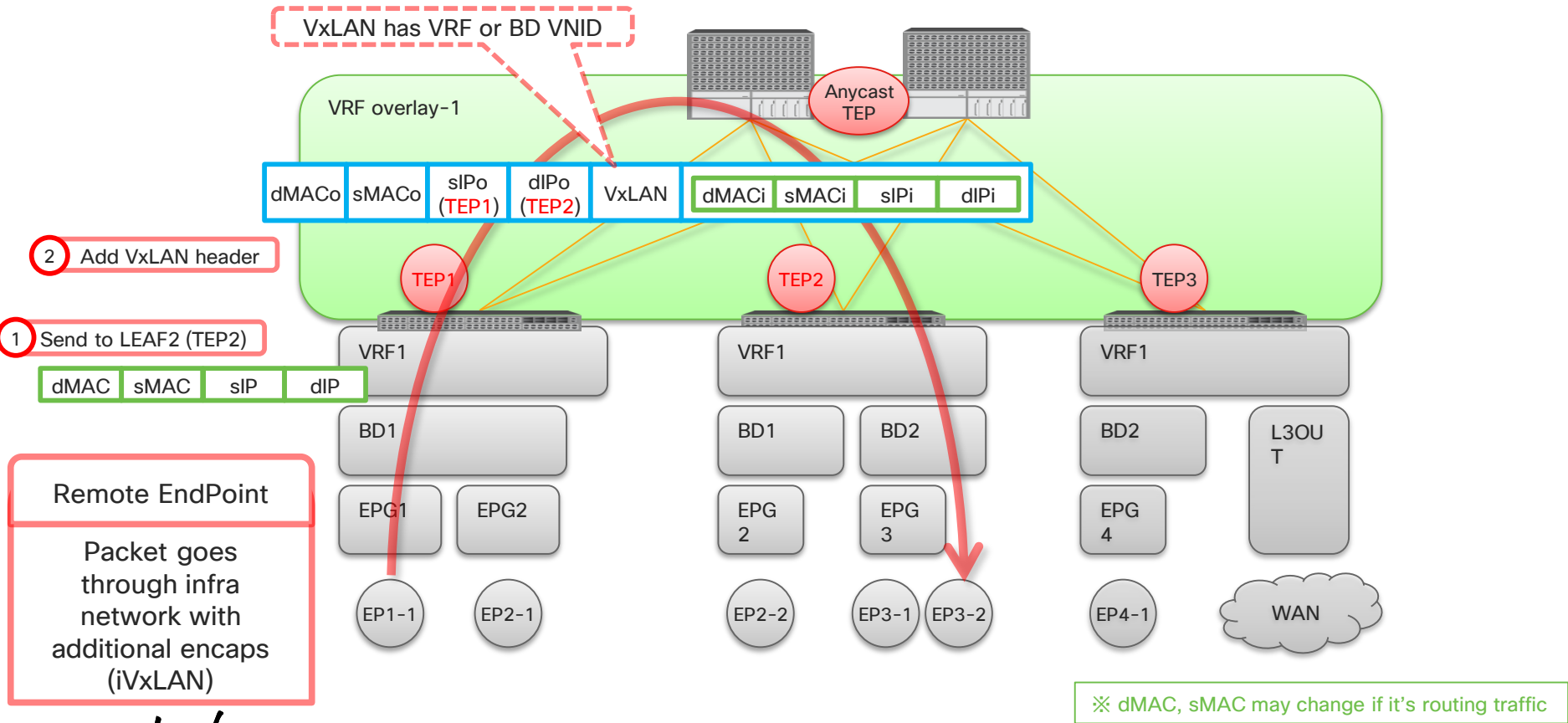
1 Send to LEAF2 (TEP2)

dMAC sMAC sIP dIP

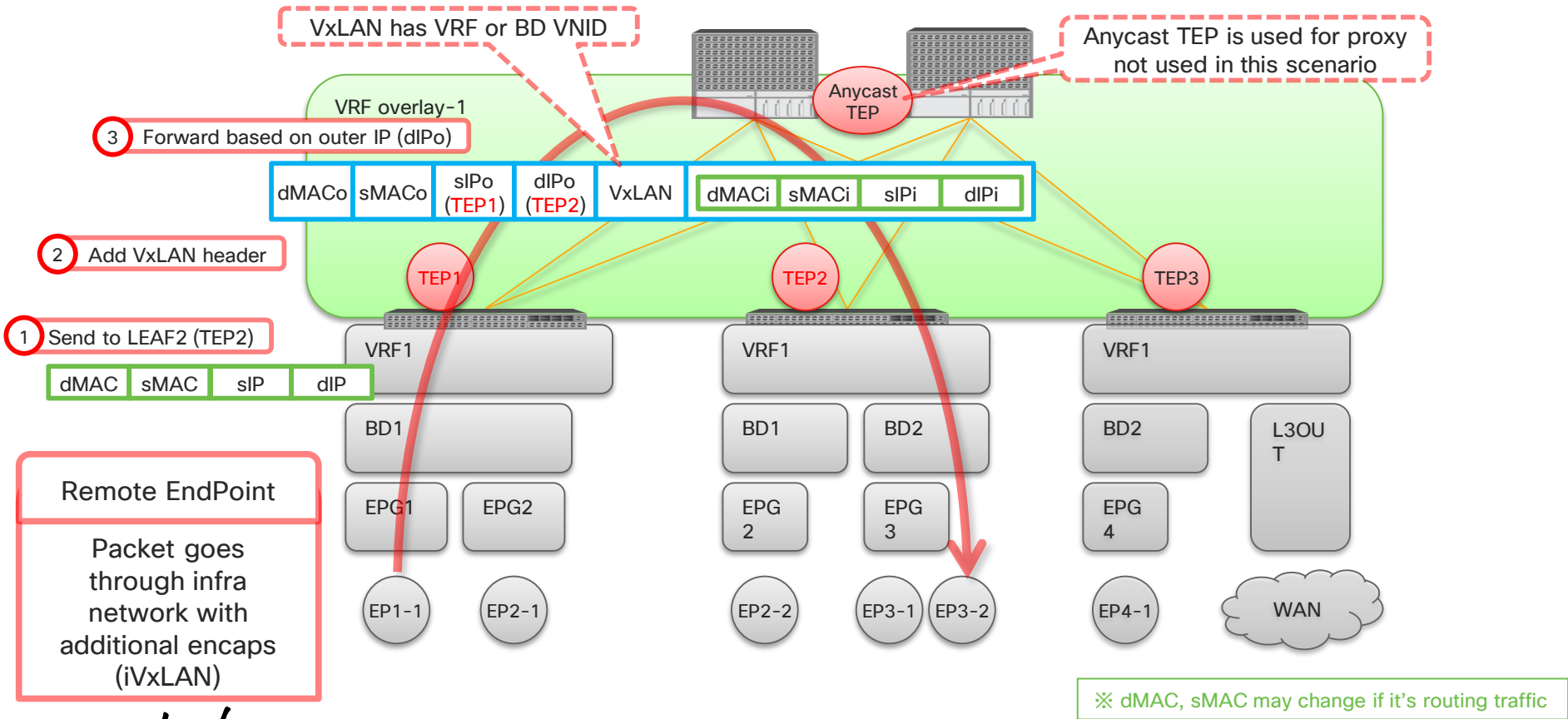
Remote EndPoint
Packet goes through infra network with additional encaps (iVxLAN)

※ dMAC, sMAC may change if it's routing traffic

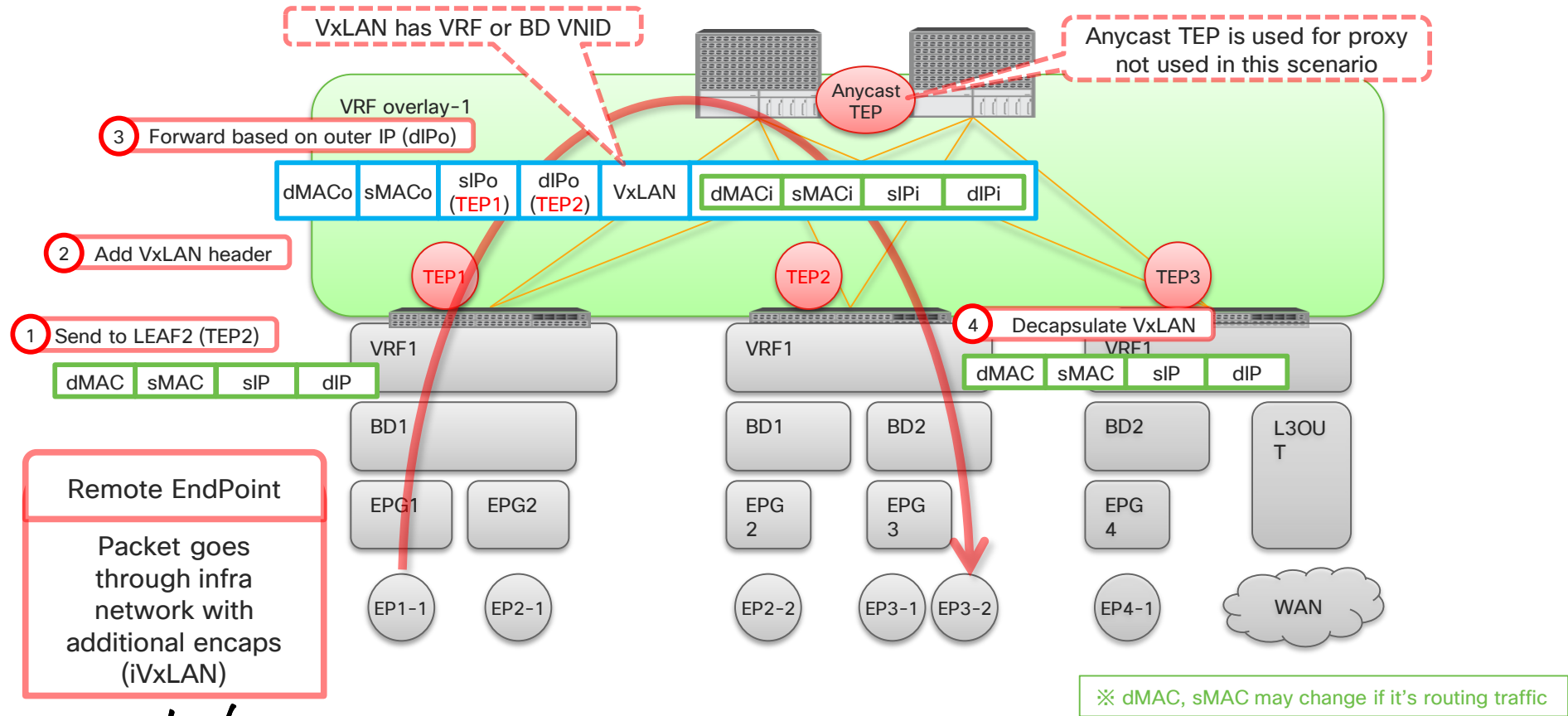
Source LEAF knows the destination (on the remote LEAF)



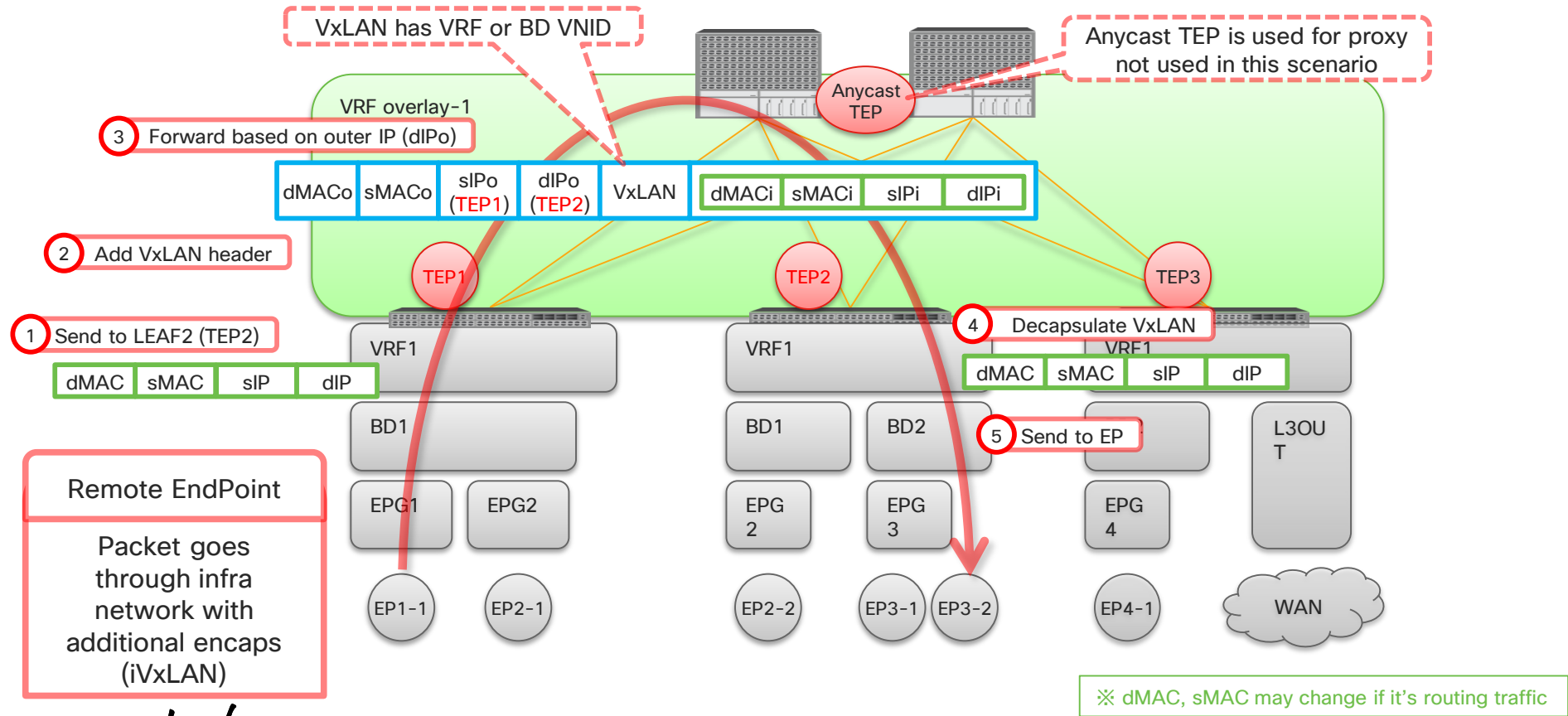
Source LEAF knows the destination (on the remote LEAF)



Source LEAF knows the destination (on the remote LEAF)



Source LEAF knows the destination (on the remote LEAF)



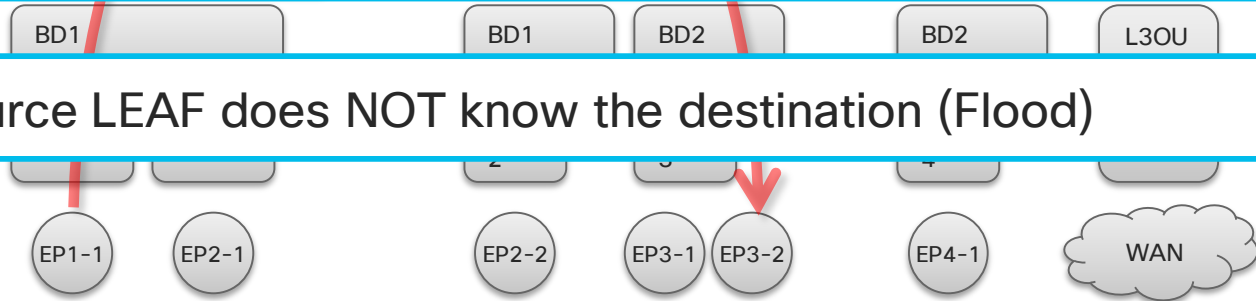
ACI Overlay VxLAN and TEP

Scenario 1 : source LEAF knows the destination (on the same LEAF)

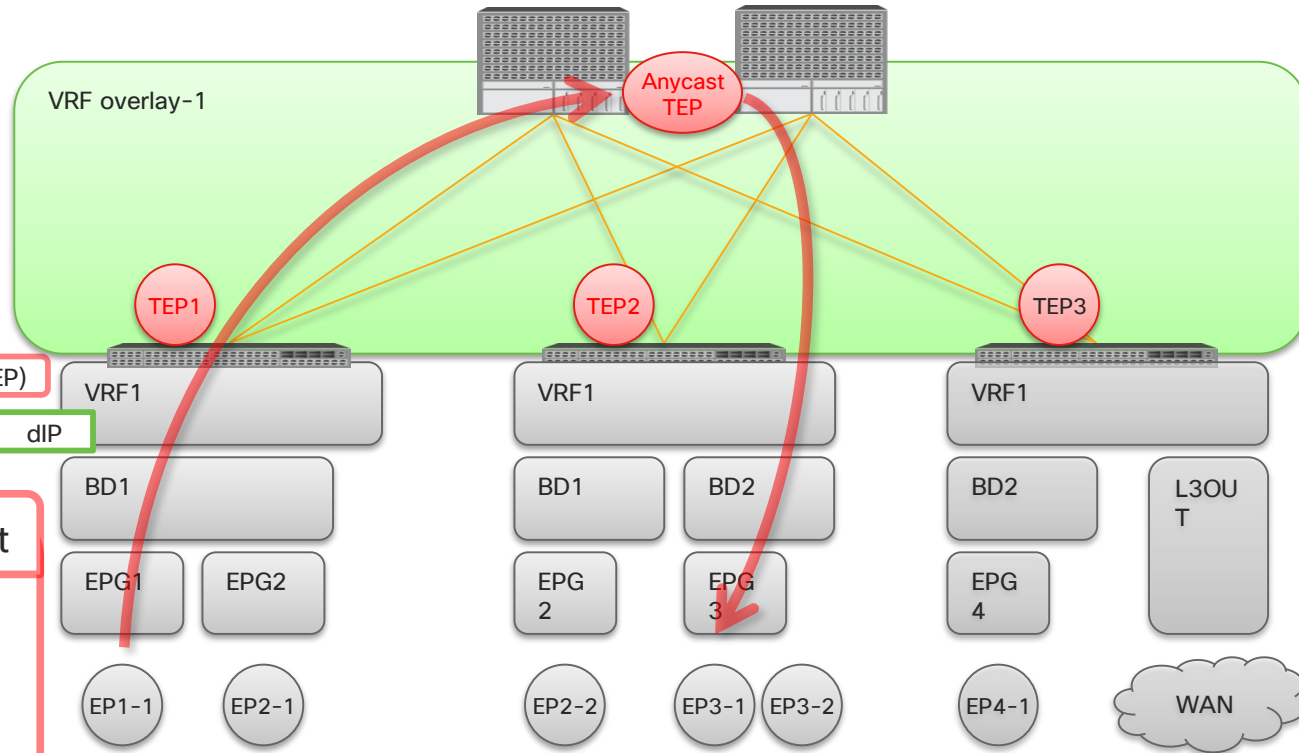
Scenario 2 : source LEAF knows the destination (on another LEAF X)

Scenario 3 : source LEAF does NOT know the destination (Spine-Proxy)

Scenario 4 : source LEAF does NOT know the destination (Flood)



Source LEAF does NOT know the destination (Spine-Proxy)



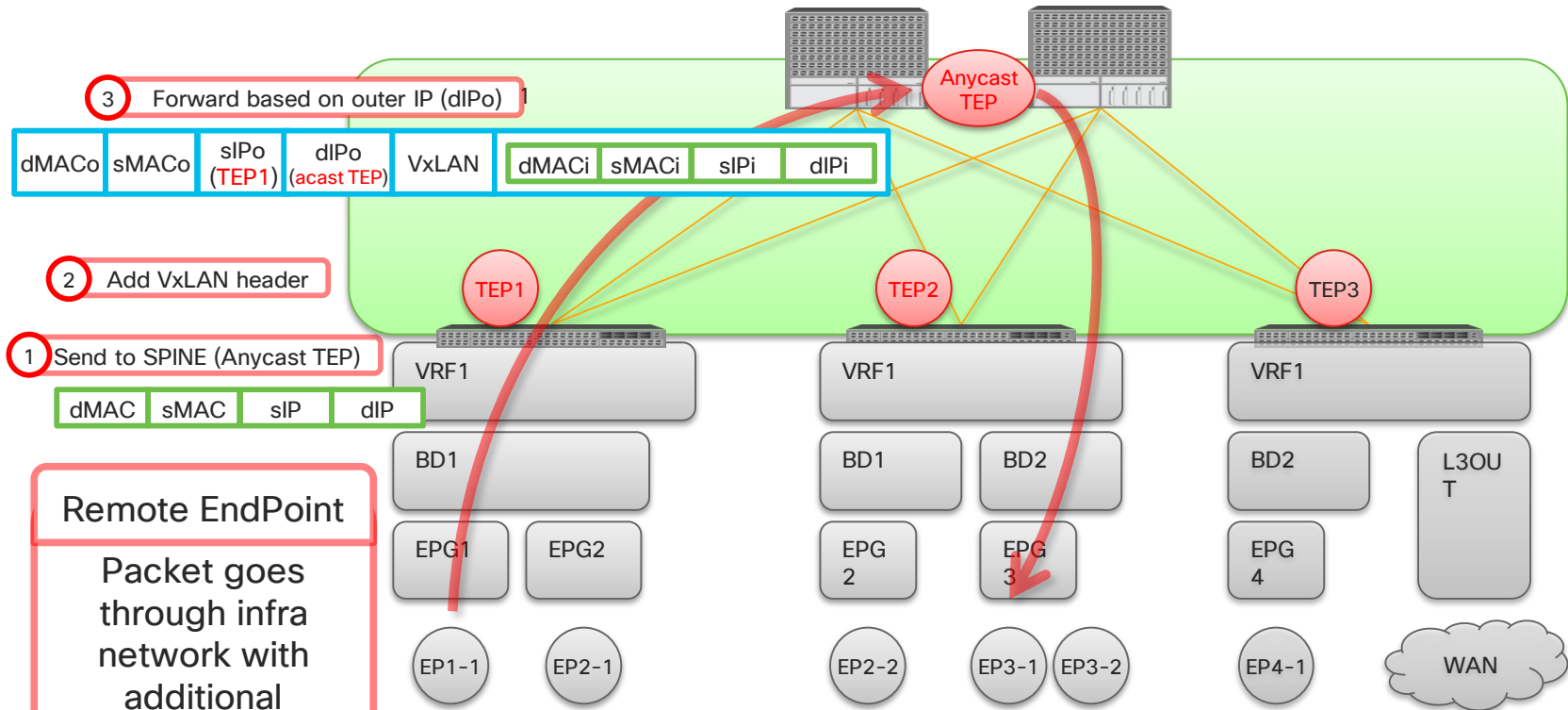
1 Send to SPINE (Anycast TEP)

dMAC sMAC sIP dIP

Remote EndPoint
Packet goes through infra network with additional encaps (iVxLAN)

※ dMAC, sMAC may change if it's routing traffic

Source LEAF does NOT know the destination (Spine-Proxy)



Remote EndPoint
 Packet goes through infra network with additional encaps (iVxLAN)

※ dMAC, sMAC may change if it's routing traffic

Source LEAF does NOT know the destination (Spine-Proxy)

4 Lookup dMACi or dIPi with BD or VRF VNID in VxLAN

3 Forward based on outer IP (dIPo)

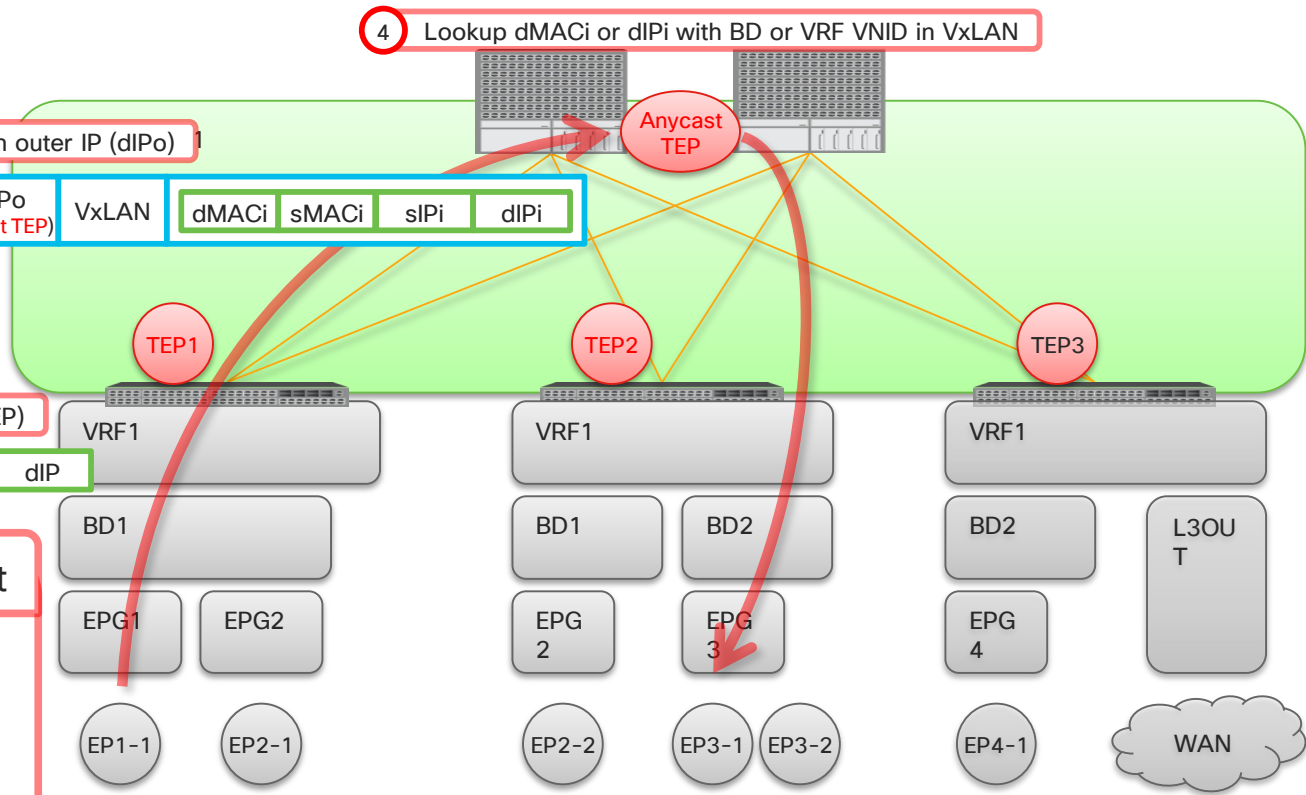
2 Add VxLAN header

1 Send to SPINE (Anycast TEP)



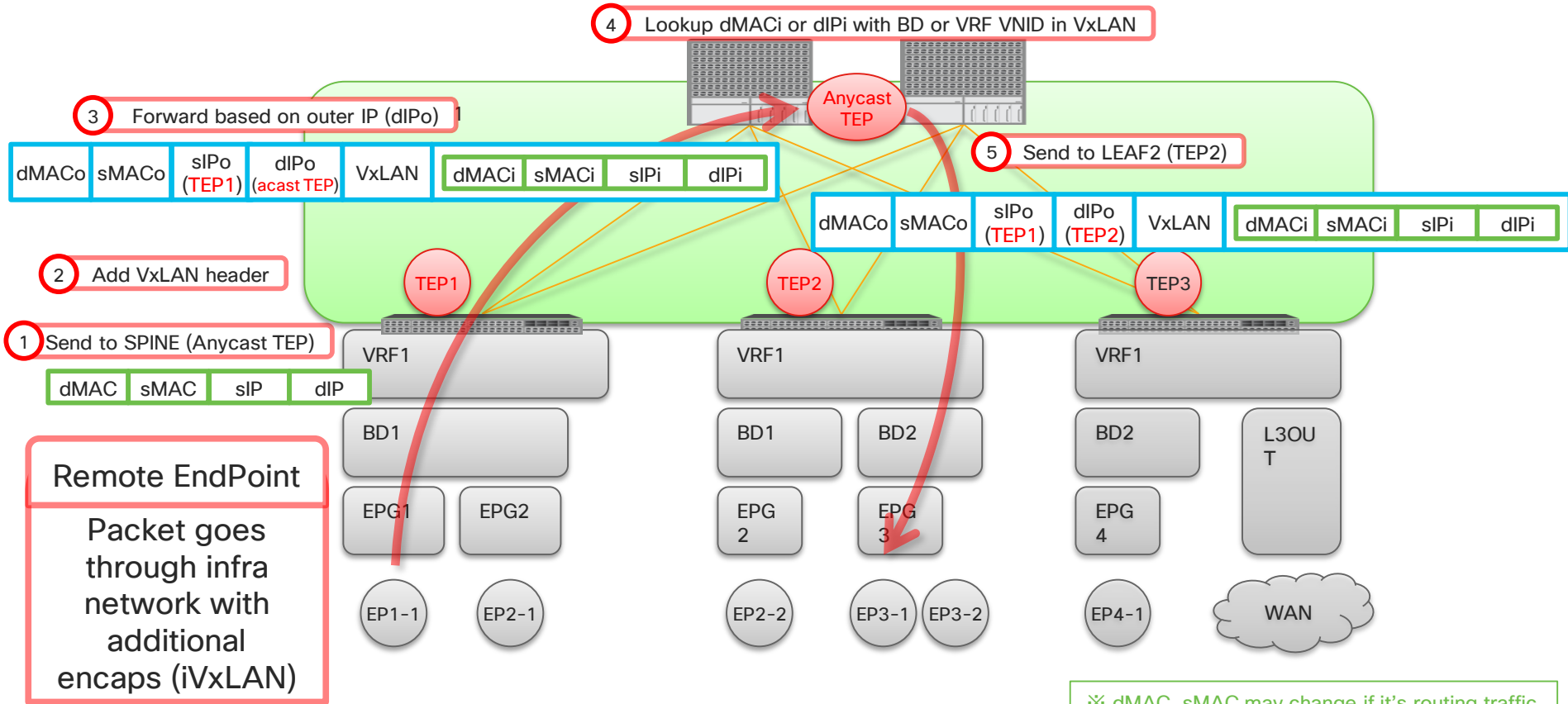
Remote EndPoint

Packet goes through infra network with additional encaps (iVxLAN)

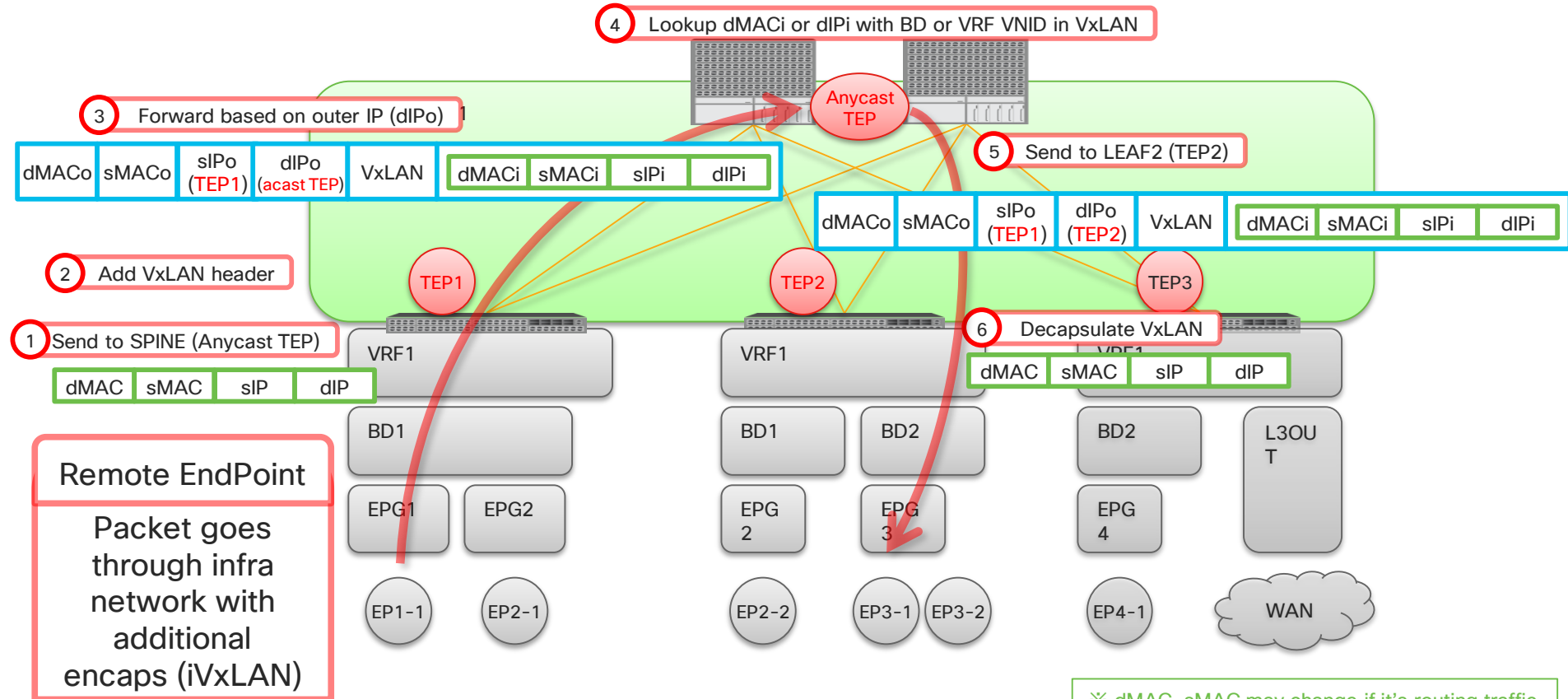


※ dMAC, sMAC may change if it's routing traffic

Source LEAF does NOT know the destination (Spine-Proxy)

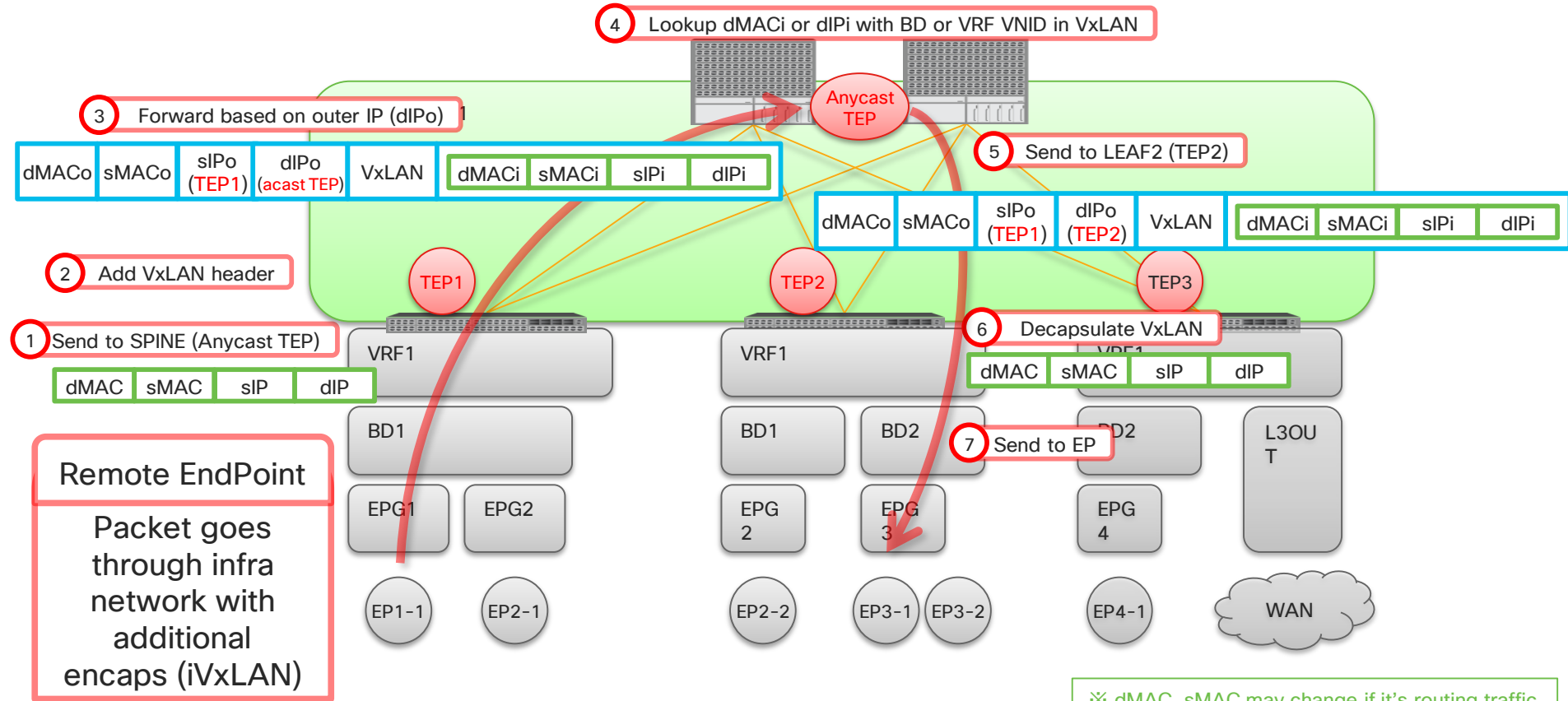


Source LEAF does NOT know the destination (Spine-Proxy)



※ dMAC, sMAC may change if it's routing traffic

Source LEAF does NOT know the destination (Spine-Proxy)



ACI Overlay VxLAN and TEP

Scenario 1 : source LEAF knows the destination (on the same LEAF)



Scenario 2 : source LEAF knows the destination (on another LEAF X)



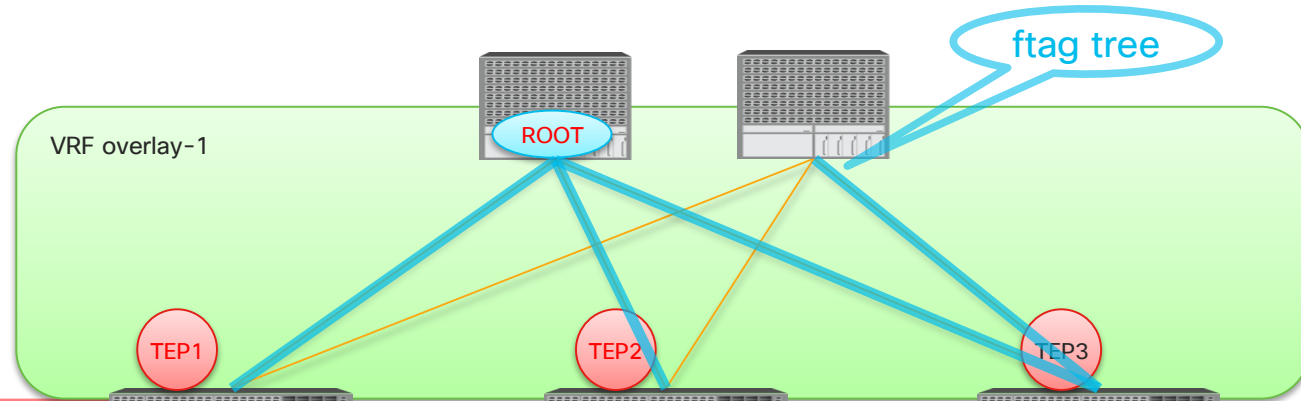
Scenario 3 : source LEAF does NOT know the destination (Spine-Proxy)



Scenario 4 : source LEAF does NOT know the destination (Flood)



Source LEAF does NOT know the destination (Flood)

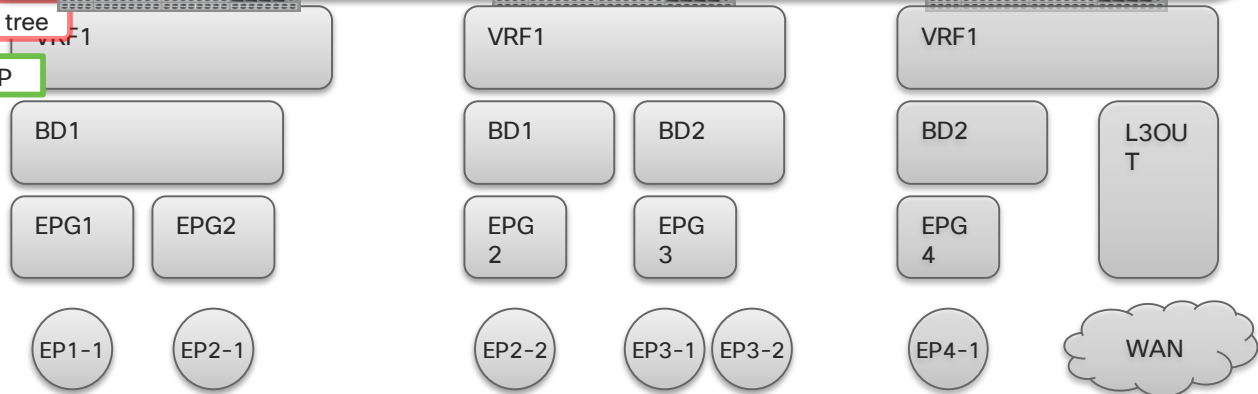


1 Flood with BD mcast TEP and tree

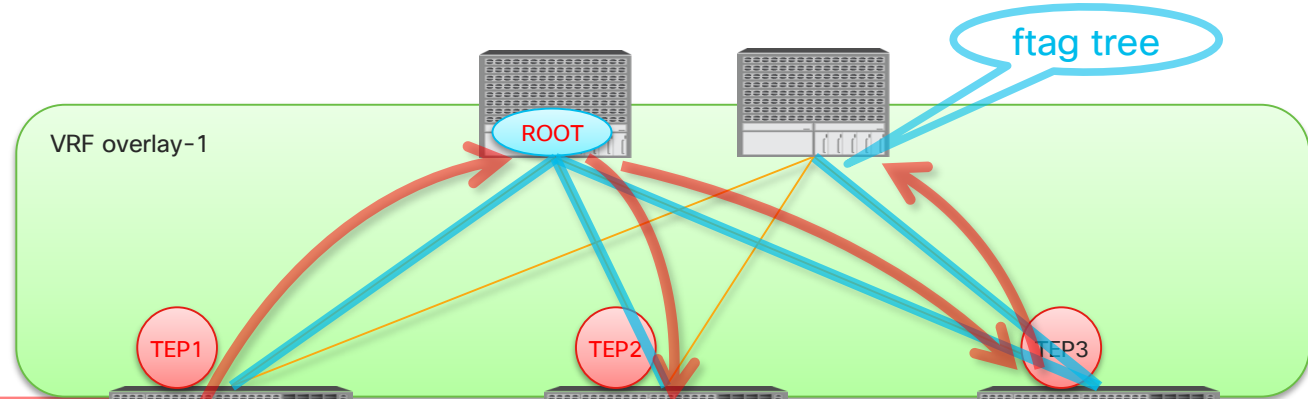
dMAC sMAC sIP dIP

Remote EndPoint

Packet goes through infra network with additional encaps (iVxLAN)



Source LEAF does NOT know the destination (Flood)

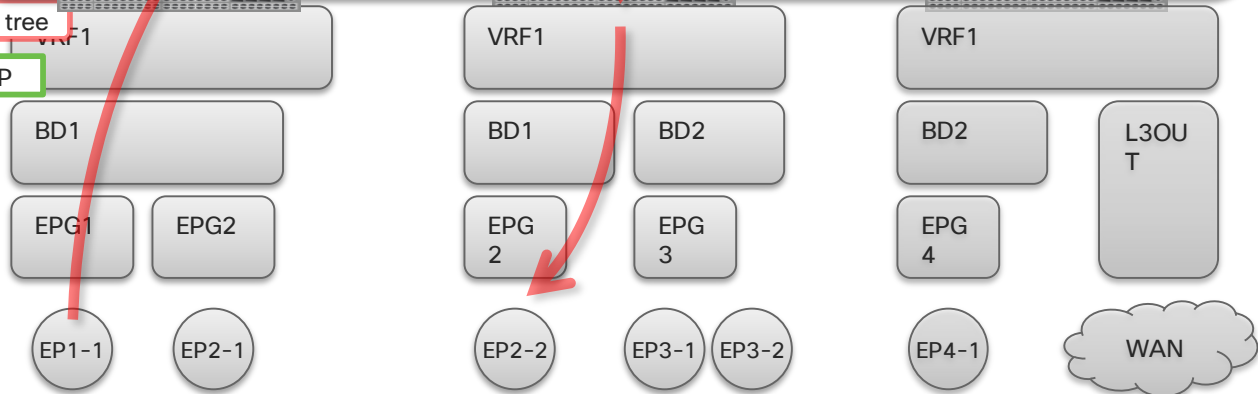


1 Flood with BD mcast TEP and tree

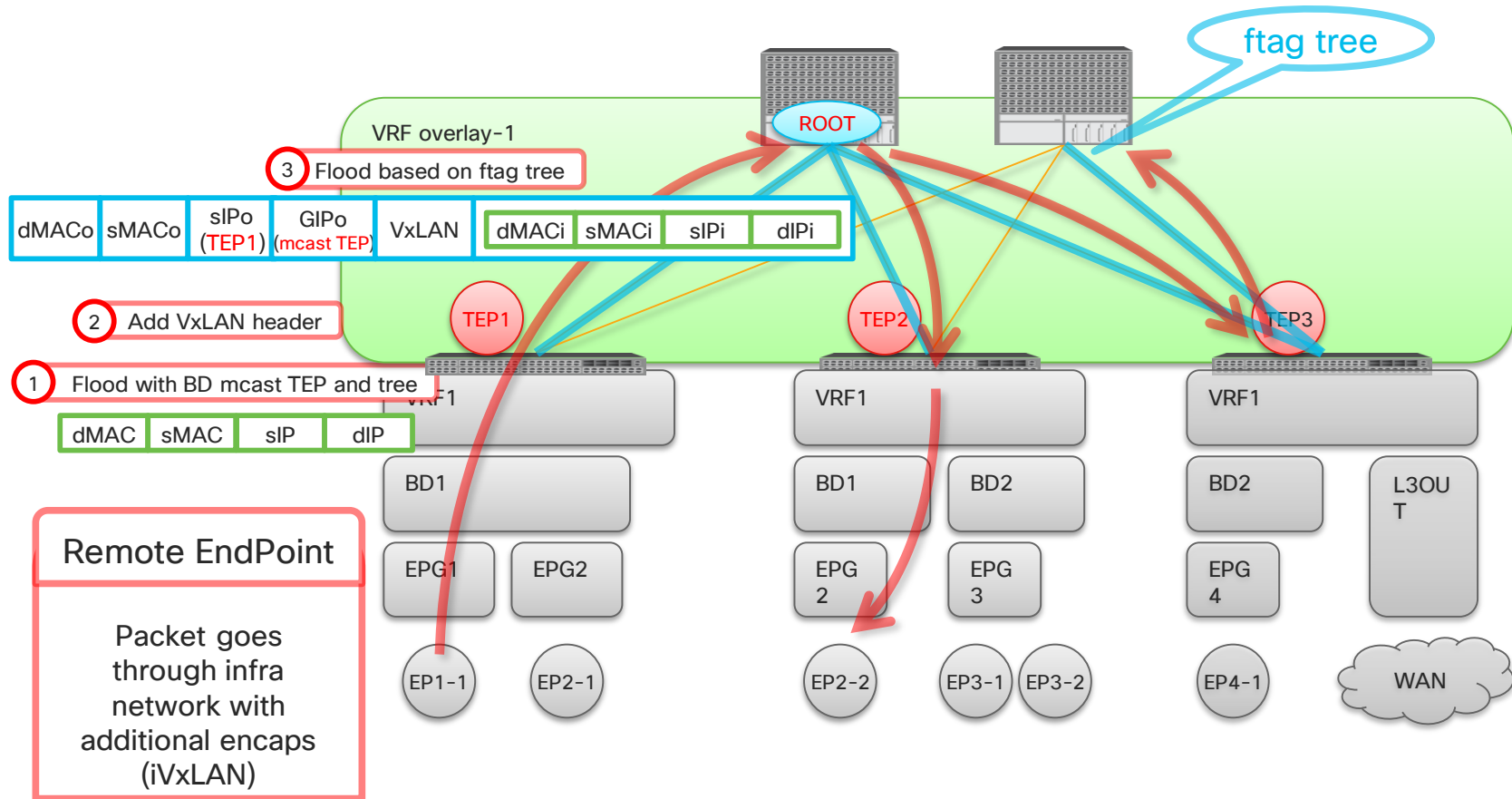
dMAC sMAC sIP dIP

Remote EndPoint

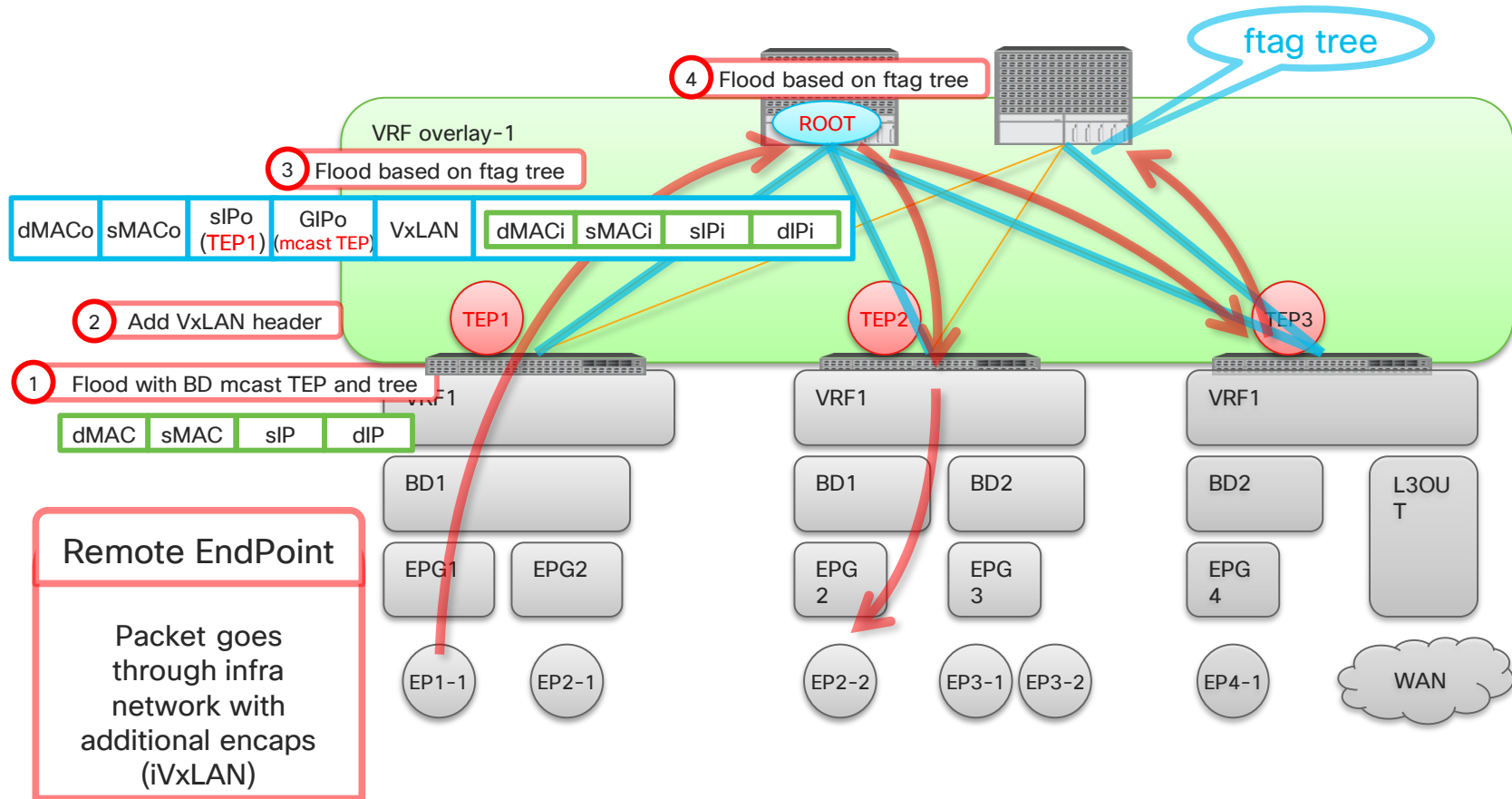
Packet goes through infra network with additional encaps (iVxLAN)



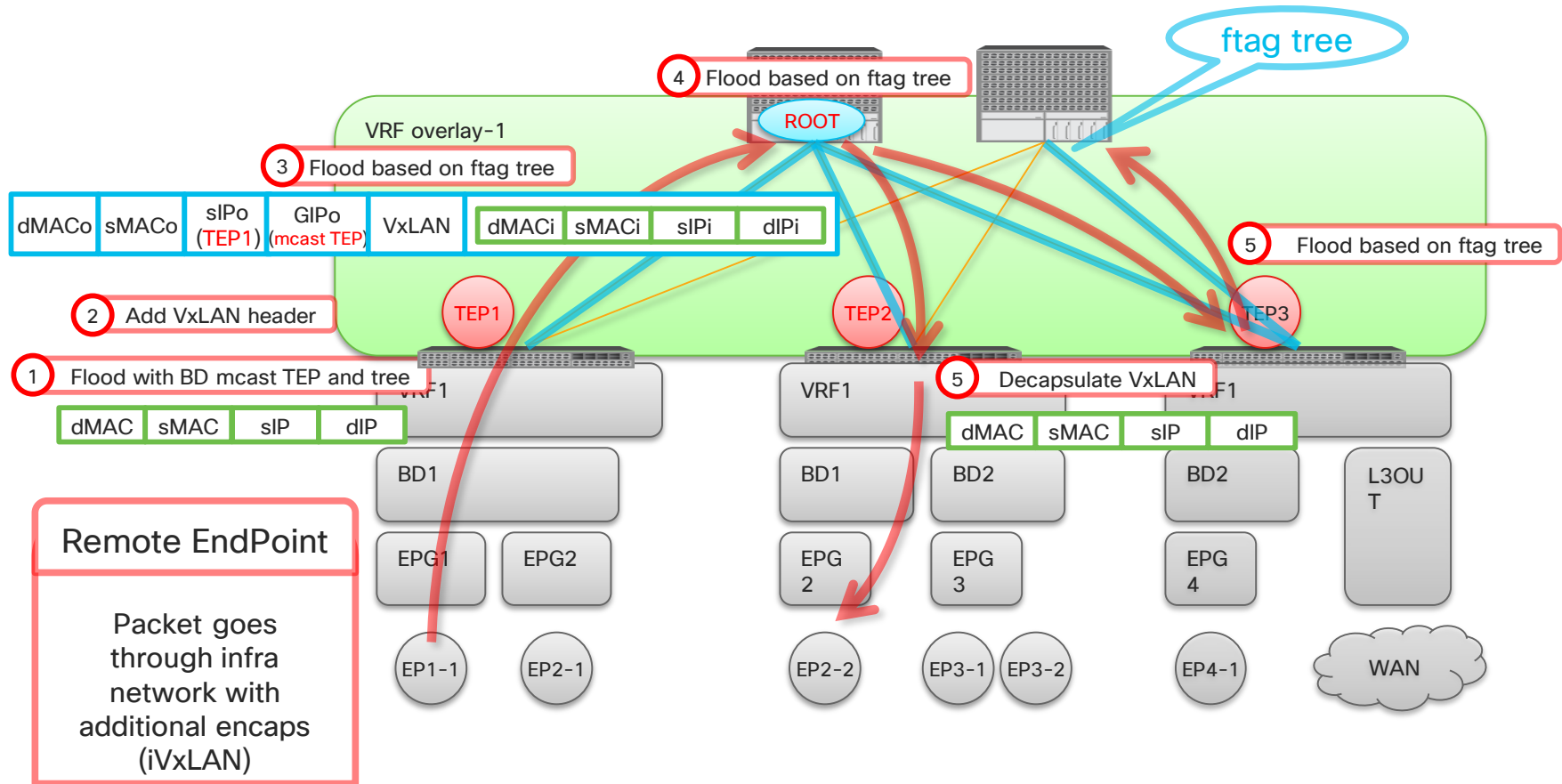
Source LEAF does NOT know the destination (Flood)



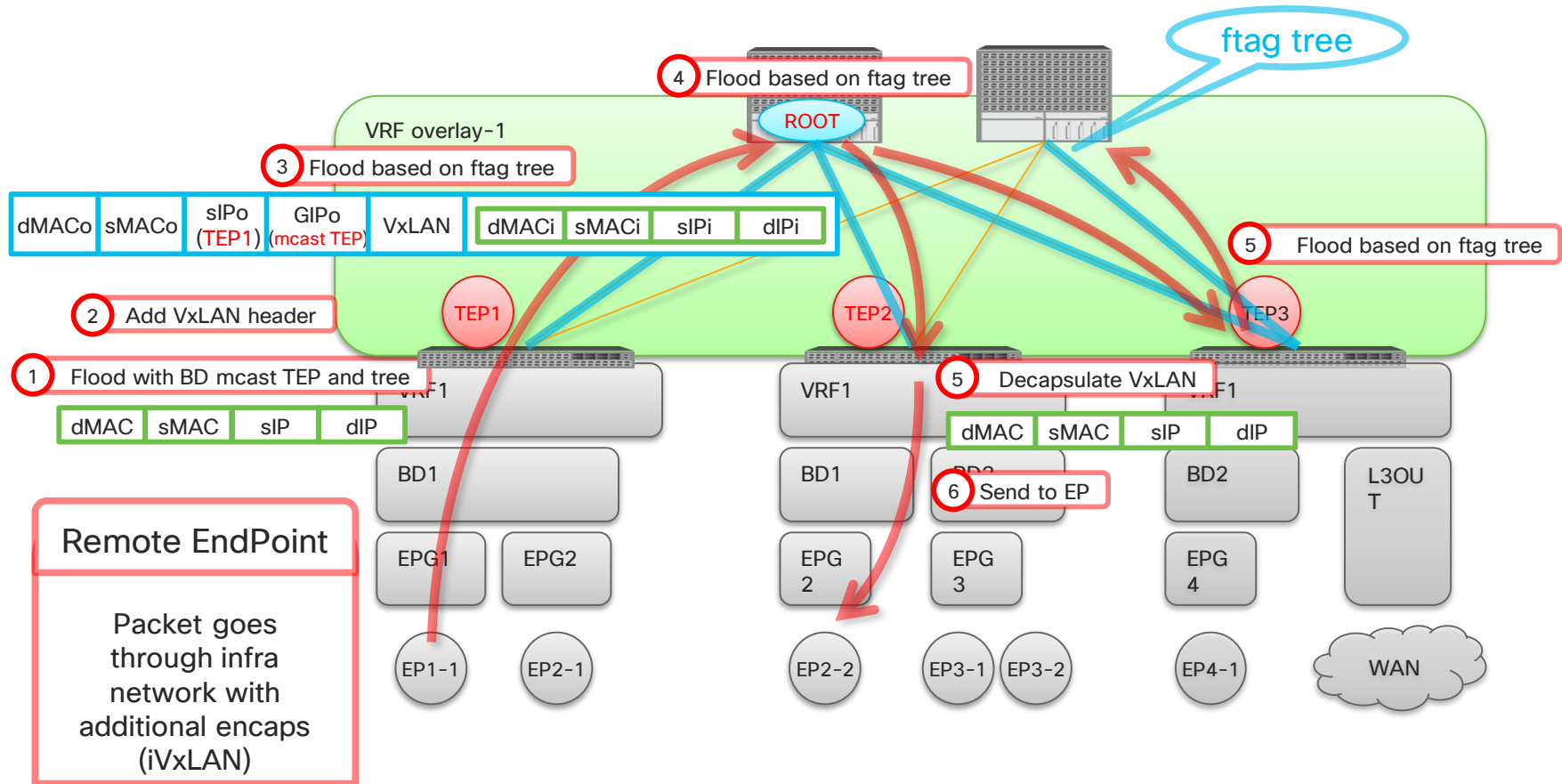
Source LEAF does NOT know the destination (Flood)



Source LEAF does NOT know the destination (Flood)



Source LEAF does NOT know the destination (Flood)



ACI Overlay VxLAN and TEP

Scenario 1 : source LEAF knows the destination (on the same LEAF)

Scenario 2 : source LEAF knows the destination (on another LEAF X)

Scenario 3 : source LEAF does NOT know the destination (Spine-Proxy)

Scenario 4 : source LEAF does NOT know the destination (Flood)

How does LEAF pick one of these scenario?

➤ **based on EP information**

Agenda

- Introduction
 - ACI Overlay VxLAN and TEP
- ACI Forwarding components
 - Endpoints, EPG, EP Learning, COOP and How it all works
 - BD, VRF forwarding scope and detailed options
 - Spine-Proxy and ARP Glean
 - Forwarding Software Architecture and ASIC Generation
- ACI Packet Walk
 - Walk through the life of a packet going through ACI

ACI Forwarding Component 1

- Endpoint
- EPG (EndPoint Group)
- VLAN Type in ACI
- Endpoint Type
- Endpoint Learning
- COOP (Council of Oracle Protocol)

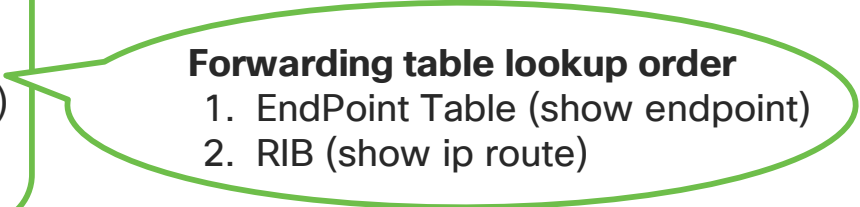
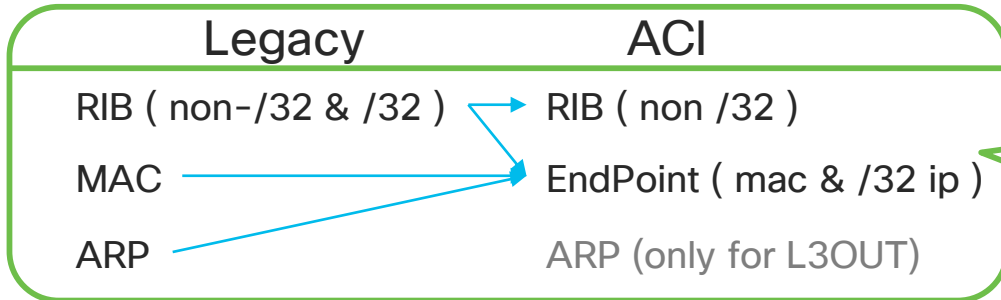
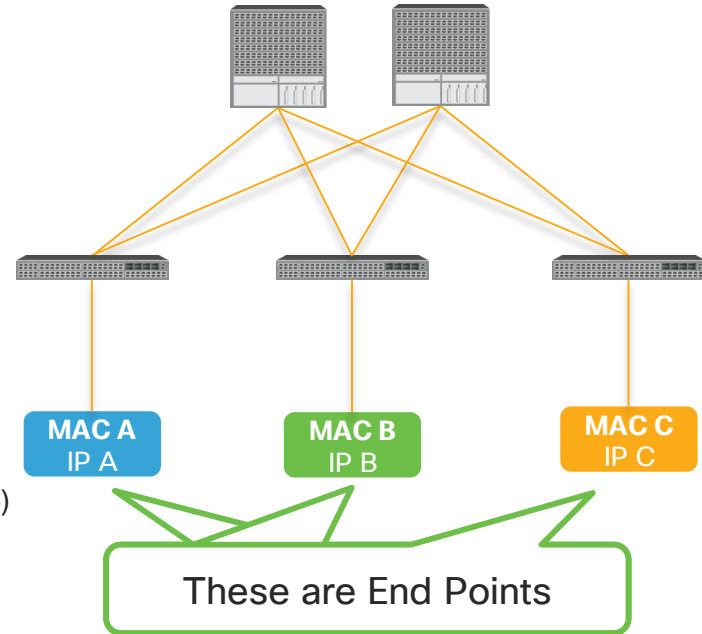
End Point (EP)

What is an EP?

- It stands for hosts, in other words MAC address with IP(s)
 - sometimes MAC only
 - IP in EP is always /32

What Forwarding Table is used?

- End Point Table
 - host information (MAC and /32 IP address)
- LPM(Longest Prefix Match) Table
 - non /32 IP route information (exception: /32 for SVI or L3OUT route)



End Point Group (EPG)

What is an EPG?

- Logical grouping of hosts (EPs)
- Each EPG belongs to a Bridge Domain (BD).

What is the EPG used for?

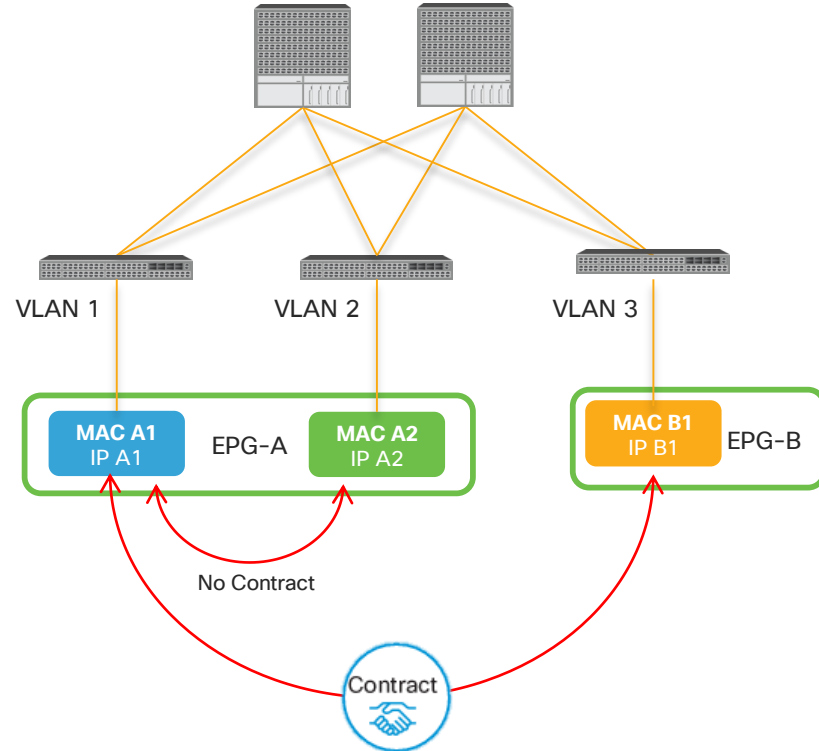
- **To implement traffic filtering**
 - Traffic within the same EPG doesn't get blocked
 - Traffic across EPGs always blocked without a contract
- A contract is applied between EPGs to allow traffic

What is a VLAN in ACI?

- VLAN is just an identifier to classify EPs to each EPG
- BD is the L2 domain instead of VLAN

What happens with forwarding?

- It will be done by BD or VRF to which EPG belongs



How to check End Points

From APIC GUI (Fabric perspective)

APIC GUI Screenshot: Tenant TK, EPG - EPG1-1, Client End-Points

End Point	MAC	IP	Learning Source	Hosting Server	Report Contrc Name	Interface	Multicast Address	Encap
EP-00:00:00:00:51:51	00:00:00:00:51:51	192.168.1.1	learned	---	---	Pod-1/Node-101/eth1/1 (learned)	---	vlan-51
EP-00:00:11:11:51:51	00:00:11:11:51:51	192.168.1.11	learned	---	---	Pod-1/Node-102/eth1/27 (learned)	---	vlan-51

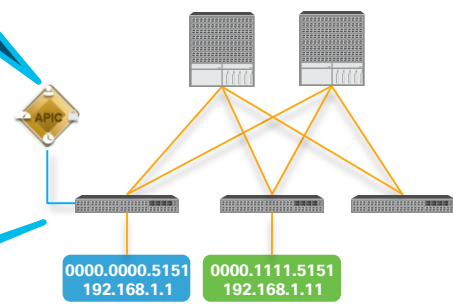
Fabric Wide Visibility shows where EPs are learned

From LEAF CLI (LEAF perspective)

```
leaf1# show endpoint vrf TK:VRF1
Legend:
s - arp          O - peer-attached    a - local-aged      S - static
V - vpc-attached p - peer-aged       M - span            L - local
B - bounce      H - vtep
```

VLAN/Domain	Encap VLAN	MAC Address IP Address	MAC Info/ IP Info	Interface
TK:VRF1		192.168.1.11		tunnel8
17/TK:VRF1	vxlan-15826915	0000.1111.5151		tunnel8
19	vlan-5	0000.0000.5151 L		eth1/1
TK:VRF1	vlan-5	192.168.0.51 L		eth1/1

Good for forwarding verification shows how EPs look from each LEAF



VLAN types in ACI

※ PI-VLAN : Platform Independent VLAN

VLAN ID for external devices
(user configured value)

Internal ID on LEAF
(not shared across LEAFs)

For forwarding
(global value for entire fabric)

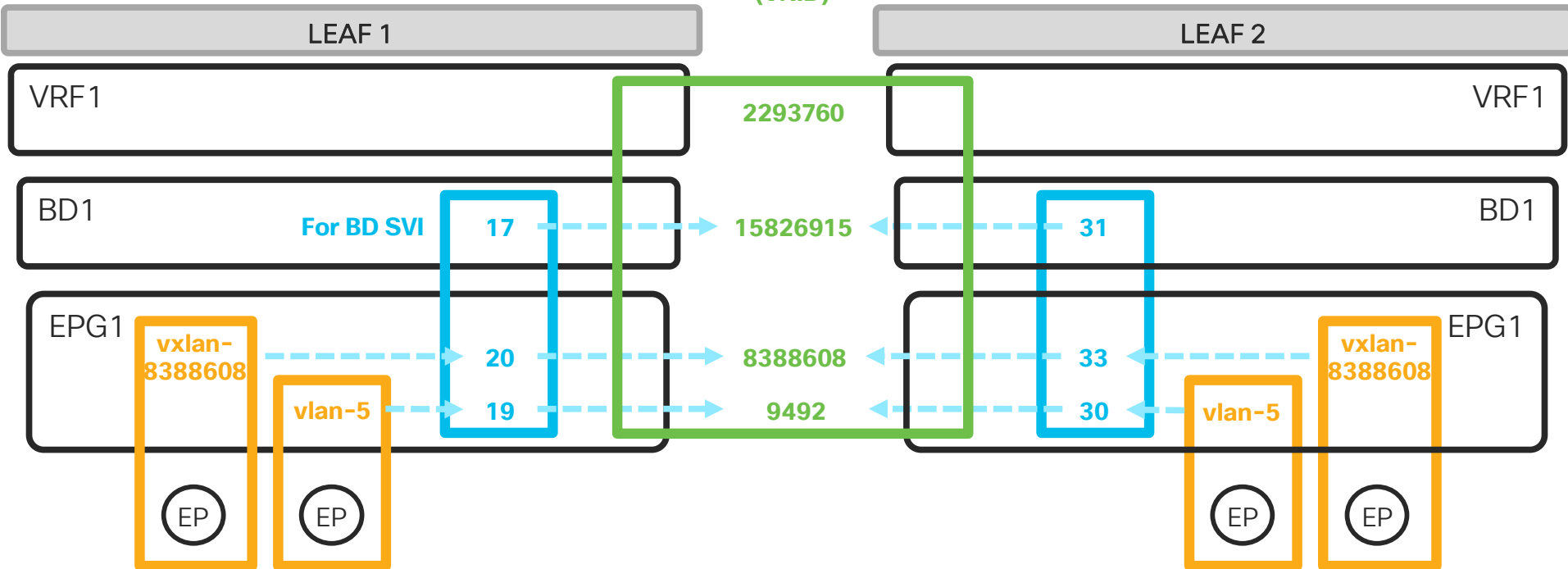
Access Encap VLAN

PI-VLAN

VxLAN ID
(VNID)

PI-VLAN

Access Encap VLAN



PI-VLAN for EPG and BD CLI



- Endpoint Table

```
leaf1# show endpoint ip 192.168.0.51
19 TK:VRF1 vlan-5 0000.5555.1111 L eth1/1
TK:VRF1 vlan-5 192.168.0.51 L eth1/1
```

PI-VLAN

Access Encap VLAN

- VLAN Table

NOT Access Encap VLAN.
PI-VLAN 17, 19

“extended” option to display Access Encap VLAN

```
leaf1# show vlan id 17,19 extended
VLAN Name Status Ports
-----
17 TK:BD1 active Eth1/1, Eth1/2, Po6
19 TK:AP1:EPG1 active Eth1/1

VLAN Type Vlan-mode Encap
-----
17 enet CE vxlan-15826915
19 enet CE vlan-5
```

PI-VLAN

Access Encap VLAN

PI-VLAN for EPG and BD CLI



```
leaf1# show vlan id 17,19 extended
```

VLAN Name	Status	Ports
17 TK:BD1	active	Eth1/1, Eth1/2, Po6
19 TK:AP1:EPG1	active	Eth1/1

VLAN Type	Vlan-mode	Encap
17 enet	CE	vxlan-15826915
19 enet	CE	vlan-5

EPG
PI-VLAN

```
leaf1# show system internal epm vlan 19
```

VLAN ID	Type	Access Encap (Type Value)	Fabric Encap	H/W id	BD VLAN	Endpoint Count	
19	FD vlan	802.1Q	5	8294	14	17	2

BD
PI-VLAN

How to check details of EndPoints



With MAC keyword :
show system internal epm endpoint mac 0000.5555.1111

```
leaf1# show system internal epm endpoint ip 192.168.0.51
```

```
MAC : 0000.5555.1111 ::: Num IPs : 1
IP# 0 : 192.168.0.51 ::: IP# 0 flags :
Vlan id : 19 ::: Vlan vnid : 9492 ::: VRF name : TK:VRF1
BD vnid : 15826915 ::: VRF vnid : 2293760
Phy If : 0x1a000000 ::: Tunnel If : 0
Interface : Ethernet1/1
Flags : 0x80004c04 ::: sclass : 49154 ::: Ref count : 5
EP Create Timestamp : 05/03/2017 09:33:56.654606
EP Update Timestamp : 05/04/2017 16:11:37.584734
EP Flags : local|IP|MAC|sclass|timer|
::::
```

PI-VLAN
for EPG

VNID for
EPG(Vlan), BD and VRF

Interface this EP is
learned on

End Point Types

Physical Local Endpoint (PL)

- An endpoint attached to this LEAF

```
fab1-leaf1# show endpoint ip 192.168.0.51
19                               vlan-5      0000.5555.1111 L          eth1/1
TK:VRF1                          vlan-5      192.168.0.51  L          eth1/1
```

Virtual Local Endpoint (VL)

- An endpoint on AVS/AVE attached to this LEAF

```
fab1-leaf1# show endpoint ip 192.168.66.2
14                               vxlan-8388608 0050.5680.34eb L          tunnel10
TK:VRF1                          vxlan-8388608 192.168.66.2  L          tunnel10
```

Remote Endpoint (Xr)

- An endpoint on a remote LEAF

```
fab1-leaf1# show endpoint mac 0000.5555.2222
17/TK:VRF1                      vxlan-15826915 0000.5555.2222          tunnel8
```

```
fab1-leaf1# show endpoint ip 192.168.0.52
TK:VRF1                          192.168.0.52          tunnel8
```

On-Peer Endpoint

- An endpoint connected to an orphan port on vPC peer

```
fab1-leaf1# show endpoint ip 192.168.0.52
19                               vlan-5      0000.5555.2222 O          tunnel8
TK:VRF1                          vlan-5      192.168.0.52  O          tunnel8
```

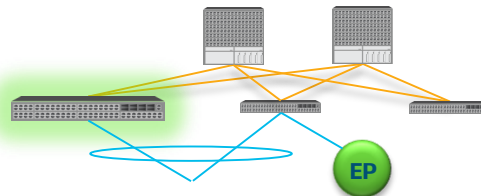
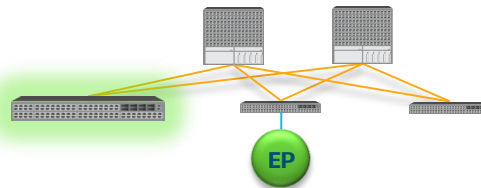
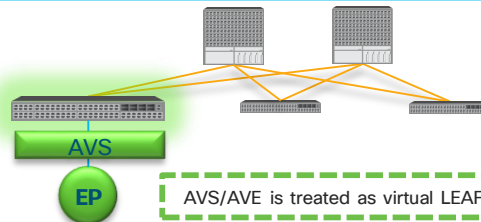
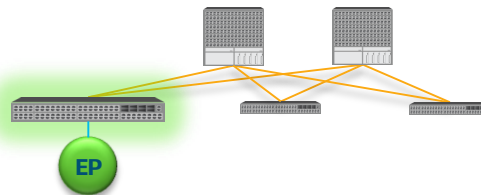
all commands on



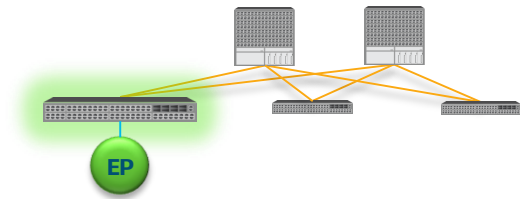
Legend:

O - peer-attached

L - local



End Point Learning (Local EP)



PI-VLAN ID(19) of EPG to which MAC belongs

```
leaf1# show endpoint ip 192.168.0.51
19                               vlan-5      0000.5555.1111 L      eth1/1
TK:VRF1                          vlan-5      192.168.0.51 L      eth1/1
```

VRF to which MAC & IP belong

Access Encap VLAN of EP(MAC+IP)

Local Endpoint (MAC)

A leaf learns **MAC A** as **local** if a packet with **src MAC A** comes in from its **front panel port**.

Local Endpoint (/32 host IP)

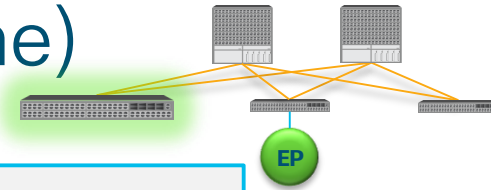
A leaf learns **IP A /32** as **local**

- if a packet with **src IP A** comes in from its **front panel port AND IP lookup** is done on ACI. (which means IP addr is learned **only when** a leaf handles **L3 traffic**)
or
- if **ARP request** with **sender IP A** comes in from its **front panel port**. (regardless of ARP Flooding setup)

EPG/BD/VRF is based on Access Encap VLAN ID

What APIC GUI shows are these local Endpoints

End Point Learning (Remote EP = cache)



PI-VLAN(17) of BD and VRF to which MAC belongs

```
fab1-leaf1# show endpoint mac 0000.5555.2222
17/TK:VRF1          vxlan-15826915    0000.5555.2222    tunnel8
```

BD VNID

VRF to which MAC & IP belong

```
fab1-leaf1# show endpoint ip 192.168.0.52
TK:VRF1            192.168.0.52    tunnel8
```

tunnel represents destination leaf TEP

Remote Endpoint (MAC)

A leaf learns **MAC A** as **remote** when **L2 traffic** with **src MAC A** comes in from **SPINE**.

Remote Endpoint (/32 host IP)

A leaf learns **IP A** as **remote** when **L3 traffic** with **src IP A** comes in from **SPINE**.

- Remote MAC and remote IP is learned separately
- BD(for MAC) / VRF(for IP) is based on VNID in VxLAN header

VNID is
BD when L2 traffic
VRF when L3 traffic (not both)



How to check Tunnel Interface (TEP)



```
leaf1# show int tunnel 8 | grep Tun
Tunnel8 is up
Tunnel protocol/transport is ivxlan
Tunnel source 11.0.200.92/32 (lo0)
Tunnel destination 11.0.48.95
```

TEP IP address

```
leaf1# acidiag fmvread
```

ID	Pod ID	Name	Serial Number	IP Address	Role	State	LastUpdMsgId
101	1	leaf1	FDO20160AAA	11.0.200.92/32	leaf	active	0
102	1	leaf2	FDO20160BBB	11.0.48.95/32	leaf	active	0
103	1	leaf3	FDO20240CCC	11.0.200.91/32	leaf	active	0
201	1	spine1	FGE12345678	11.0.200.94/32	spine	active	0
202	1	spine2	FGE87654321	11.0.200.93/32	spine	active	0

```
admin@apic1:~> moquery -c vpcDom | egrep 'virtualIp|dn|#'
# vpc.Dom
dn      : topology/pod-1/node-101/sys/vpc/inst/dom-1
virtualIp : 11.0.64.65/32
# vpc.Dom
dn      : topology/pod-1/node-102/sys/vpc/inst/...
virtualIp : 11.0.64.65/32
# vpc.Dom
dn      : topology/pod-1/node-103/sys/vpc/inst/dom-2
virtualIp : 11.0.192.64/32
# vpc.Dom
dn      : topology/pod-1/node-104/sys/vpc/inst/dom-2
virtualIp : 11.0.192.64/32
```

Tunnel may point to vPC vTEP
This is vTEP for vPC LEAF 101-102

COOP (End Point Learning on Spine)

SPINEs do NOT learn EP from data plane like LEAF

SPINEs receive all EP data from Leafs

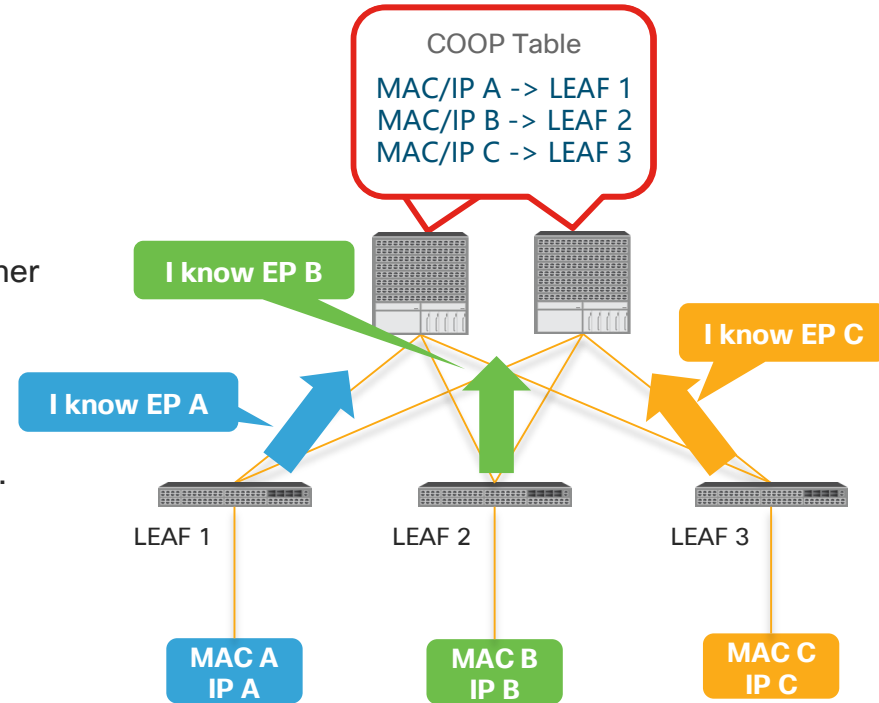
1. LEAF learns EP (either MAC or/and IP) as **local**
2. LEAF reports local EP to Spine via COOP process
3. SPINE stores these in COOP DB and synchronize with other SPINEs

What is the purpose of COOP?

When Leaf doesn't know dst EP, LEAF can forward packet to Spine in order to let Spine decide where to send. This behavior is called **Spine-Proxy**.

Note :

- Normally SPINE doesn't push COOP DB entries to each LEAF. It just receives and stores. The exception is for bounce entries.
- Remote Endpoints are stored on each Leaf nodes as cache. This is not reported to Spine COOP.



How to check COOP DB on Spine



```
fab5-spine2# show coop internal info repo ep key 15826915 0000.5555.1111 | egrep 'vnid|mac|id|Real'
```

```
EP bd vnid : 15826915  
EP mac : 00:00:55:55:11:11  
Vrf vnid : 2293760
```

EP data

BD VNID

```
Epg vnid : 0  
Ep vpc-id : 0  
Ep vpc virtual switch-id : 0.0.0.
```

TEP address of LEAF

```
publisher id : 11.0.200.92
```

```
Real IPv4 EP : 192.168.5.111
```

EP data

```
fab5-spine2# show coop internal info ip-db key 2293760 192.168.5.111
```

```
IP address : 192.168.5.111  
Vrf : 2293760  
Flags : 0  
EP bd vnid : 15826915  
EP mac : 00:00:55:55:11:11  
Publisher Id : 11.0.200.92
```

EP data

VRF VNID

TEP address of LEAF

```
---- snip ----
```

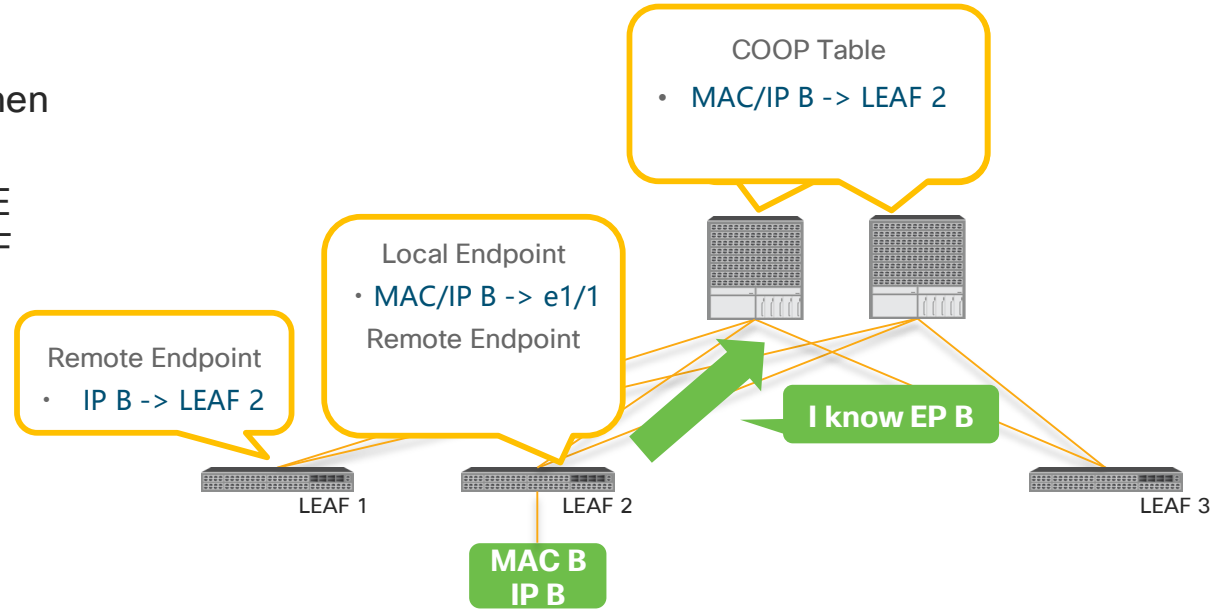
Bounce Entry

What is Bounce Entry?

- Remote EPs created by COOP when an EP moved
 - The exception where SPINE pushes COOP DB to a LEAF

What is Bounce Entry for?

- An old LEAF (LEAF2) can bounce packets to the latest EP location in case other LEAFs (LEAF1) with old Remote EPs still send packets to the old LEAF (LEAF2)



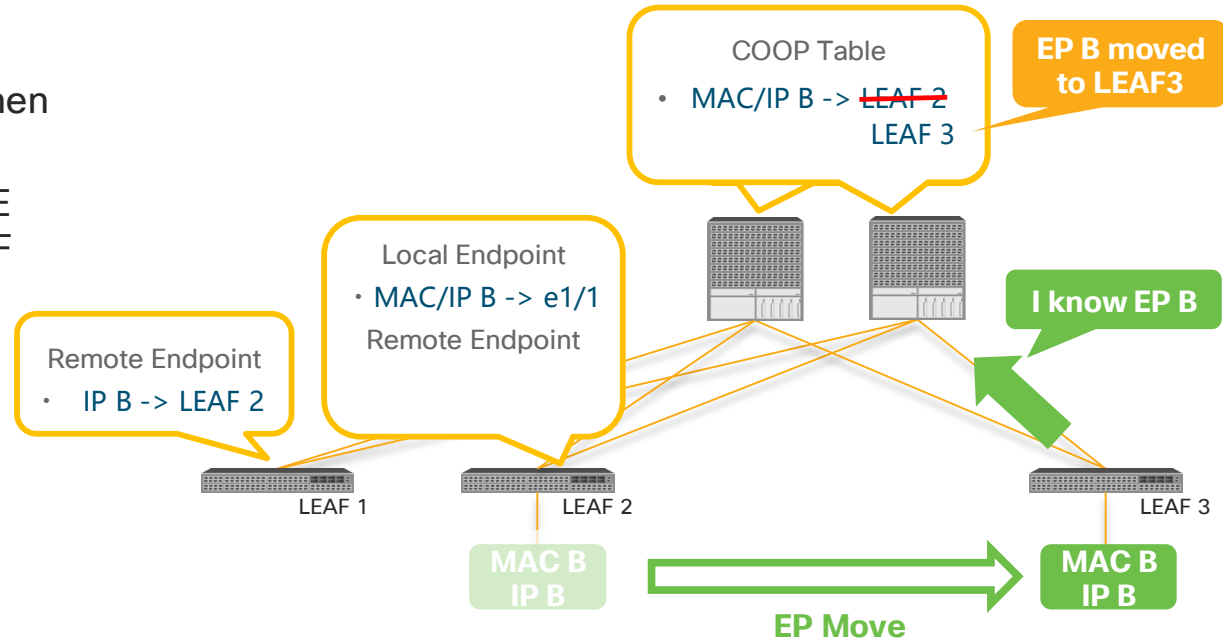
Bounce Entry

What is Bounce Entry?

- Remote EPs created by COOP when an EP moved
 - The exception where SPINE pushes COOP DB to a LEAF

What is Bounce Entry for?

- An old LEAF (LEAF2) can bounce packets to the latest EP location in case other LEAFs (LEAF1) with old Remote EPs still send packets to the old LEAF (LEAF2)



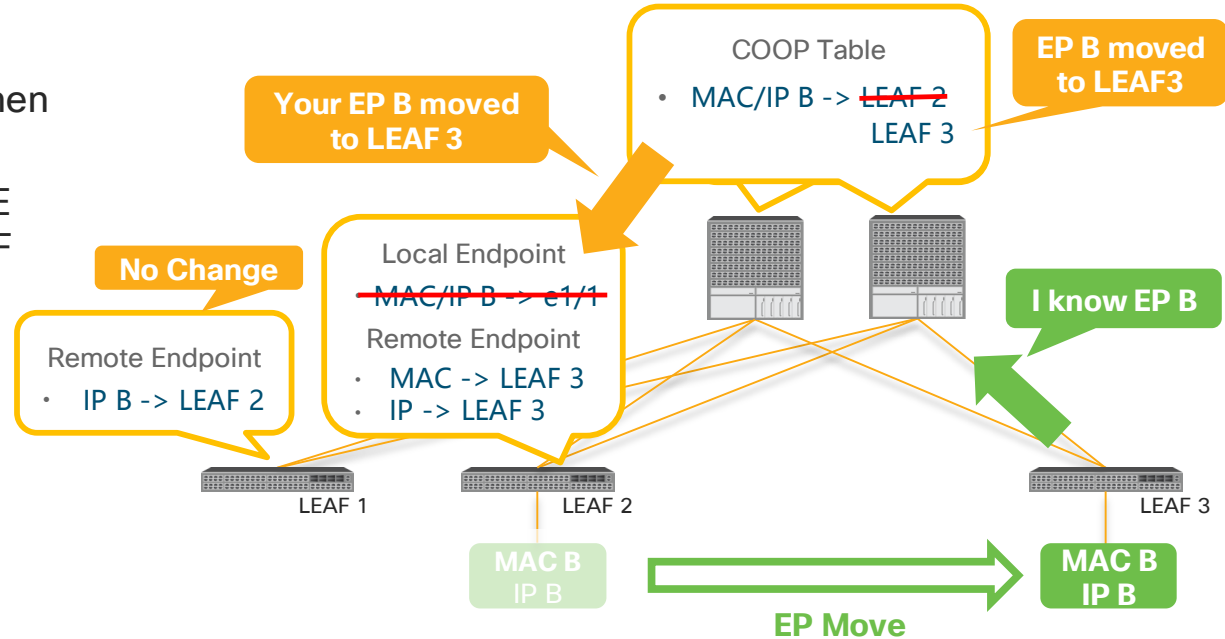
Bounce Entry

What is Bounce Entry?

- Remote EPs created by COOP when an EP moved
 - The exception where SPINE pushes COOP DB to a LEAF

What is Bounce Entry for?

- An old LEAF (LEAF2) can bounce packets to the latest EP location in case other LEAFs (LEAF1) with old Remote EPs still send packets to the old LEAF (LEAF2)



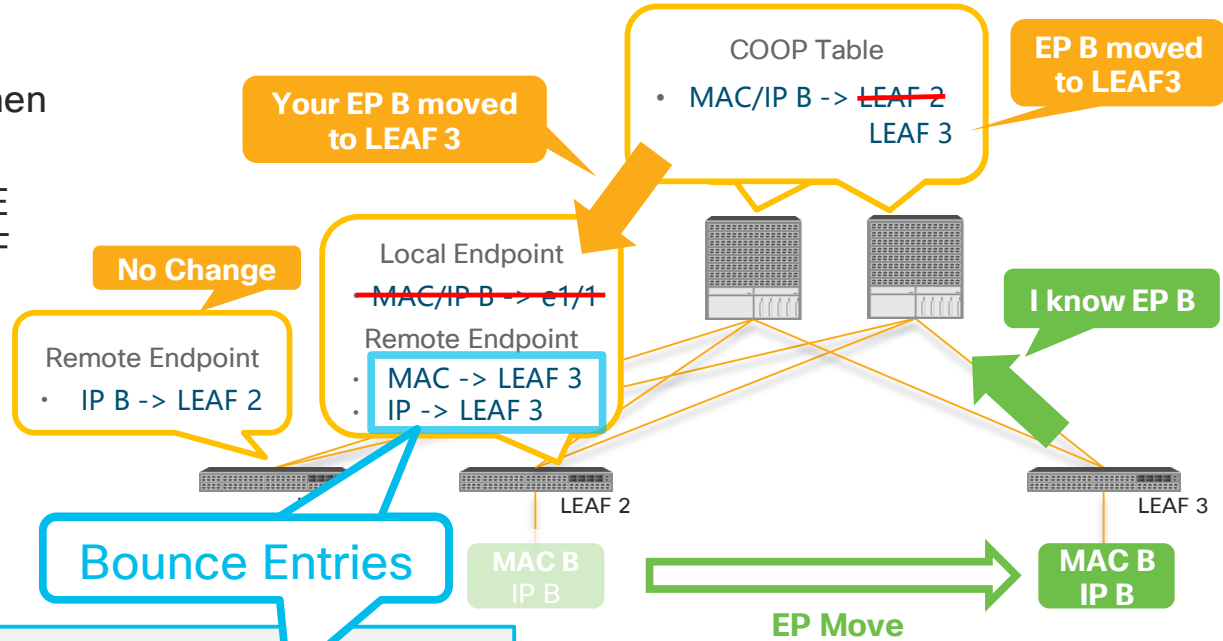
Bounce Entry

What is Bounce Entry?

- Remote EPs created by COOP when an EP moved
 - The exception where SPINE pushes COOP DB to a LEAF

What is Bounce Entry for?

- An old LEAF (LEAF2) can bounce packets to the latest EP location in case other LEAFs (LEAF1) with old Remote EPs still send packets to the old LEAF (LEAF2)



```
leaf2# show end vrf TK:VRF1

TK:VRF1          192.168.1.11 B      tunnel4
60/TK:VRF1      vxlan-15826915 0000.1111.5151 B      tunnel4
```

Agenda

- Introduction
 - ACI Overlay VxLAN and TEP
- ACI Forwarding components
 - Endpoints, EPG, EP Learning, COOP and How it all works
 - BD, VRF forwarding scope and detailed options
 - Spine-Proxy and ARP Glean
 - Forwarding Software Architecture and ASIC Generation
- ACI Packet Walk
 - Walk through the life of a packet going through ACI

ACI Forwarding Component 2

- Pervasive Gateway (BD SVI)
- Forwarding Scope (VRF or BD)
- Forwarding mode in BD

Pervasive Gateway(BD SVI)

Tenant TK

- Quick Start
- Tenant TK
 - Application Profiles
 - Networking
 - Bridge Domains
 - BD1
 - DHCP Relay Labels
 - L4-L7 Service Parameters
 - Subnets**
 - 192.168.0.254/24
 - 192.168.1.254/24
 - ND Proxy Subnets
 - BD2
 - BD3
 - BD_SG_PBR1

Subnet - 192.168.0.254/24

Properties

IP Address: 192.168.0.254/24

Description: optional

Treat as virtual IP address:

Make this IP address primary:

Scope: Private to VRF
 Advertised Externally
 Shared between VRFs

Subnet Control:
 No Default SVI Gateway
 Querier IP

L3 Out for Route Profile: select a value

Route Profile: select a value

What is pervasive GW for?

- To be a default GW for EPs in the Fabric
 - All EPs can have consistent gateway IP address one hop away
- To represent subnets(IP ranges) for a BD
 - ACI knows which BD may have potential hidden/silent EPs

How is pervasive GW deployed?

- Installed as an SVI on LEAFs
 - PI-VLAN for BD is used to represent a pervasive GW SVI
 - A pervasive SVI has secondary IP when multiple pervasive GWs are configured on the same BD
 - User can choose a primary address

```
leaf1# show ip route vrf TK:VRF1
```

```
192.168.0.0/24, ubest/mbest: 1/0, attached, direct, pervasive  
*via 10.0.184.64%overlay-1, [1/0], 04:32:16, static
```

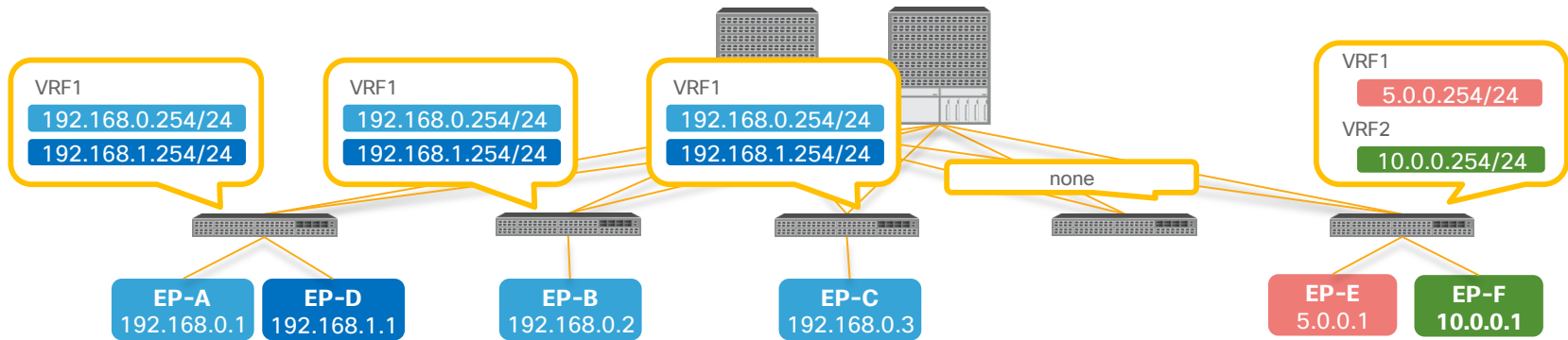
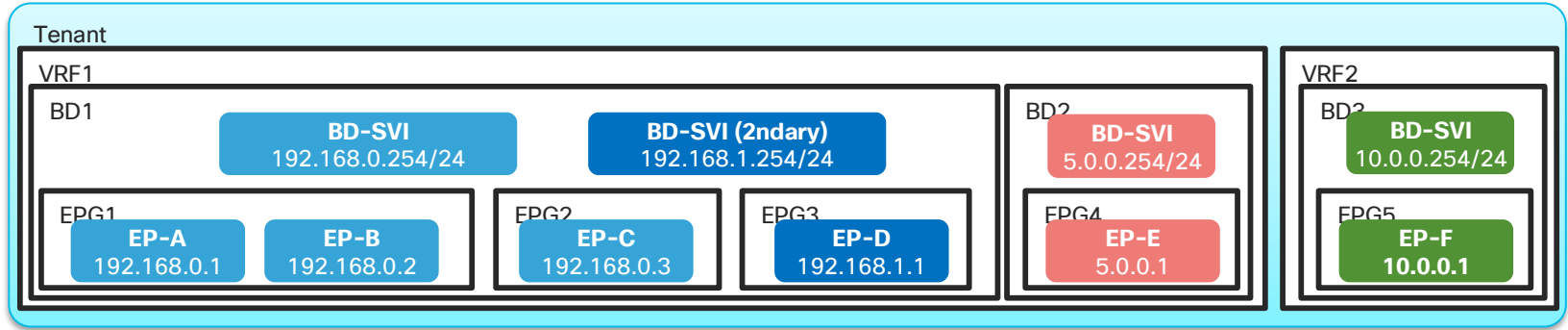
```
192.168.0.254/32, ubest/mbest: 1/0, attached  
*via 192.168.0.254, vlan10, [1/0], 04:32:16, local, local
```

Pervasive route

Pervasive SVI

BD SVI with PI-VLAN

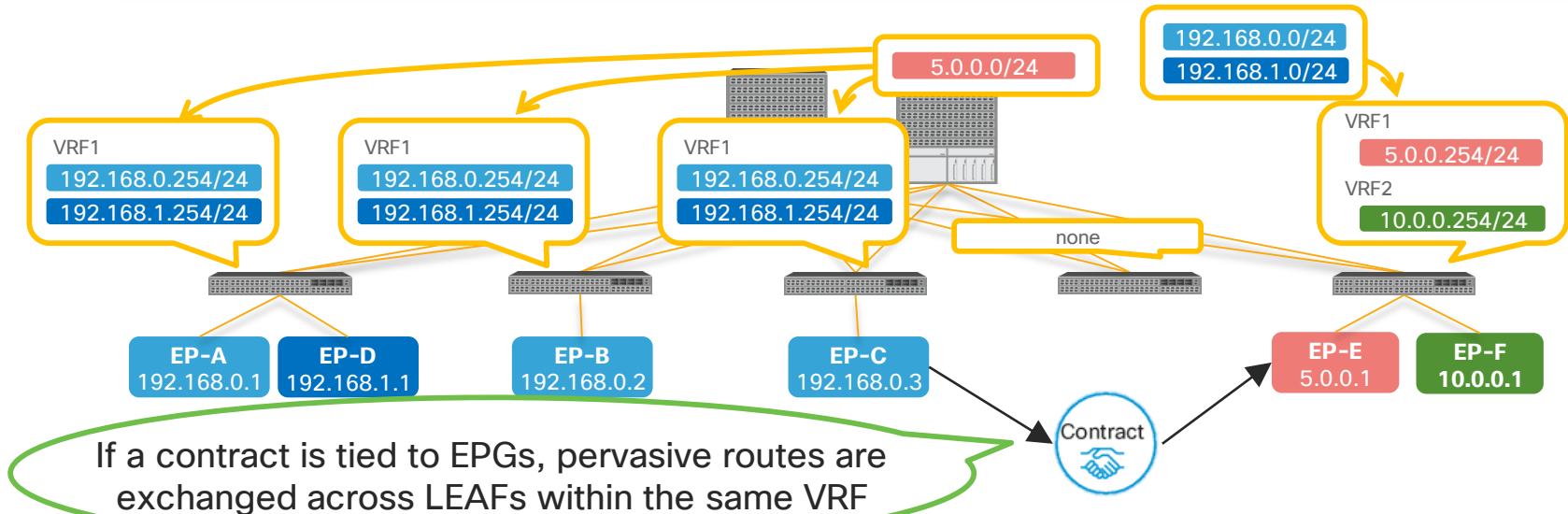
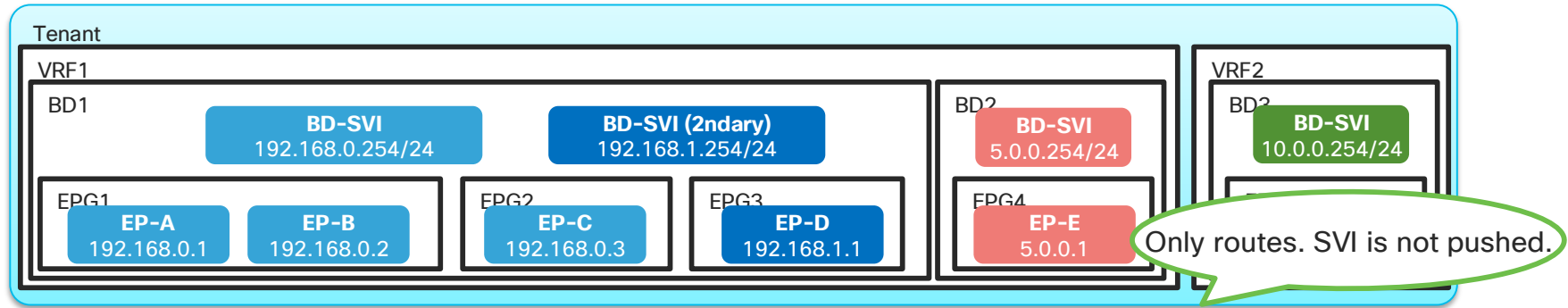
Pervasive Gateway(BD SVI) example



Both pervasive route and SVI

If LEAF has EPG for BD,
Pervasive GW will be programmed

Pervasive Gateway(BD SVI) example



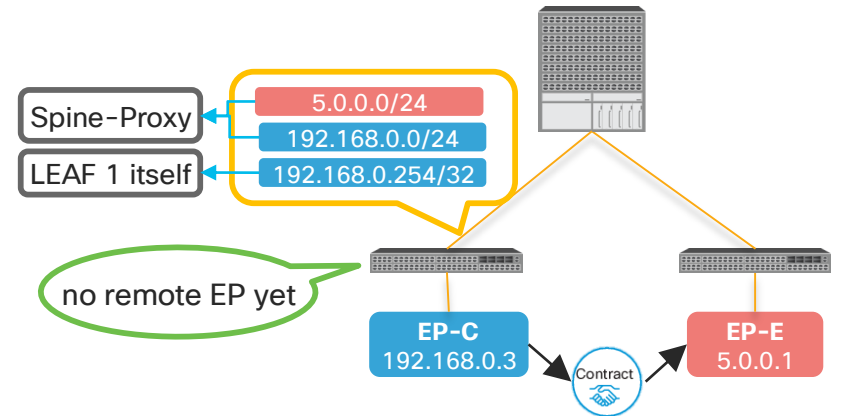
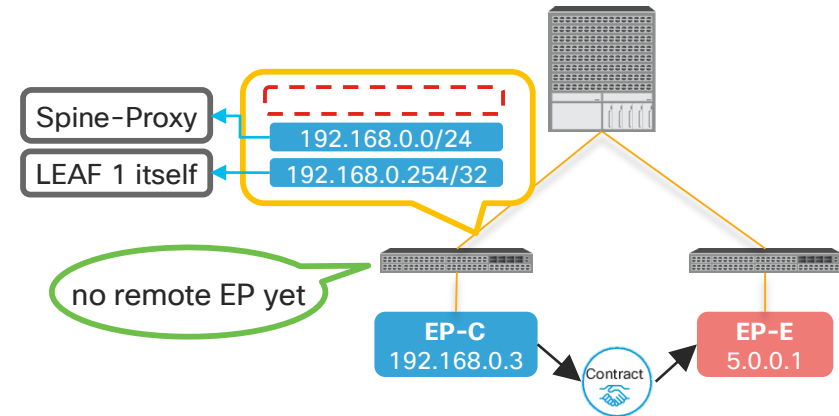
Pervasive Gateway(BD SVI) cont.

Why does ACI push pervasive routes to other LEAFs after a contract?

➤ Pervasive routes are required for Spine-Proxy

❌ what if no pervasive route, no remote EP?

✅ with pervasive route and no remote EP?



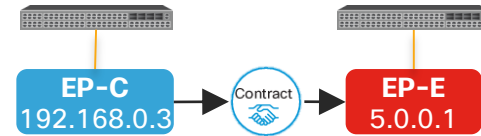
No Spine-Proxy for 5.0.0.1

It may be either dropped or forwarded to L3OUT if a default route exists

Spine-Proxy for 5.0.0.1

With the contract, ACI knows the LEAF needs to reach out to 5.0.0.0/24

Pervasive Gateway



Without contract

```
L101# show ip route vrf TK:VRF1
192.168.0.0/24, ubest/mbest: 1/0, attached, direct, pervasive
  *via 10.0.184.64%overlay-1, [1/0], 04:32:16, static
192.168.0.254/32, ubest/mbest: 1/0, attached
  *via 192.168.0.254, vlan10, [1/0], 04:32:16, local, local

L103# show ip route vrf TK:VRF1
5.0.0.0/24, ubest/mbest: 1/0, attached, direct, pervasive
  *via 10.0.184.64%overlay-1, [1/0], 00:00:06, static
5.0.0.254/32, ubest/mbest: 1/0, attached
  *via 192.168.2.254, vlan13, [1/0], 00:00:06, local, local
```

With contract

```
L101# show ip route vrf TK:VRF1
192.168.0.0/24, ubest/mbest: 1/0, attached, direct, pervasive
  *via 10.0.184.64%overlay-1, [1/0], 04:32:27, static
192.168.0.254/32, ubest/mbest: 1/0, attached
  *via 192.168.0.254, vlan10, [1/0], 04:32:27, local, local
5.0.0.0/24, ubest/mbest: 1/0, attached, direct, pervasive
  *via 10.0.184.64%overlay-1, [1/0], 00:00:02, static

L103# show ip route vrf TK:VRF1
192.168.0.0/24, ubest/mbest: 1/0, attached, direct, pervasive
  *via 10.0.184.64%overlay-1, [1/0], 00:00:10, static
5.0.0.0/24, ubest/mbest: 1/0, attached, direct, pervasive
  *via 10.0.184.64%overlay-1, [1/0], 00:00:32, static
5.0.0.254/32, ubest/mbest: 1/0, attached
  *via 192.168.2.254, vlan13, [1/0], 00:00:32, local, local
```

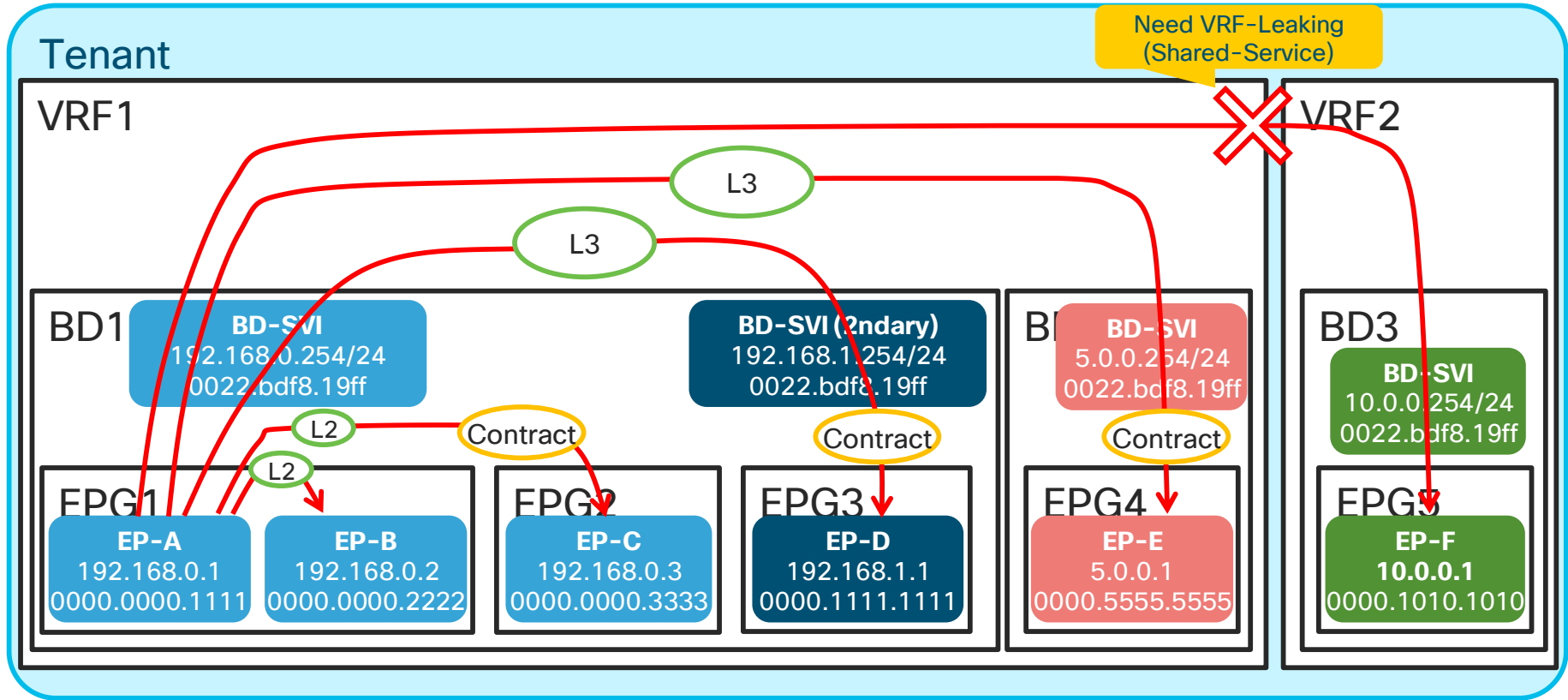
Exchange pervasive route

➤ Pervasive routes are pushed to other LEAFs with contracts

ACI Forwarding Component 2

- Pervasive Gateway (BD SVI)
- **Forwarding Scope (VRF or BD)**
- Forwarding mode in BD

ACI Forwarding Scope Concepts



Forwarding Scope

Tenant

Basic L2/L3 lookup is same as legacy switch
IP Lookup will be done with VRF scope even though subnets are configured under BD

VRF1

scope : VRF-VNID

```
192.168.0.1 -> EP-A      5.0.0.1      -> EP-E
192.168.0.2 -> EP-B      192.168.0.254 -> BD1 SVI
192.168.0.3 -> EP-C      192.168.1.254 -> BD1 SVI
192.168.1.1 -> EP-D      5.0.0.254    -> BD2 SVI
```

scope : BD-VNID

BD1

```
0000.0000.1111 -> EP-A      0000.1111.1111 -> EP-D
0000.0000.2222 -> EP-B      0022.bdf8.19ff -> BD SVI
0000.0000.3333 -> EP-C
```

scope : BD-VNID

BD2

```
0000.5555.5555 -> EP-E
0022.bdf8.19ff -> BD SVI
```

EPG1

EP-A
192.168.0.1
0000.0000.1111

EP-B
192.168.0.2
0000.0000.2222

EPG2

EP-C
192.168.0.3
0000.0000.3333

EPG3

EP-D
192.168.1.1
0000.1111.1111

EPG4

EP-E
5.0.0.1
0000.5555.5555

Forwarding Scope

L2 traffic(=same subnet) use only MAC
hence BD lookup only

Tenant

VRF1

scope : VRF-VNID

```
192.168.0.1 -> EP-A      5.0.0.1      -> EP-E
192.168.0.2 -> EP-B      192.168.0.254 -> BD1 SVI
192.168.0.3 -> EP-C      192.168.1.254 -> BD1 SVI
192.168.1.1 -> EP-D      5.0.0.254    -> BD2 SVI
```

BD1

scope : BD-VNID

```
0000.0000.1111 -> EP-A      0000.1111.1111 -> EP-D
0000.0000.2222 -> EP-B      0022.bdf8.19ff -> BD SVI
0000.0000.3333 -> EP-C
```

EPG1

EP-A

192.168.0.1

0000.0000.1111

EP-B

192.168.0.2

0000.0000.2222

EPG2

EP-C

192.168.0.3

0000.0000.3333

EPG3

EP-D

192.168.1.1

0000.1111.1111

EPG4

EP-E

5.0.0.1

0000.5555.5555

BD2

scope : BD-VNID

```
0000.5555.5555 -> EP-E
0022.bdf8.19ff -> BD SVI
```

Forwarding Scope

It's same even if EPG is different

Tenant

VRF1

scope : VRF-VNID

```
192.168.0.1 -> EP-A      5.0.0.1      -> EP-E
192.168.0.2 -> EP-B      192.168.0.254 -> BD1 SVI
192.168.0.3 -> EP-C      192.168.1.254 -> BD1 SVI
192.168.1.1 -> EP-D      5.0.0.254    -> BD2 SVI
```

BD1

scope : BD-VNID

```
0000.0000.1111 -> EP-A      0000.1111.1111 -> EP-D
0000.0000.2222 -> EP-B      0022.bdf8.19ff -> BD SVI
0000.0000.3333 -> EP-C
```

BD2

scope : BD-VNID

```
0000.5555.5555 -> EP-E
0022.bdf8.19ff -> BD SVI
```

EPG1

EP-A

192.168.0.1

0000.0000.1111

EP-B

192.168.0.2

0000.0000.2222

EPG2

EP-C

192.168.0.3

0000.0000.3333

EPG3

EP-D

192.168.1.1

0000.1111.1111

EPG4

EP-E

5.0.0.1

0000.5555.5555

Forwarding Scope

L3 traffic(=different subnet) use IP Lookup

1. Dst MAC hits default gw svi mac
2. IP Lookup in VRF
even though EPs are in the same BD

Tenant

VRF1

scope : VRF-VNID

```
192.168.0.1 -> EP-A      5.0.0.1      -> EP-E
192.168.0.2 -> EP-B      192.168.0.254 -> BD1 SVI
192.168.0.3 -> EP-C      192.168.1.254 -> BD1 SVI
192.168.1.1 -> EP-D    5.0.0.254    -> BD2 SVI
```

scope : BD-VNID

BD1

```
0000.0000.1111 -> EP-A      0000.1111.1111 -> EP-D
0000.0000.2222 -> EP-B      0022.bdf8.19ff -> BD SVI
0000.0000.3333 -> EP-C
```

scope : BD-VNID

BD2

```
0000.5555.5555 -> EP-E
0022.bdf8.19ff -> BD SVI
```

FPG1

EP-A

192.168.0.1

0000.0000.1111

EP-B

192.168.0.2

0000.0000.2222

FPG2

EP-C

192.168.0.3

0000.0000.3333

FPG3

EP-D

192.168.1.1

0000.1111.1111

EPG4

EP-E

5.0.0.1

0000.5555.5555

EP-A (192.168.0.1) -> EP-D (192.168.1.1)

Forwarding Scope

It's same even if BD is different

Tenant

VRF1

scope : VRF-VNID

```
192.168.0.1 -> EP-A      5.0.0.1      -> EP-E
192.168.0.2 -> EP-B      192.168.0.254 -> BD1 SVI
192.168.0.3 -> EP-C      192.168.1.254 -> BD1 SVI
192.168.1.1 -> EP-D      5.0.0.254    -> BD2 SVI
```

BD1

scope : BD-VNID

```
0000.0000.1111 -> EP-A      0000.1111.1111 -> EP-D
0000.0000.2222 -> EP-B      0022.bdf8.19ff -> BD SVI
0000.0000.3333 -> EP-C
```

BD2

scope : BD-VNID

```
0000.5555.5555 -> EP-E
0022.bdf8.19ff -> BD SVI
```

EPG1

EP-A

192.168.0.1

0000.0000.1111

EP-B

192.168.0.2

0000.0000.2222

EPG2

EP-C

192.168.0.3

0000.0000.3333

EPG3

EP-D

192.168.1.1

0000.1111.1111

EPG4

EP-E

5.0.0.1

0000.5555.5555

EP-A (192.168.0.1) -> EP-E (5.0.0.1)

ACI Forwarding Component 2

- Pervasive Gateway (BD SVI)
- Forwarding Scope (VRF or BD)
- **Forwarding mode in BD**

ACI BD Forwarding Option

Tenant TK

Bridge Domain - BD1

Policy Operational Stats Health Faults History

General **L3 Configurations** Advanced/Troubleshooting

100

Properties

Name: BD1

Alias:

Description: optional

Type: fc **regular**

Global Alias:

Legacy Mode: No

VRF: VRF1

Resolved VRF: TK/VRF1

L2 Unknown Unicast: Flood **Hardware Proxy**

L3 Unknown Multicast Flooding: Flood **Optimized Flood**

Multi Destination Flooding: **Flood in BD** Drop Flood in Encapsulation

PIM:

IGMP Policy: select an option

ARP Flooding:

Endpoint Dataplane Learning:

Limit IP Learning To Subnet:

End Point Retention Policy: select a value

IGMP Snoop Policy: select a value

Unicast Routing:

Operational Value for Unicast Routing: true

Custom MAC Address: 00:22:BD:F8:19:FF

Virtual MAC Address: Not Configured

- Unicast Routing
- L2 Unknown Unicast
- L3 Unknown Multicast Flooding
- Multi Destination Flooding
- ARP Flooding

Properties

Unicast Routing:

Operational Value for Unicast Routing: true

Custom MAC Address: 00:22:BD:F8:19:FF

Virtual MAC Address: Not Configured

※ Please check a whitepaper “ACI Fabric EP Learning” for EP learning options
<https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-739989.html>

Unicast Routing

Unicast Routing:

On

Tenant

VRF1

scope : VRF-VNID

```
192.168.0.1 -> EP-A      5.0.0.1      -> EP-E
192.168.0.2 -> EP-B      192.168.0.254 -> BD1 SVI
192.168.0.3 -> EP-C      192.168.1.254 -> BD1 SVI
192.168.1.1 -> EP-D      5.0.0.254   -> BD2 SVI
```

Unicast Routing : OFF

Unicast Routing : ON

scope : BD-VNID

BD1

```
0000.0000.1111 -> EP-A      0000.1111.1111 -> EP-D
0000.0000.2222 -> EP-B      0022.bdf0.19ff -> BD SVI
0000.0000.3333 -> EP-C
```

scope : BD-VNID

BD2

```
0000.5555.5555 -> EP-E
0022.bdf8.19ff -> BD SVI
```

EPG1

EP-A

```
192.168.0.1
0000.0000.1111
```

EP-B

```
192.168.0.2
0000.0000.2222
```

EPG2

EP-C

```
192.168.0.3
0000.0000.3333
```

EPG3

EP-D

```
192.168.1.1
0000.1111.1111
```

EPG4

EP-E

```
5.0.0.1
0000.5555.5555
```

If Unicast Routing is disabled, (BD1 in above)

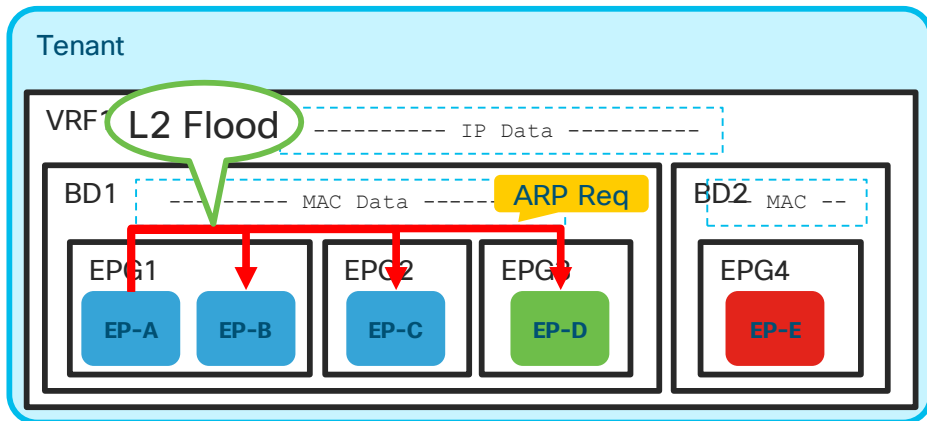
- IP Learning is disabled on BD
 - BD SVI is disabled
- => Only L2 Forwarding is available

In above example :

- EP-A <-> EP-B : GOOD (L2 forwarding)
- EP-A <-> EP-C : GOOD (L2 forwarding)
- EP-A <-> EP-D : FAIL (L3 forwarding)
- EP-A <-> EP-E : FAIL (L3 forwarding)

ARP Flooding

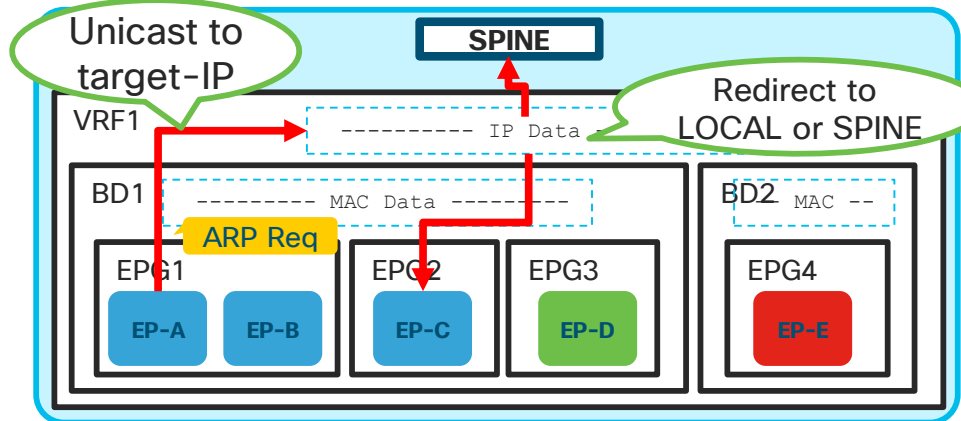
• ARP Flood On



Always **flood** ARP Request **within the same BD**

- Flood as broadcast if DST-MAC is FFFF.FFFF.FFFF
- Flood to other Leaf switches through Spine
- EP IP Data is not used for forwarding but still Sender-IP is learned if Unicast Routing is enabled.
- Good option when BD is supposed to be pure L2 without Unicast Routing like legacy VLAN

• ARP Flood Off (= Spine-Proxy)

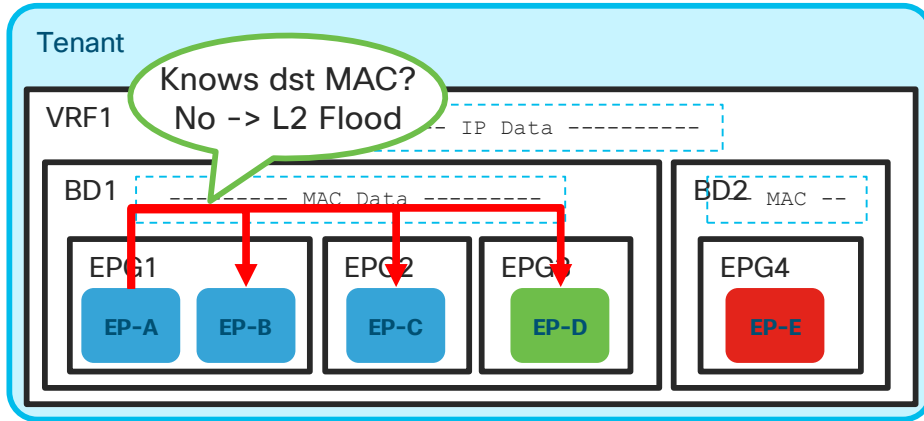


ARP Request is handled as **L3 Unicast** with Target-IP

- If IP is **learned** on ingress Leaf,
 - Ingress Leaf forwards ARP Req **directly to dest**
- If IP is **not learned** on ingress Leaf,
 - Ingress Leaf forwards ARP Req **to Spine** **Spine-Proxy**
Spine will forward it to Leaf on which DstIP resides
- If IP is **not learned** even on **Spine**,
 - Drop and **ARP Clean** (only within BD)

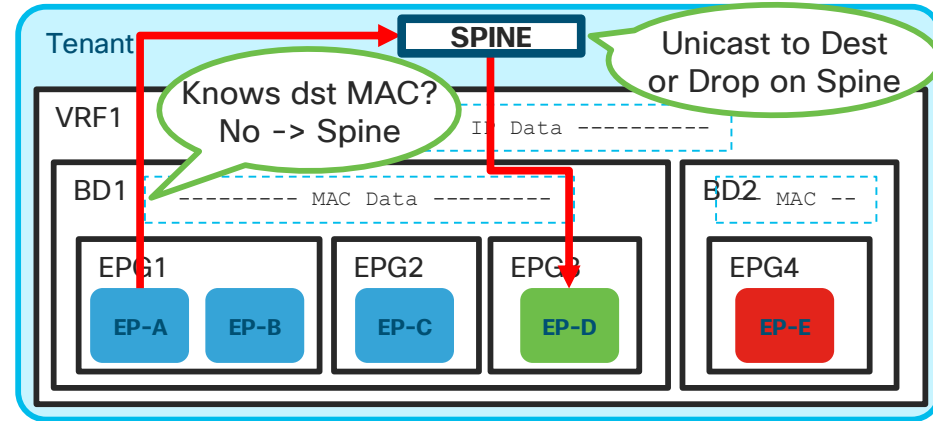
L2 Unknown Unicast

- Flood



L2 Unknown Unicast: Flood Hardware Proxy

- Hardware Proxy (= Spine-Proxy)



Always **flood** L2 Unknown Unicast **within the same BD**

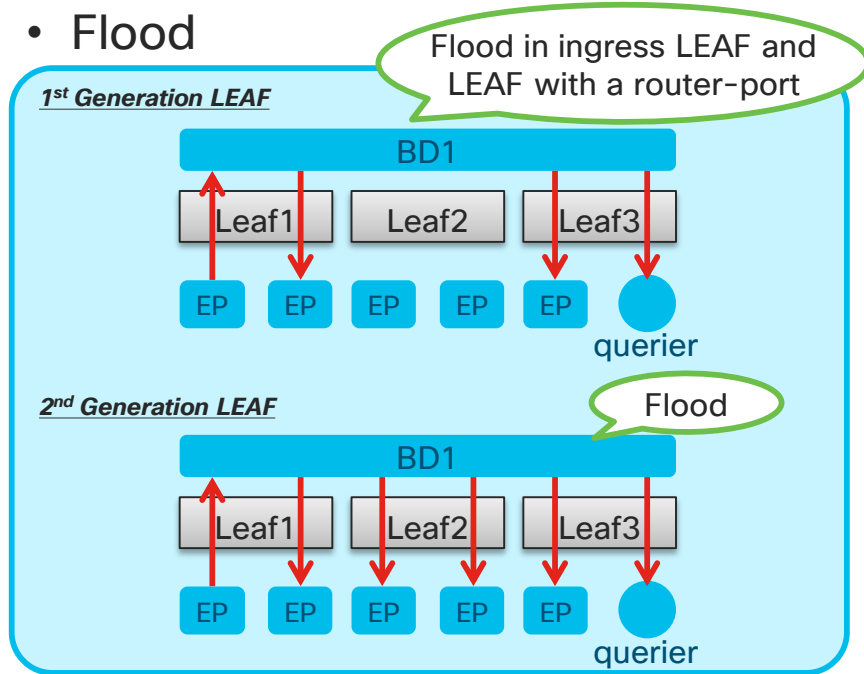
- Flood as well as legacy VLAN.
- Flood happens locally and on other Leaf switches.
- Good option when BD is supposed to be pure L2 without Unicast Routing as in legacy VLAN
- Good option when there are silent L2 hosts

L2 Unknown Unicast is sent to Spine-Proxy

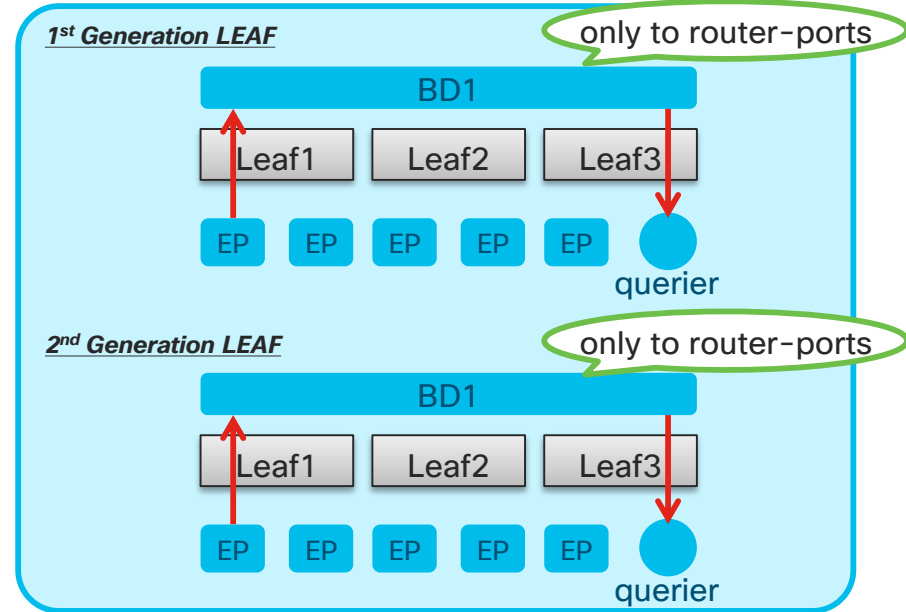
- If DST-MAC is **learned** on **Spine**,
 - Spine forwards it **directly to dest Leaf**
- If DST-MAC is **not learned** even on **Spine**
 - Drop

L3 Unknown Multicast Flooding

- Flood



- OMF (Optimized Multicast Flood)



L3 Unknown Multicast = IP multicast group unknown to LEAF IGMP snooping
➤ Controls flooding **unknown** IGMP snooping groups

Multi Destination Flooding

Multi Destination Flooding: Flood in BD Drop Flood in Encapsulation

This mode does not apply to OSPF/OSPFv6, BGP, EIGRP, CDP, LACP, LLDP, ISIS, IGMP, PIM, ST-BPDU, ARP/GARP, RARP, ND

Flooding mode for L2 multicast, Broadcast and link-local

- Flood in BD**

Flood within the same BD regardless of EPG or VLAN.

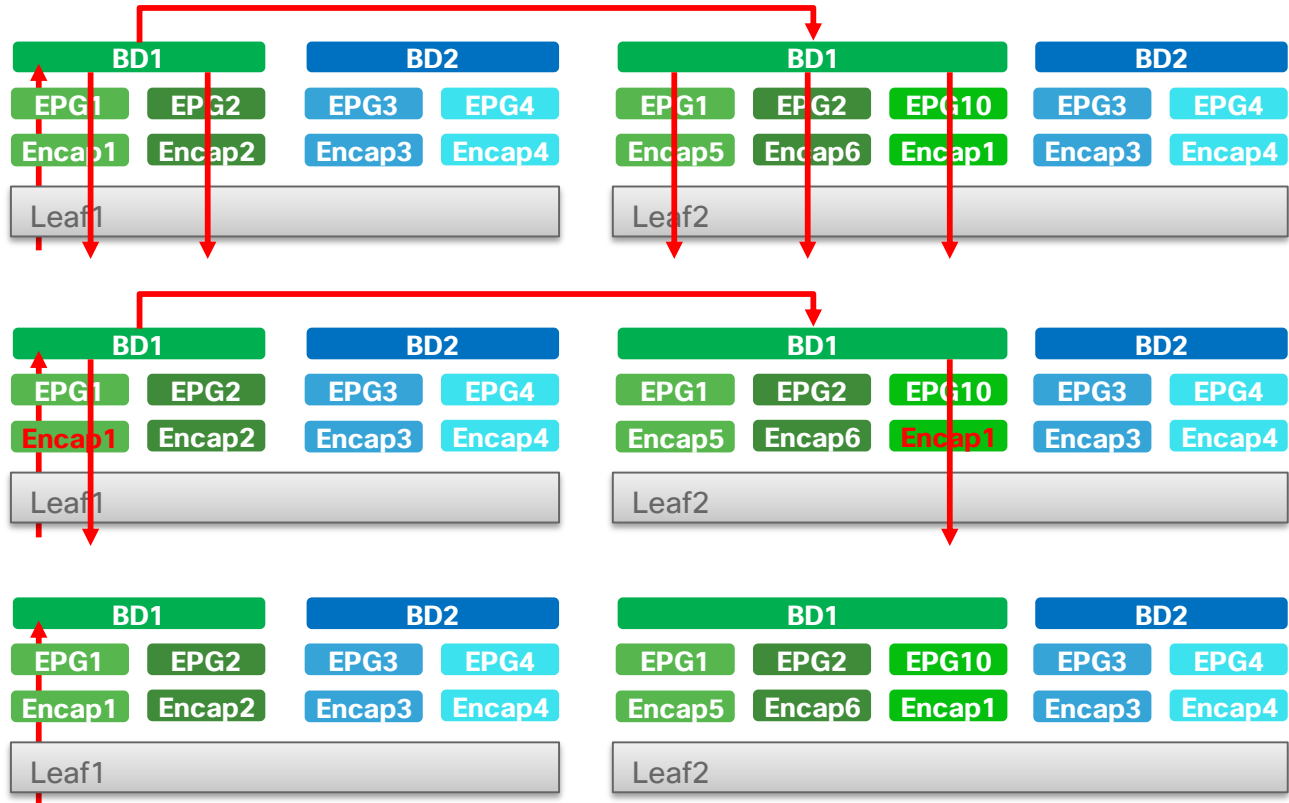
Behavior change from 3.1

- Flood in Encapsulation**

Flood within the same access encap VLAN and BD regardless of EPG.

- Drop**


No Flood. Just drop.



Flood in Encapsulation

BD Multi Destination Flooding: Flood in BD Drop Flood in Encapsulation

EPG Flood on Encapsulation: Disabled Enabled



L2 Unknown Unicast: Flood Hardware Proxy

ARP Flooding:

L3 Unknown Multicast Flooding: Flood Optimized Flood

Flood in BD Drop Flood in Encapsulation

Traffic Type

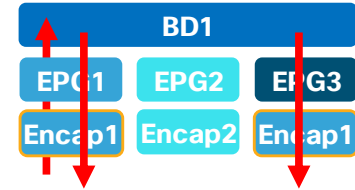
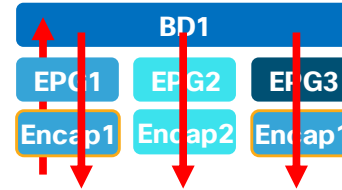
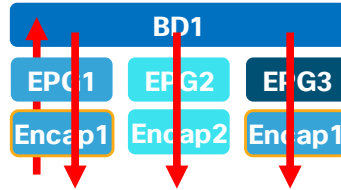
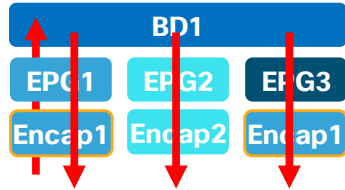
L2 Unknown Unicast Flood

ARP Request Flood

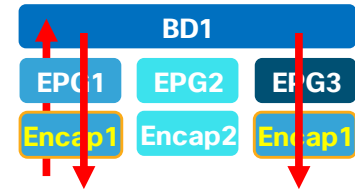
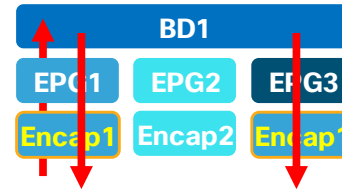
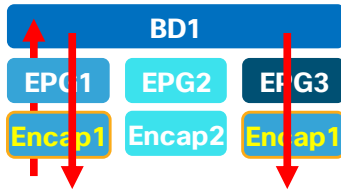
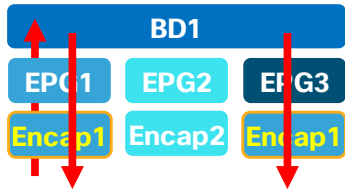
Unknown L3 Mcast Flood

other multi-dest traffic

Prior to 3.1



From 3.1

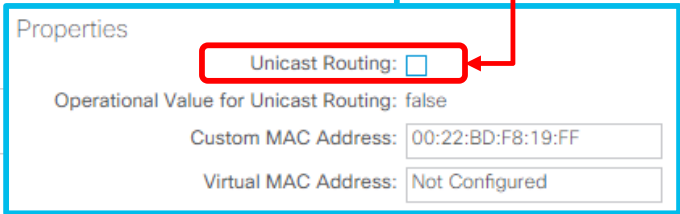
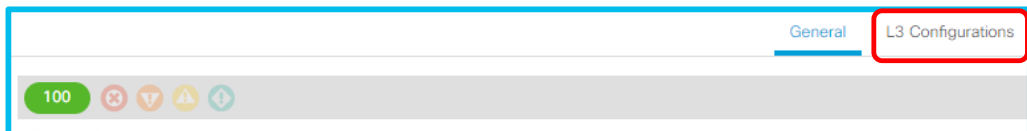


From 3.1 and 2nd generation LEAF,
all packets are flooded within encapsulation without exceptions

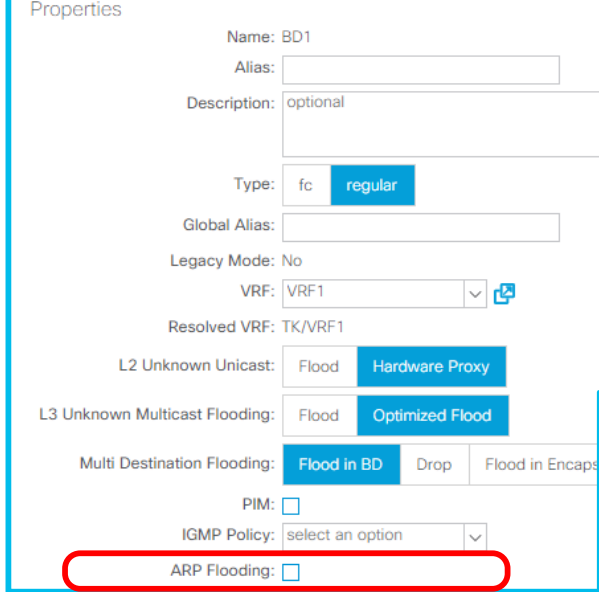
Including OSPF, BGP etc.

- ※ Flood in Encap behavior stays same on 1st generation LEAF
- ※ If traffic is not to be flooded in the first place due to other options, it will not be flooded.

ACI BD Forwarding Option (cont.)



-- TIPS --
When Unicast Routing is OFF, ARP Flooding is enabled internally even though config shows off



```
leaf1# show vlan | grep TK:BD1
22 TK:BD1 active Eth1/1, Eth1/2

leaf1# vsh_lc -c 'show system internal eltc info vlan 22 detail' | grep _mode
      fwd_mode:      bridge
      arp_mode:      unicast
      hw_arp_mode:    flood
      unk_uc_mode:    proxy
```

Unicast Routing - Off

ARP Flooding - Off

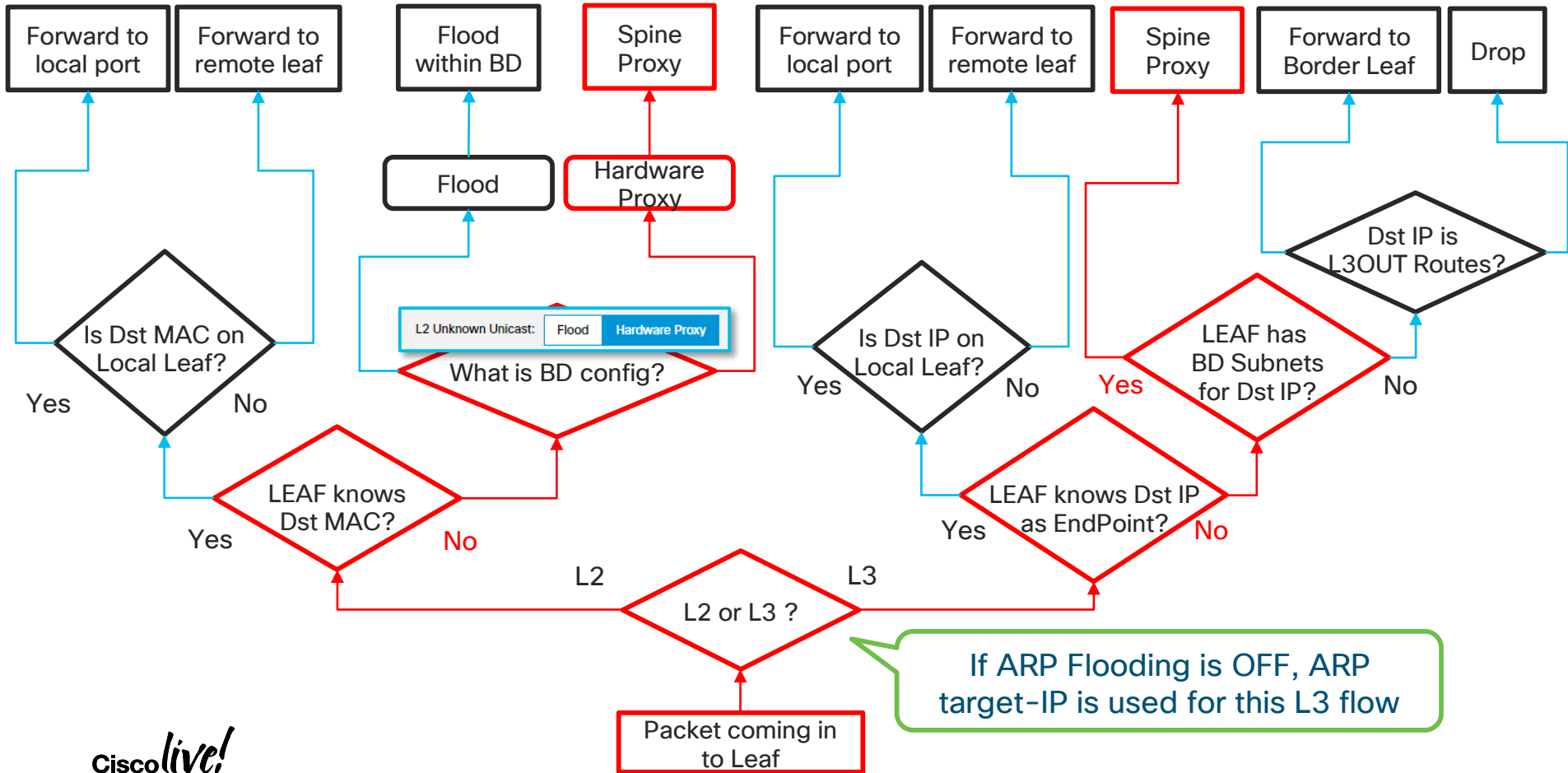
ARP Flooding in H/W - On

L2 Unknown Unicast - Hardware Proxy

Agenda

- Introduction
 - ACI Overlay VxLAN and TEP
- ACI Forwarding components
 - Endpoints, EPG, EP Learning, COOP and How it all works
 - BD, VRF forwarding scope and detailed options
 - Spine-Proxy and ARP Glean
 - Forwarding Software Architecture and ASIC Generation
- ACI Packet Walk
 - Walk through the life of a packet going through ACI

Spine Proxy Summary



How to check Spine-Proxy TEP



```
leaf1# show ip route vrf TK: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 dsteps 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

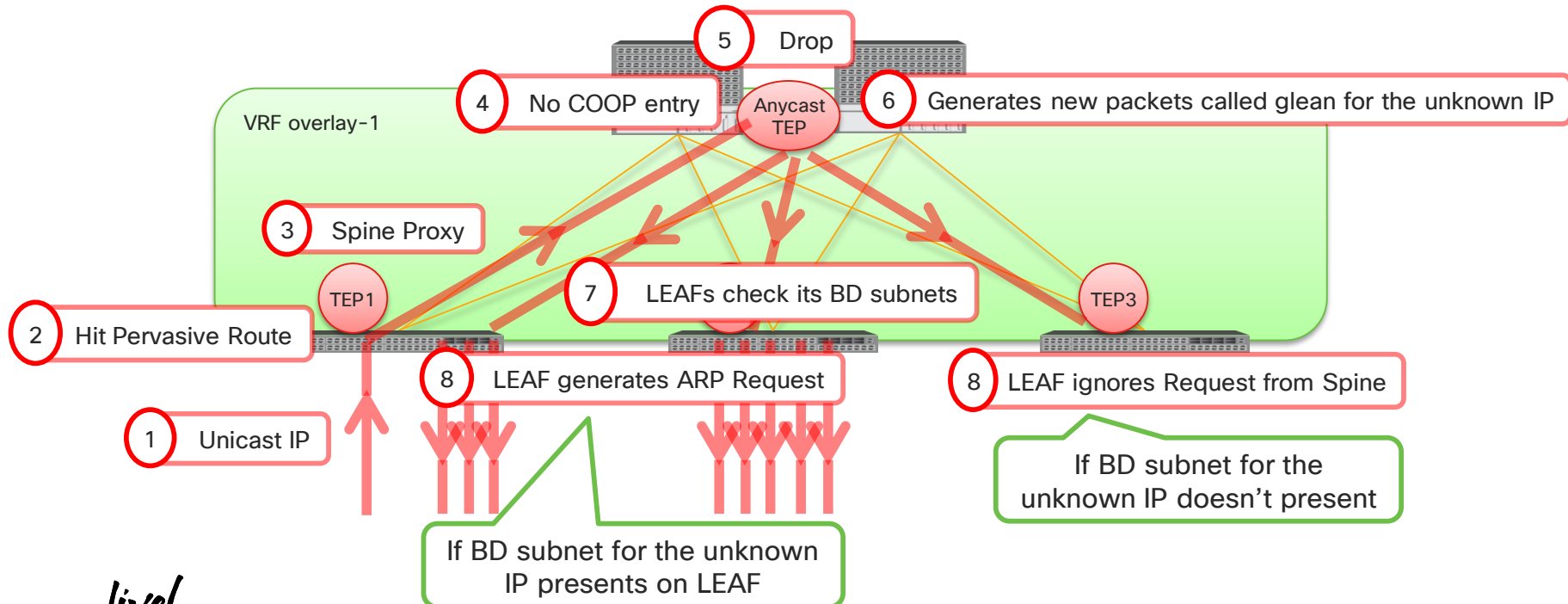
Three types of Spine Proxy TEP

- 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

ARP Glean (Silent Host Tracking)

What if even SPINE COOP doesn't know the destination when proxy'ed?

- ✓ L2 Traffic : Drop
- ✓ L3 Traffic : ARP Glean



Agenda

- Introduction
 - ACI Overlay VxLAN and TEP
- ACI Forwarding components
 - Endpoints, EPG, EP Learning, COOP and How it all works
 - BD, VRF forwarding scope and detailed options
 - Spine-Proxy and ARP Glean
 - Forwarding Software Architecture and ASIC Generation
- ACI Packet Walk
 - Walk through the life of a packet going through ACI

ACI Forwarding Table & Software Architecture

on the **Supervisor Engine**:

EPM (EndPoint Manager): manages host MAC & IP learning

uRIB (Unicast RIB): contains the unicast routing information

Policy Mgr: manages contracts between EPGs or L3OUT.

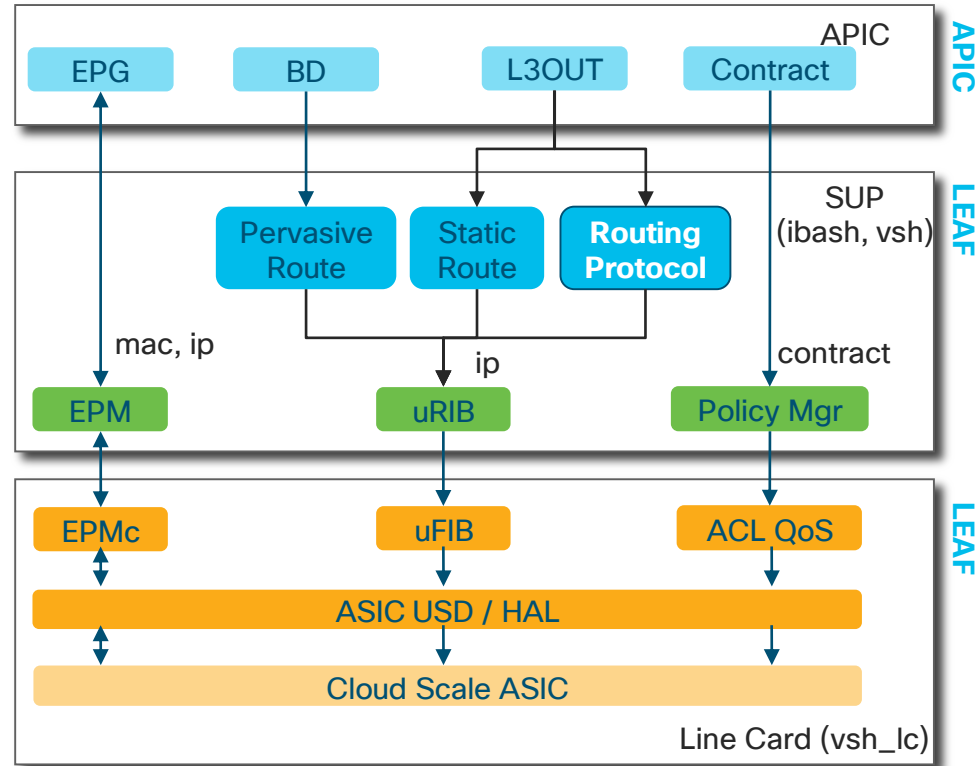
on the **Linecards**:

EPMc (EndPoint Manager Client): learns host MAC & IP addresses from hardware(dataplane) via HAL

uFIB (Unicast FIB): programs the hardware unicast routing table via HAL

ACL QoS: programs contracts via HAL

HAL (Hardware Abstraction Layer): pass the messages between hardware(ASIC) and software



※ ASIC USD (User Space Driver) is only for 1st generation ASIC

ACI Forwarding Table & Software Architecture



on the **Supervisor Engine: ibash (default)**

EPM show endpoint
show system internal epm

uRIB show ip route vrf xxx

Policy Mgr show system internal policymgr

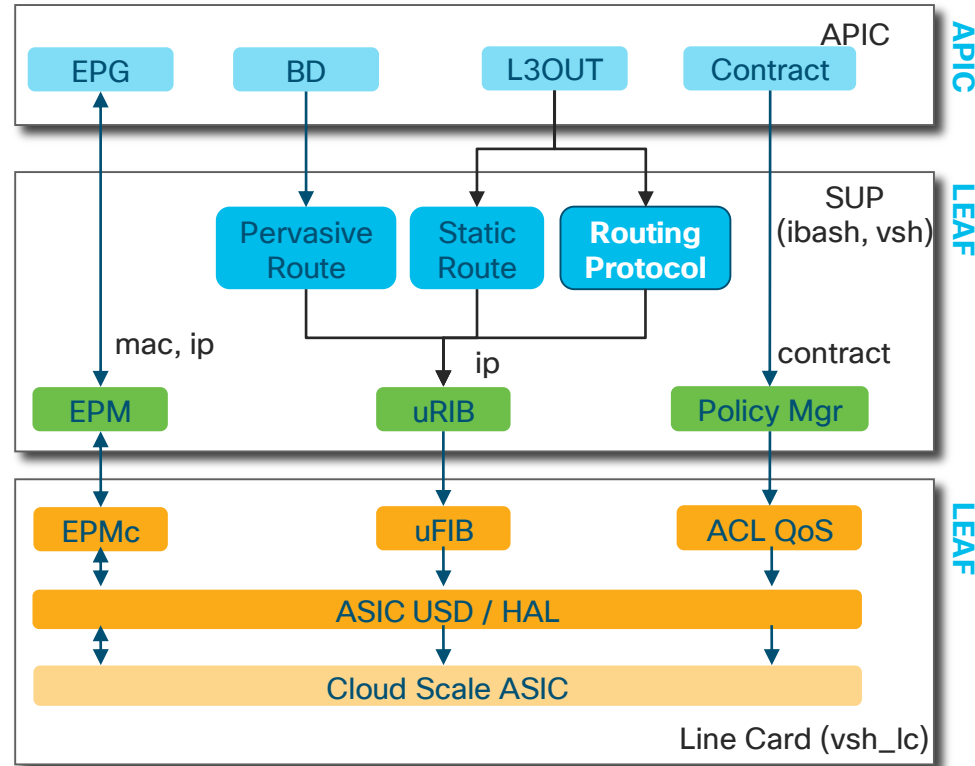
on the **Linecards: vsh_lc**

EPMc show system internal epmc ...

uFIB show forwarding ...

ACL QoS show system internal aclqos ...

HAL show platform internal hal ...



※ ASIC USD (User Space Driver) is only for 1st generation ASIC

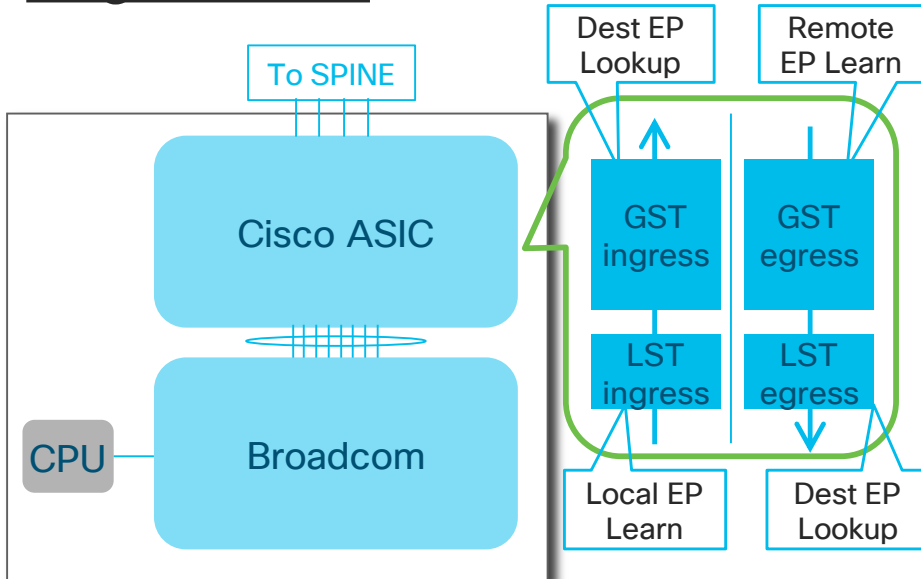
LEAF ASIC Generations

※ LST: Local Station Table, GST: Global Station Table

※ FP Tile: Forwarding and Policy Tile

※ HAL: Hardware Abstraction Layer

1st generation

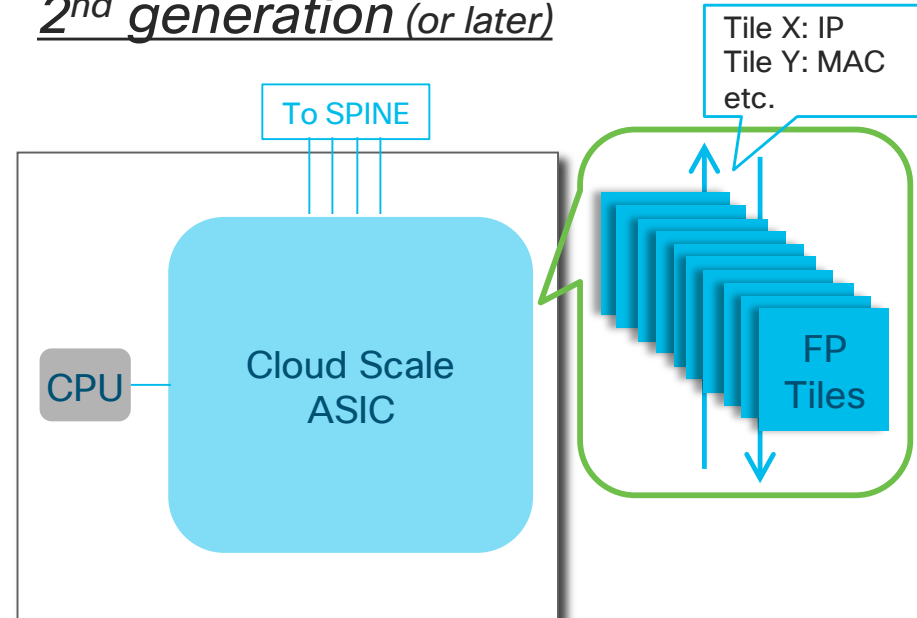


N9K-C9332PQ N9K-C9396PX
 N9K-C9372PX N9K-C9396TX
 N9K-C9372PX-E N9K-C93120TX
 N9K-C9372TX N9K-C93128TX
 N9K-C9372TX-E

Cisco live!

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

2nd generation (or later)



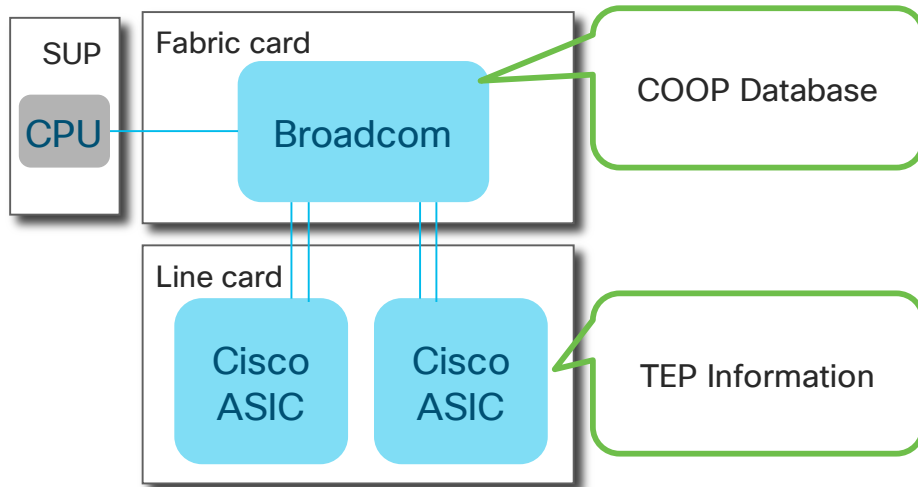
N9K-C93180YC-EX N9K-C93180YC-FX
 N9K-C93108TC-EX N9K-C93108TC-FX
 N9K-C93180LC-EX N9K-C9348GC-FXP
 N9K-C9336C-FX2
 N9K-C93240YC-FX2

- 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

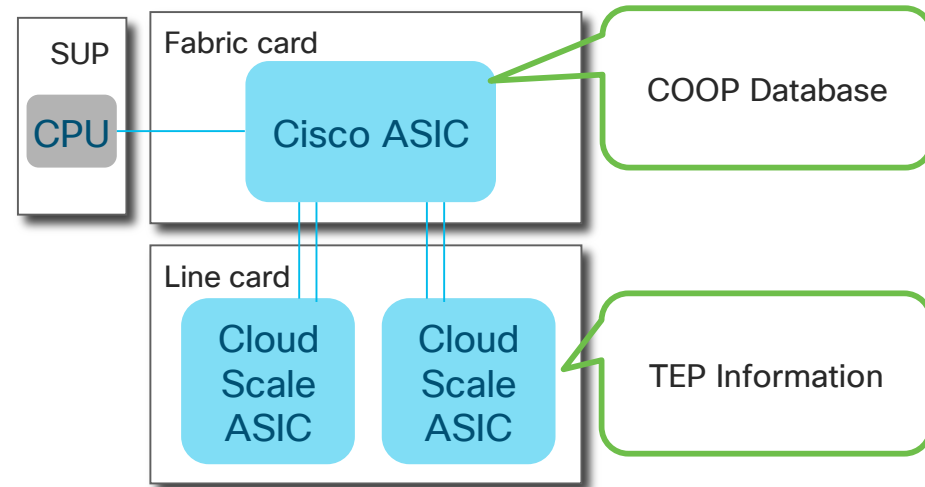
Fabric card

N9K-C9504-FM

N9K-C9508-FM

N9K-C9516-FM

2nd generation (or later)



Line card

N9K-X9732C-EX

N9K-X9736C-FX

Box spine

N9K-C9364C

N9K-C9332C

Fabric card

N9K-C9504FM-E

N9K-C9508FM-E

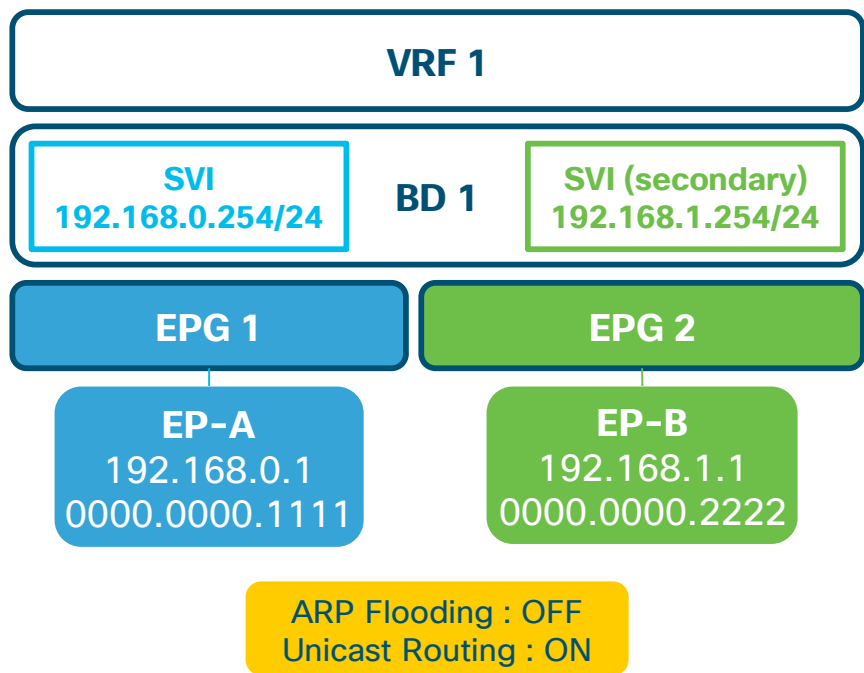
N9K-C9508FM-E2

N9K-C9516FM-E2

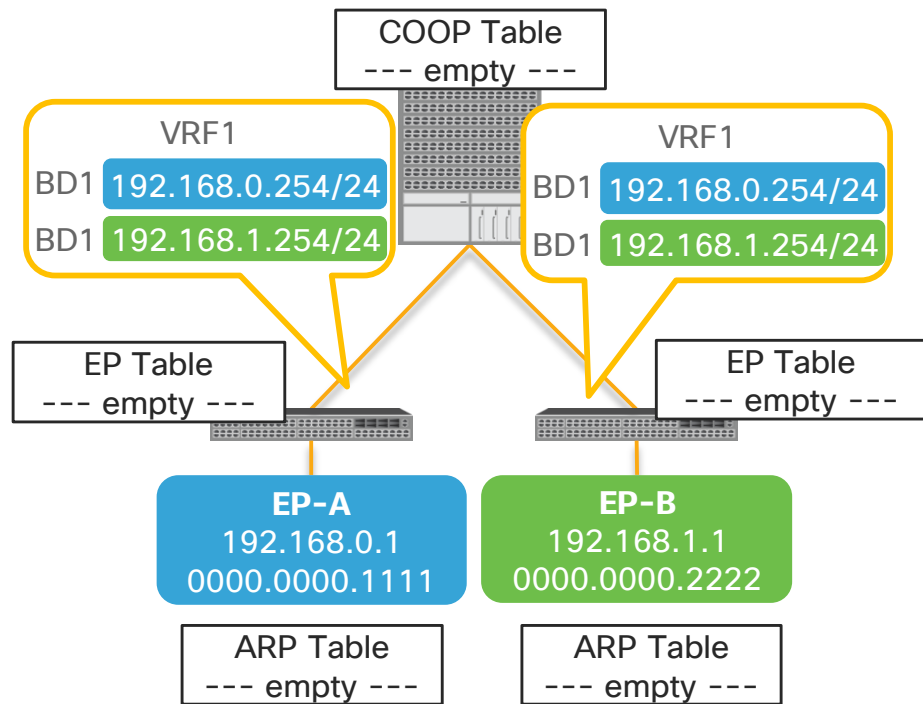
Agenda

- Introduction
 - ACI Overlay VxLAN and TEP
- ACI Forwarding components
 - Endpoints, EPG, EP Learning, COOP and How it all works
 - BD, VRF forwarding scope and detailed options
 - Spine-Proxy and ARP Glean
 - Forwarding Software Architecture and ASIC Generation
- ACI Packet Walk
 - Walk through the life of a packet going through ACI

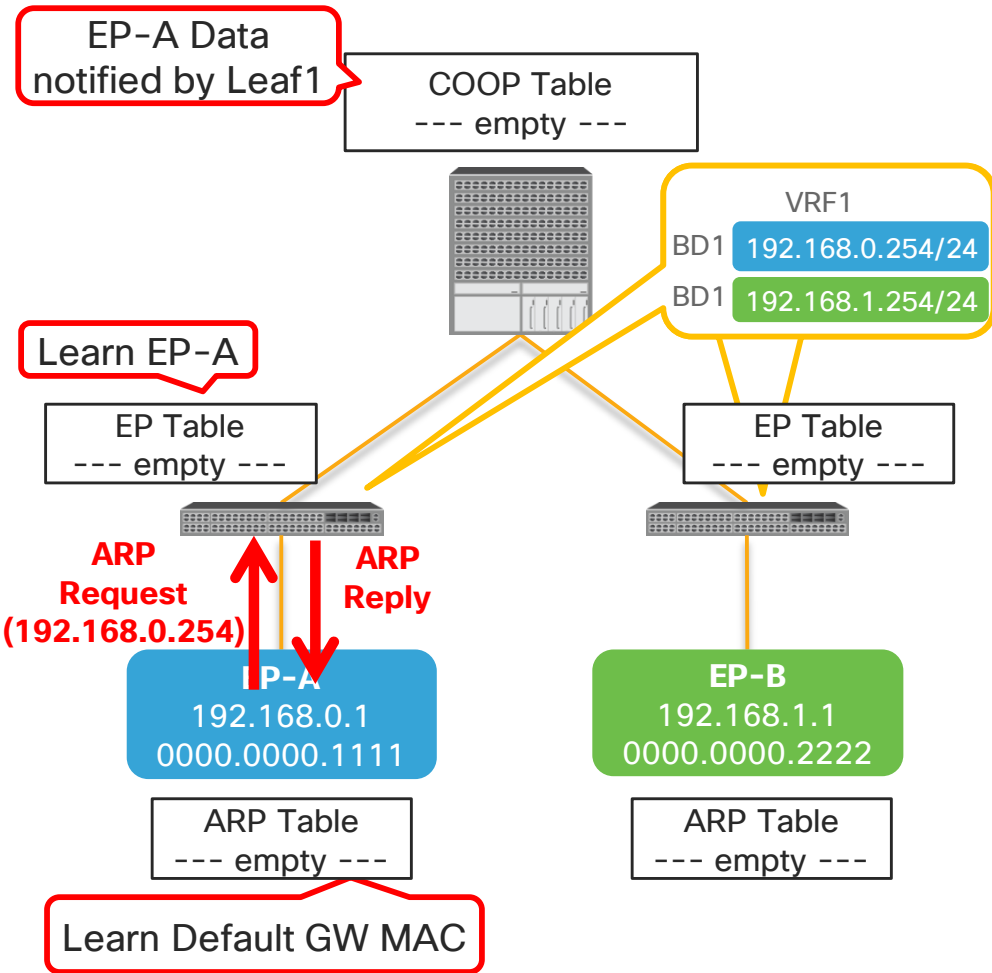
• Logical Topology



• Physical Topology



PING : EP-A (192.168.0.1) -> EP-B (192.168.1.1)



1. ARP Request to default GW

1. ARP Req is sent out to GW (192.168.0.254)
2. LEAF1 learns src IP/MAC from ARP.
 - Leaf1 notify that to Spine COOP
3. LEAF1 sends ARP reply to EP-A.

CLI notes (VLAN/EPG/BD programming)



```
LEAF1# show vlan id 69 extended
```

VLAN	Name	Status	Ports
69	TK:APP1:EPG1	active	Eth1/11

VLAN	Type	Vlan-mode	Encap
69	enet	CE	vlan-753

VLAN and I/F mapping

```
LEAF1# show system internal epm vlan 69
```

VLAN ID	Type	Access Encap (Type Value)	Fabric Encap	H/W id	BD VLAN	Endpoint Count
69	FD vlan	802.1Q	753 8994	68	68	1

BD PI-VLAN

```
LEAF1# show ip interface vlan 68 vrf TK:VRF1
```

```
vlan68, Interface status: protocol-up/link-up/admin-up, iod: 86, mode: pervasive  
IP address: 192.168.0.254, IP subnet: 192.168.0.0/24  
IP address: 192.168.1.254, IP subnet: 192.168.1.0/24 secondary
```

BD Pervasive GW

CLI notes (Source learning)



```
LEAF1# show endpoint ip 192.168.0.1 detail
```

Legend:

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

EndPoint Table
(= host table)

VLAN/ Domain	Encap VLAN	MAC Address IP Address	MAC Info/ IP Info	Interface	Endpoint Group Info
69	vlan-753	0000.0000.1111 L		eth1/11	TK:APP1:EPG1
TK:VRF1	vlan-753	192.168.0.1 L		eth1/11	

```
LEAF1# show ip route vrf TK:VRF1
```

```
192.168.0.0/24, ubest/mbest: 1/0, attached, direct, pervasive  
  *via 10.0.184.65%overlay-1, [1/0], 01w08d, static  
192.168.0.254/32, ubest/mbest: 1/0, attached, pervasive  
  *via 192.168.0.254, vlan68, [1/0], 01w08d, local, local  
192.168.1.0/24, ubest/mbest: 1/0, attached, direct, pervasive  
  *via 10.0.184.65%overlay-1, [1/0], 01w08d, static  
192.168.1.254/32, ubest/mbest: 1/0, attached, pervasive  
  *via 192.168.1.254, vlan68, [1/0], 01w08d, local, local
```

RIB
(= LPM table)

CLI notes (COOP sync)



```
LEAF1# show vrf TK:VRF1 detail extended | grep vxlan
Encap: vxlan-2228224
```

VRF VNID

```
LEAF1# show vlan id 68 extended
```

VLAN Name	Status	Ports
68 TK:BD1	active	Eth1/11

VLAN Type	Vlan-mode	Encap
68 enet	CE	vxlan-16711542

BD VNID

```
fab2-spine1# show coop internal info ip-db key 2228224 192.168.0.1
```

```
IP address : 192.168.0.1
Vrf : 2228224
Flags : 0
EP bd vnid : 16711542
EP mac : 00:00:00:00:11:11
Publisher Id : 10.0.8.95
URIB Tunnel Info
Num tunnels : 1
    Tunnel address : 10.0.8.95
    Tunnel ref count : 1
```

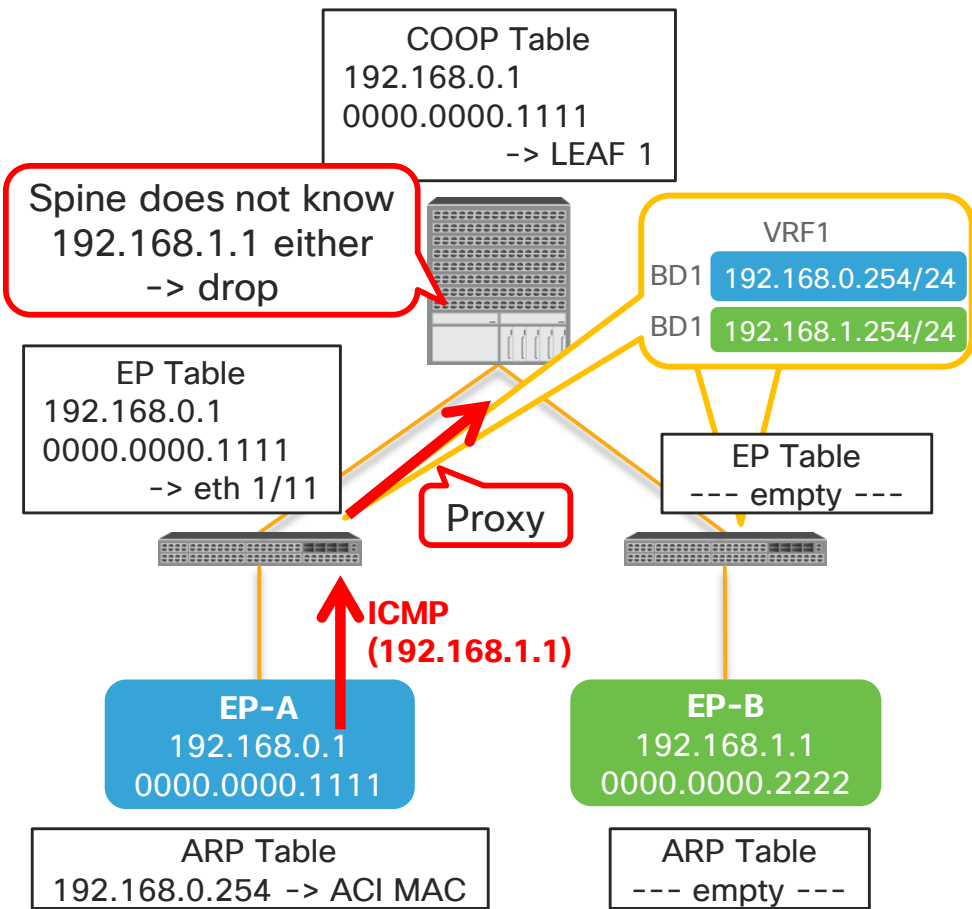
COOP on SPINE
(with IP as a key)

```
fab2-spine1# show coop internal info repo ep key
16711542 0000.0000.1111
```

```
EP bd vnid : 16711542
EP mac : 00:00:00:00:11:11
Vrf vnid : 2228224
publisher id : 10.0.8.95
Real IPv4 EP : 192.168.0.1
```

COOP on SPINE
(with MAC as a key)

❖ command outputs are sniped



1. ARP Request to default GW

1. ARP Req is sent out to GW (192.168.0.254)
2. LEAF1 learns src IP/MAC from ARP.
 - Leaf1 notify that to Spine COOP
3. LEAF1 sends ARP reply to EP-A.

2. ICMP from EP-A to EP-B (192.168.1.1)

1. Dst MAC is ACI MAC (BD SVI router-mac)
 - L3 Lookup within VRF
2. LEAF1 doesn't know 192.168.1.1 but knows it's subnet (192.168.1.0/254)
 - Spine-Proxy

3. Spine COOP lookup

1. COOP doesn't know 192.168.1.1 either
 - drop

CLI notes (Destination lookup)



```
LEAF1# show endpoint ip 192.168.1.1 detail
```

```
Legend:
```

```
--- snip ---
```

VLAN/ Domain	Encap VLAN	MAC Address IP Address	MAC Info/ IP Info	Interface	Endpoint Group Info
<----- no output ----->					

EndPoint Table
(= host table)

```
LEAF1# show ip route vrf TK:VRF1
```

```
192.168.0.0/24, ubest/mbest: 1/0, attached, direct, pervasive
  *via 10.0.184.65%overlay-1, [1/0], 01w08d, static
192.168.0.254/32, ubest/mbest: 1/0, attached, pervasive
  *via 192.168.0.254, vlan68, [1/0], 01w08d, local, local
192.168.1.0/24, ubest/mbest: 1/0, attached, direct, pervasive
  *via 10.0.184.65%overlay-1, [1/0], 00:00:06, static
192.168.1.254/32, ubest/mbest: 1/0, attached, pervasive
  *via 192.168.1.254, vlan68, [1/0], 00:00:06, local, localn
```

RIB
(= LPM table)

```
LEAF1# show isis dteps vrf overlay-1 | grep 10.0.184.65
```

```
10.0.184.65          SPINE    N/A          PHYSICAL, PROXY-ACAST-V4
```

NextHop is
IPv4 Proxy

```
LEAF1# show ip route 10.0.184.65 vrf overlay-1
```

```
10.0.184.65/32, ubest/mbest: 2/0
  *via 10.0.48.97, eth1/50.2, [115/2], 02w14d, isis-isis_infra, L1
  *via 10.0.48.94, eth1/49.1, [115/2], 02w14d, isis-isis_infra, L1
```

NextHop I/F
(to SPINE)

CLI notes (Coop Check)

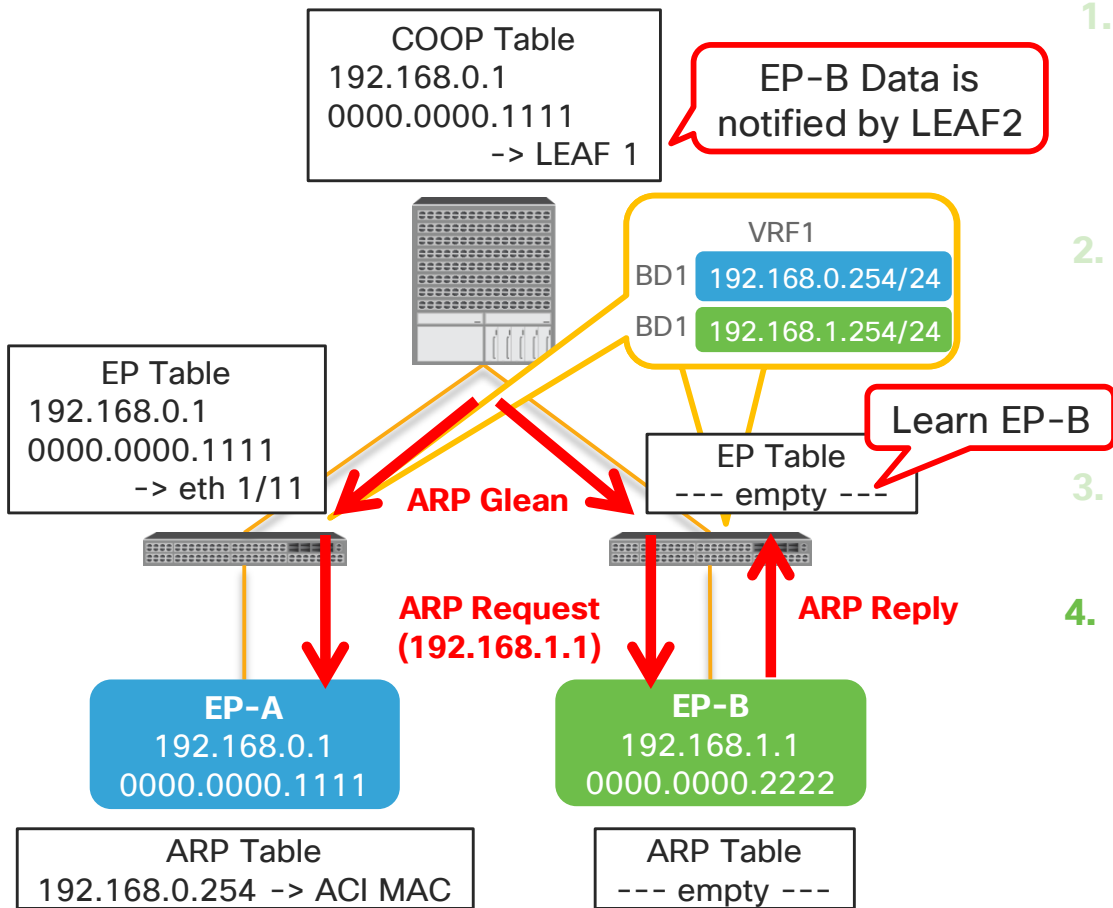


```
LEAF1# show vrf TK:VRF1 detail extended | grep vxlan  
Encap: vxlan-2228224
```

VRF VNID

```
LEAF1# show coop internal info ip-db key 2228224 192.168.1.1  
  
Key not found in ip db
```

COOP on
SPINE



1. ARP Request to default GW

1. ARP Req is sent out to GW (192.168.0.254)
2. LEAF1 learns src IP/MAC from ARP.
 - Leaf1 notify that to Spine COOP
3. LEAF1 sends ARP reply to EP-A.

2. ICMP from EP-A to EP-B (192.168.1.1)

1. Dst MAC is ACI MAC (BD SVI router-mac)
 - L3 Lookup within VRF
2. LEAF1 doesn't know 192.168.1.1 but knows it's subnet (192.168.1.0/24)
 - Spine-Proxy

3. Spine COOP lookup

1. COOP doesn't know 192.168.1.1 either
 - drop

4. ARP Glean for 192.168.1.1 to each LEAFs

1. LEAF1 and LEAF2 has a BD with 192.168.1.0/24 subnet
 - Both LEAFs **generates** an ARP Request for 192.168.1.1 out of ports on the BD
2. EP-B sends ARP Reply to LEAF2
3. LEAF2 learns EP-B IP/MAC
 - LEAF2 notifies that to Spine COOP

CLI notes (LEAF2 VLAN/EPG/BD programming)



```
LEAF2# show vlan id 10 extended
```

VLAN Name	Status	Ports
-----	-----	-----
10 TK:APP1:EPG2	active	Eth1/11

VLAN Type	Vlan-mode	Encap
-----	-----	-----
10 enet	CE	vlan-754

VLAN and I/F mapping

```
LEAF2# show system internal epm vlan 10
```

VLAN ID	Type	Access Encap (Type Value)	Fabric Encap	H/W id	BD VLAN	Endpoint Count
-----	-----	-----	-----	-----	-----	-----
10	FD vlan	802.1Q	754 8987	7	9	1

BD PI-VLAN

```
LEAF2# show ip interface vlan 9 vrf TK:VRF1
```

```
vlan9, Interface status: protocol-up/link-up/admin-up, iod: 80, mode: pervasive  
IP address: 192.168.0.254, IP subnet: 192.168.0.0/24  
IP address: 192.168.1.254, IP subnet: 192.168.1.0/24 secondary
```

BD Pervasive GW

CLI notes (LEAF2 Source learning)



```
LEAF2# show endpoint ip 192.168.1.1 detail
```

Legend:

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

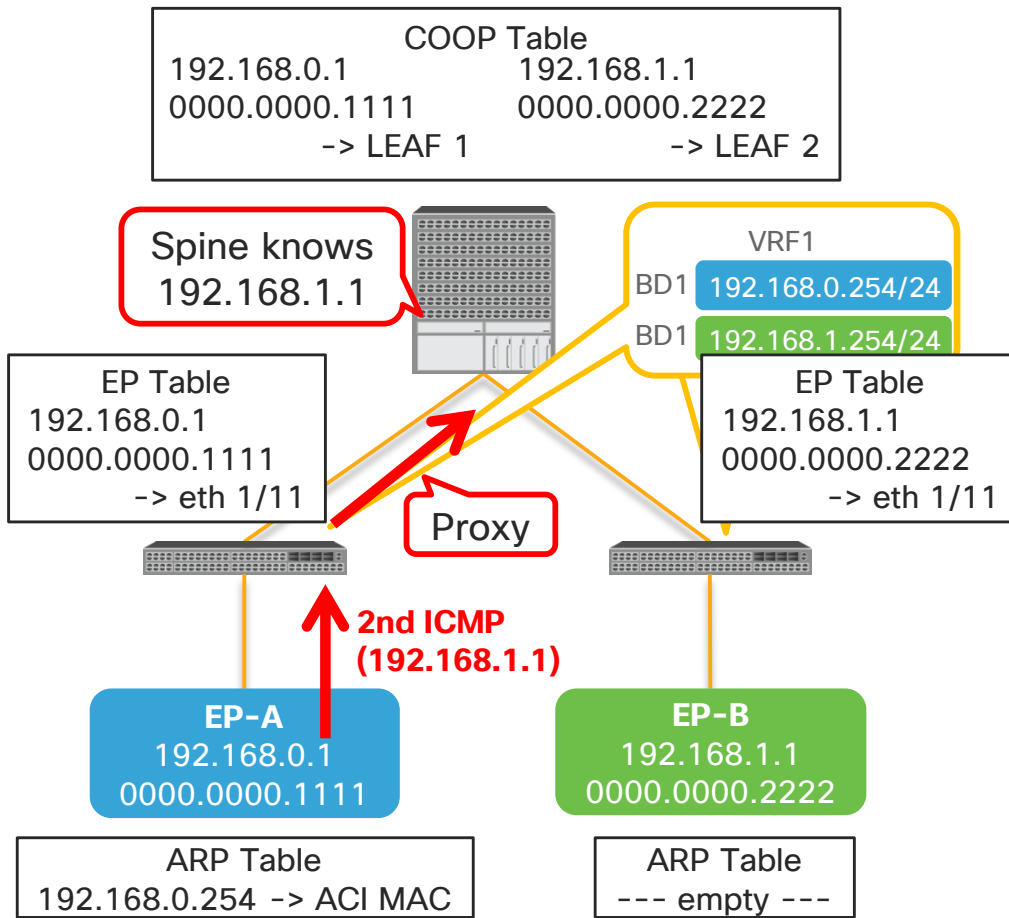
EndPoint Table
(= host table)

VLAN/ Domain	Encap VLAN	MAC Address IP Address	MAC Info/ IP Info	Interface	Endpoint Group Info
10	vlan-754	0000.0000.2222 L		eth1/11	TK:APP1:EPG2
TK:VRF1	vlan-754	192.168.1.1 L		eth1/11	

```
LEAF2# show ip route vrf TK:VRF1
```

```
192.168.1.0/24, ubest/mbest: 1/0, attached, direct, pervasive  
*via 10.0.184.65%overlay-1, [1/0], 01w08d, static  
192.168.1.254/32, ubest/mbest: 1/0, attached, pervasive  
*via 192.168.1.254, vlan9, [1/0], 01w08d, local, local  
192.168.1.0/24, ubest/mbest: 1/0, attached, direct, pervasive  
*via 10.0.184.65%overlay-1, [1/0], 01w08d, static  
192.168.1.254/32, ubest/mbest: 1/0, attached, pervasive  
*via 192.168.1.254, vlan9, [1/0], 01w08d, local, local
```

RIB
(= LPM table)

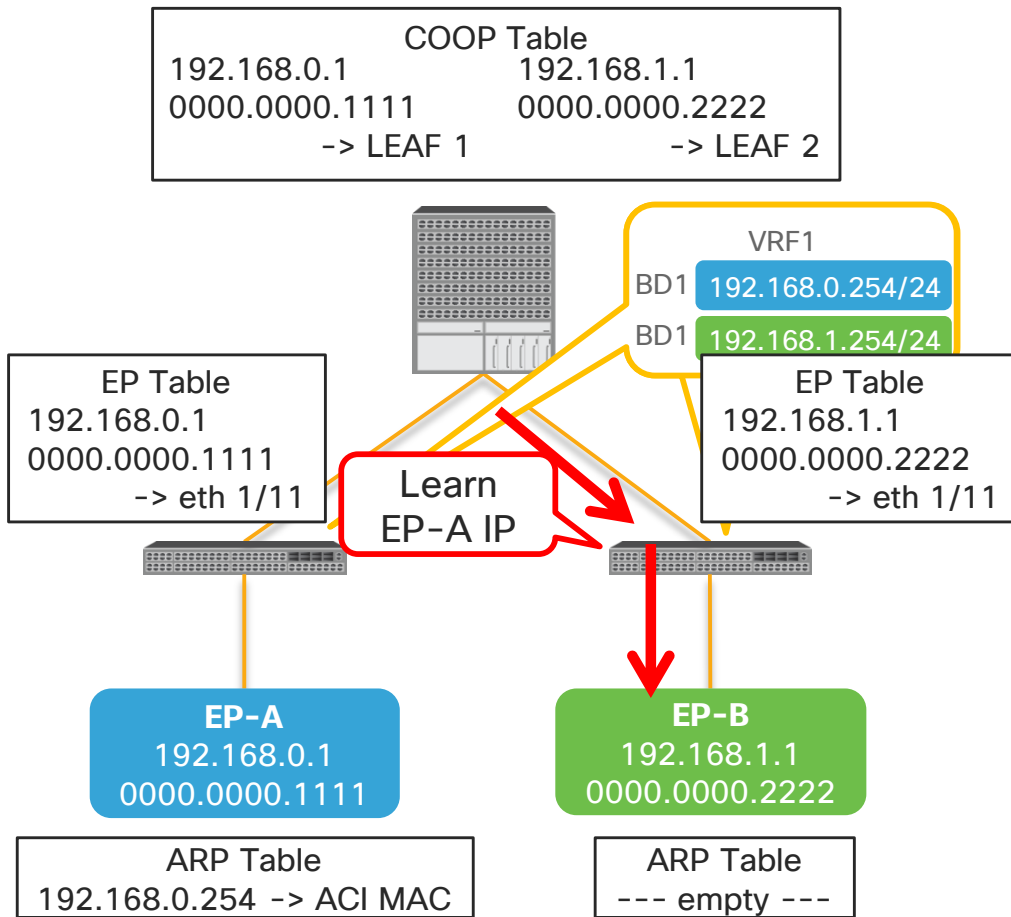


4. EP-A sends 2nd ICMP to EP-B (192.168.1.1)

1. Dst MAC is ACI MAC (BD SVI router-mac)
 - L3 Lookup within VRF
2. LEAF1 still doesn't know 192.168.1.1 but knows its subnet (192.168.1.0/254)
 - Spine-Proxy

5. Spine COOP lookup for 2nd ICMP

1. Now COOP knows 192.168.1.1
2. Spine sends it to Leaf2



4. EP-A sends 2nd ICMP to EP-B (192.168.1.1)

1. Dst MAC is ACI MAC (BD SVI router-mac)
 - L3 Lookup within VRF
2. LEAF1 still doesn't know 192.168.1.1 but knows its subnet (192.168.1.0/254)
 - Spine-Proxy

5. Spine COOP lookup for 2nd ICMP

1. Now COOP knows 192.168.1.1
2. Spine sends it to Leaf2

6. LEAF2 learns EP-A as a remote EP

- The packet is routed = sent out with VRF VNID.
- Only IP is learned

7. LEAF2 sends it out to EP-B

CLI notes (remote EP learning)



```
LEAF2# show endpoint vrf TK:VRF1 detail
```

```
Legend:
```

```
--- snip ---
```

VLAN/ Domain	Encap VLAN	MAC Address IP Address	MAC Info/ IP Info	Interface	Endpoint Group Info
TK:VRF1		192.168.0.1		tunnell1	
10	vlan-754	0000.0000.2222	L	eth1/11	TK:APP1:EPG2
TK:VRF1	vlan-754	192.168.1.1	L	eth1/11	

Learn Remote EP

```
LEAF2# show int tunnel 11 | grep dest
```

```
Tunnel destination 10.0.8.95
```

```
LEAF2# acidiag fvnread | grep 8.95
```

```
101 1 LEAF1 ABC1234DEFG 10.0.8.95/32 leaf active 0
```

Tunnel points to
LEAF1

```
LEAF2# show sys int epm endpoint ip 192.168.0.1
```

```
MAC : 0000.0000.0000 ::: Num IPs : 1
```

```
IP# 0 : 192.168.0.1 ::: IP# 0 flags :
```

```
Vlan id : 0 ::: Vlan vnid : 0 ::: VRF name : TK:VRF1
```

```
BD vnid : 0 ::: VRF vnid : 2228224
```

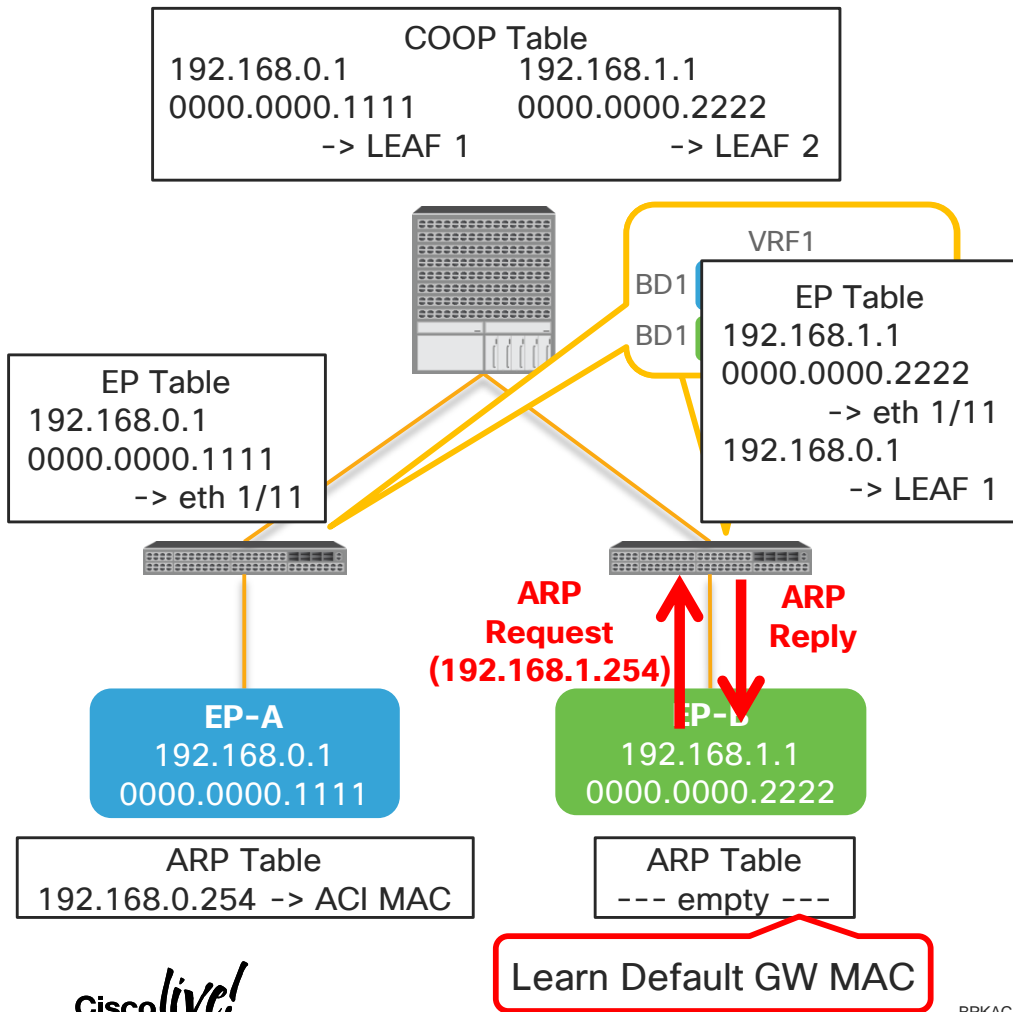
```
Phy If : 0 ::: Tunnel If : 0x1801000b
```

```
Interface : Tunnell1
```

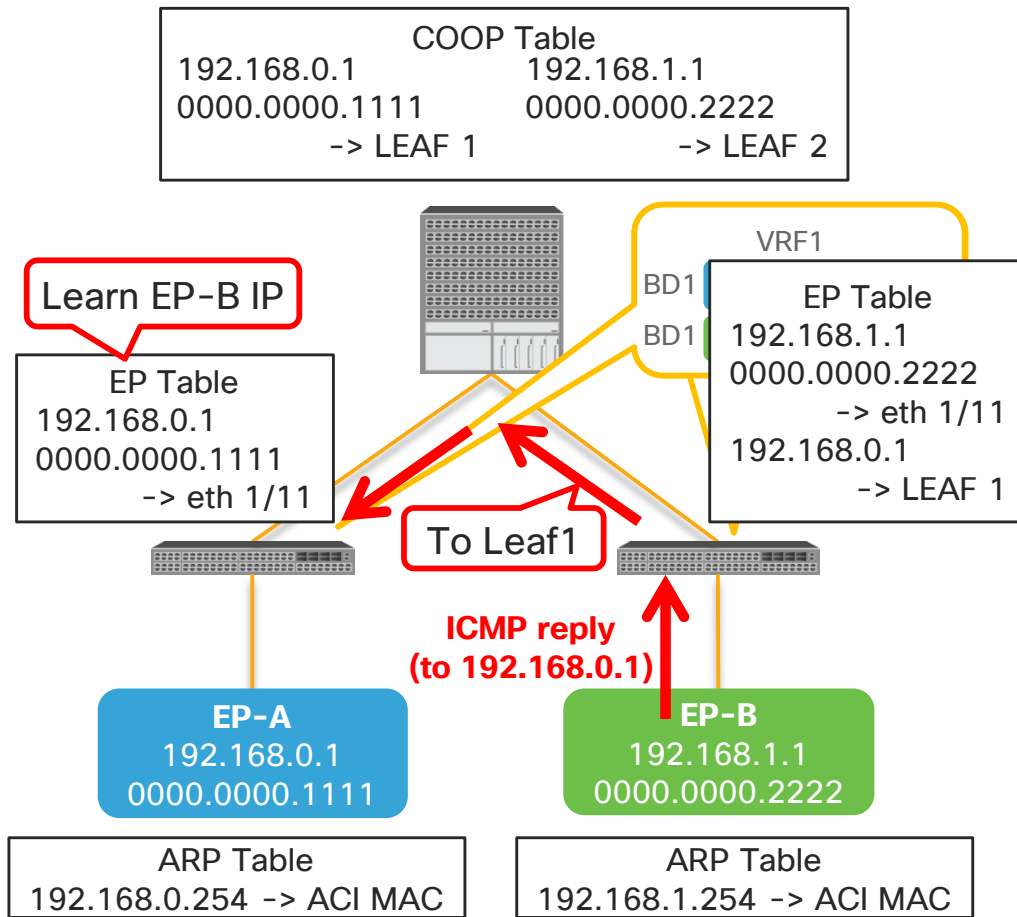
```
Flags : 0x80004400 ::: sclass : 5479 ::: Ref count : 3
```

```
--- snip ---
```

Packet was L3 traffic
-> MAC won't be learned



- 4. EP-A sends 2nd ICMP to EP-B (192.168.1.1)**
 - Dst MAC is ACI MAC (BD SVI router-mac)
 - L3 Lookup within VRF
 - LEAF1 still doesn't know 192.168.1.1 but knows its subnet (192.168.1.0/254)
 - Spine-Proxy
- 5. Spine COOP lookup for 2nd ICMP**
 - Now COOP knows 192.168.1.1
 - Spine sends it to Leaf2
- 6. LEAF2 learns EP-A as a remote EP**
 - The packet is routed = sent out with VRF VNID.
 - Only IP is learned
- 7. LEAF2 sends it out to EP-B**
- 8. EP-B resolves ARP for its gateway (192.168.1.254)**



4. EP-A sends 2nd ICMP to EP-B (192.168.1.1)

1. Dst MAC is ACI MAC (BD SVI router-mac)
 - L3 Lookup within VRF
2. LEAF1 still doesn't know 192.168.1.1 but knows it's subnet (192.168.1.0/254)
 - Spine-Proxy

5. Spine COOP lookup for 2nd ICMP

1. Now COOP knows 192.168.1.1
2. Spine sends it to Leaf2

6. LEAF2 learns EP-A as a remote EP

- The packet is routed = sent out with VRF VNID.
- Only IP is learned

7. LEAF2 sends it out to EP-B

8. EP-B resolves ARP for its gateway (192.168.1.254)

9. EP-B sends ICMP reply

1. LEAF2 already knows where EP-A IP is
 - Directly sends it to LEAF1

10. LEAF1 learns EP-B IP as a remote EP

- Only IP is learned as well

CLI notes (remote EP learning)



```
LEAF1# show endpoint vrf TK:VRF1
```

```
Legend:
```

```
--- snip ---
```

VLAN/ Domain	Encap VLAN	MAC Address IP Address	MAC Info/ IP Info	Interface
TK:VRF1		192.168.1.1		tunnel6
69	vlan-753	0000.0000.1111 L		eth1/11
TK:VRF1	vlan-753	192.168.0.1 L		eth1/11

Learn Remote EP

```
LEAF1# show int tunnel 6 | grep dest
```

```
Tunnel destination 10.0.8.90
```

```
LEAF1# acidiag fmvread | grep 8.95
```

```
102      1      LEAF2      ABC5678DEFG      10.0.8.90/32      leaf      active      0
```

Tunnel points to
LEAF2

```
LEAF1# show sys int epm endpoint ip 192.168.1.1
```

```
MAC : 0000.0000.0000 ::: Num IPs : 1
```

```
IP# 0 : 192.168.1.1 ::: IP# 0 flags :
```

```
Vlan id : 0 ::: Vlan vnid : 0 ::: VRF name : TK:VRF1
```

```
BD vnid : 0 ::: VRF vnid : 2228224
```

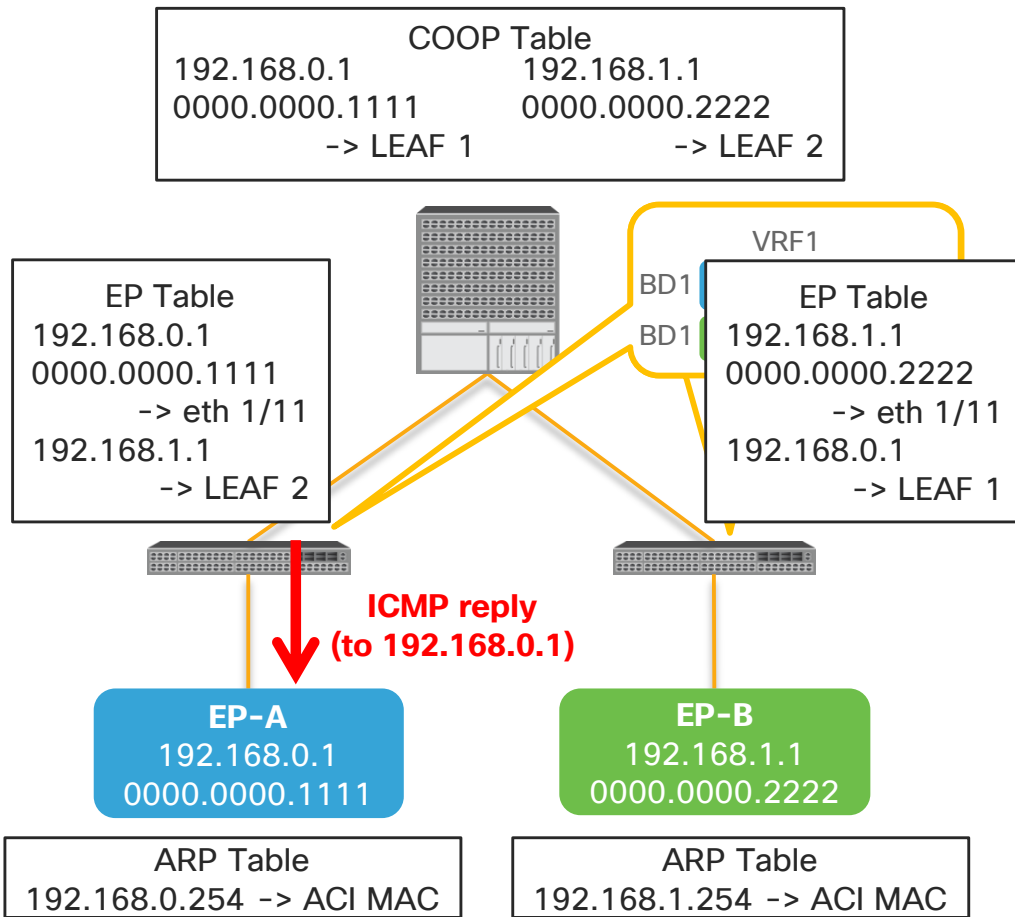
```
Phy If : 0 ::: Tunnel If : 0x18010006
```

```
Interface : Tunnel6
```

```
Flags : 0x80004400 ::: sclass : 49160 ::: Ref count : 3
```

```
--- snip ---
```

Packet was L3 traffic
-> MAC won't be learned



4. EP-A sends 2nd ICMP to EP-B (192.168.1.1)

1. Dst MAC is ACI MAC (BD SVI router-mac)
 - L3 Lookup within VRF
2. LEAF1 still doesn't know 192.168.1.1 but knows its subnet (192.168.1.0/254)
 - Spine-Proxy

5. Spine COOP lookup for 2nd ICMP

1. Now COOP knows 192.168.1.1
2. Spine sends it to Leaf2

6. LEAF2 learns EP-A as a remote EP

- The packet is routed = sent out with VRF VNID.
- Only IP is learned

7. LEAF2 sends it out to EP-B

8. EP-B resolves ARP for its gateway (192.168.1.254)

9. EP-B sends ICMP reply

1. LEAF2 already knows where EP-A IP is
 - Directly sends it to LEAF1

10. LEAF1 learns EP-B IP as a remote EP

- Only IP is learned as well

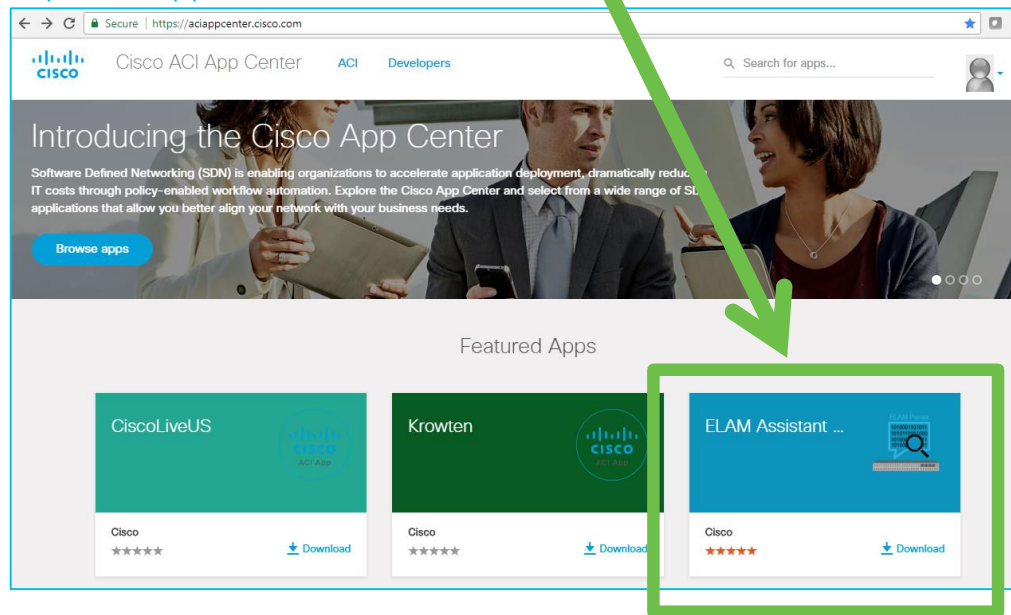
Appendix

ELAM Assistant in ACI AppCenter

Interested in more detail packet forwarding verification ?

➤ **ELAM Assistant!!**

<https://aciappcenter.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://aciappcenter.cisco.com/elam-assistant-beta-2-2-1n.html>
 - How to use video, pictures
 - A download link for ELAM Assistant
- <https://learningnetwork.cisco.com/docs/DOC-34985>
 - ACI webinar for ELAM Assistant

ELAM Assistant in ACI AppCenter (example)

1. Perform ELAM

APIC System Tenants Fabric Virtual Networking L4-L7 Services Admin Operations **Apps** admin

Apps | ELAM Assistant (Beta)

ELAM Assistant

Capture Packet

- node-101 (fab5-leaf1)
- node-102 (fab5-leaf2)
- node-103 (fab5-leaf3)
- node-104 (fab5-leaf4)
- node-105 (fab5-leaf5)
- node-107 (fab5-leaf7)
- node-201 (fab5-spine1)

slot 1 (N9K-X9732C-EX)

Capture a packet with ELAM (Embedded Logic Analyzer Module)

ELAM PARAMETERS + Add Node

name your capture : (optional)

Status	Node	Direction	Source I/F	Parameters (Outer Header)	(Inner Header)
Triggered	node-101	from frontport	any	dst_ip: 192.168.1.4 src_ip: 192.168.1.1	
Set	node-102	from SPINE	any		dst_ip: 192.168.1.4 src_ip: 192.168.1.1
Triggered	node-103	from SPINE	any		dst_ip: 192.168.1.4 src_ip: 192.168.1.1

▶ Set ELAM(s) ↻ Check Trigger

ELAM Assistant in ACI AppCenter (example)

2. Read a report

Packet Forwarding Information

Destination is Local port

Destination Port	eth1/15
Destination Logical Port	Po1

EPG Classification (pcTag)

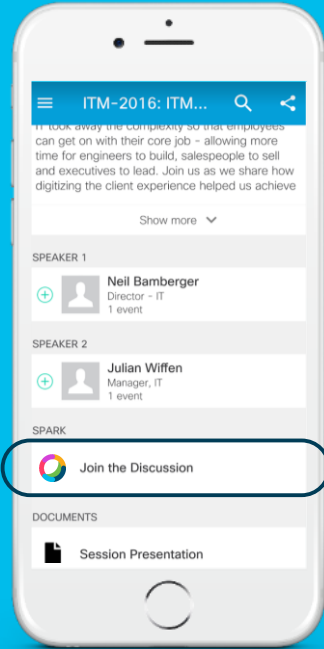
Destination EPG pcTag (dclass)	0x4006 (16390)
Source EPG pcTag (sclass)	0x2ABA (10938)
Policy Applied	1 (Contract was applied)

Lookup Drop

drop reason	no drop
-------------	---------

The screenshot shows the ELAM Assistant interface for node-103. The left sidebar lists various nodes. The main content area displays the 'ELAM REPORT FILES' section with a table showing a report file named 'node-103_elam_report.txt' from 2018-06-09. Below this is the 'ELAM REPORT PARSE RESULT' section, which includes a 'Trigger Information' table and a detailed 'Packet Forwarding Information' table. A blue callout bubble labeled 'Zoom' points to the 'Packet Forwarding Information' table in the screenshot.

Express	Detail	Raw
Captured Packet Information --- [node-103_elam_report.txt]		
Trigger Information		
Packet Direction	SPINE -> LEAF	
Incoming Interface	eth1/50	Slice Source ID(Ss) 0x38 in "show plat int hal i2 port gpd"
Outer L2 Header		
Destination MAC	000C.0C0C.8C0C	
Source MAC	000D.0D0D.9D0D	
CoS	0x0 (0)	
Access Encap VLAN	0x2 (2)	
Outer L3 Header		
L3 Type	IPv4	
DSCP	0x0 (0)	
Don't Fragment Bit	0x0	
TTL	0x1F (31)	
IP Protocol Number	17 (UDP)	
Destination IP	11.0.192.67 [fab5-leaf3]	
Source IP	11.0.192.70 [fab5-leaf1]	
Outer L4 Header		
L4 Type	IPvLAN	
Don't Learn Bit	0x0	
Sic Policy Applied Bit	0x1	
Def Policy Applied Bit	0x1	
Sic EPG (sclass/src pcTag)	0x2aba (10938) [TK-AP1-EPG1-1]	
VRF or BD VNID	0x87FAA (9285610) [TK-BD1]	
Inner L2 Header		
Destination MAC	0000.5454.5454	
Source MAC	0000.0000.5151	
Inner L3 Header		
L3 Type	IPv4	
DSCP	0x0 (0)	
Don't Fragment Bit	0x0	
TTL	0x0FF (255)	
IP Protocol Number	1 (ICMP)	
Destination IP	192.168.1.4	
Source IP	192.168.1.1	
Packet Forwarding Information		
Destination is Local port		
Destination Port	eth1/15	Ovec (0x14) in "show plat int hal i2 port gpd"
Destination Logical Port	Po1	LID (0x4D) in "show plat int hal i2 port p"
EPG Classification (pcTag)		
Destination EPG pcTag (dclass)	0x4006 (16390)	TK-AP1-EPG1-4
Source EPG pcTag (sclass)	0x2ABA (10938)	TK-AP1-EPG1-1
Policy Applied	1 (Contract was applied)	
Lookup Drop		
drop reason	no drop	



cs.co/ciscolivebot#BRKACI-3545

Cisco Webex Teams

Questions?

Use Cisco Webex Teams (formerly Cisco Spark) to chat with the speaker after the session

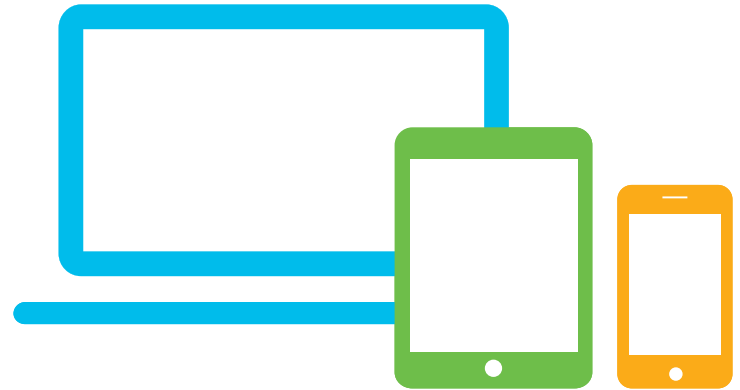
How

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

Complete your online session survey

- Please complete your Online Session Survey after each session
- Complete 4 Session Surveys & the Overall Conference Survey (available from Thursday) to receive your Cisco Live T-shirt
- All surveys can be completed via the Cisco Events Mobile App or the Communication Stations


Don't forget: Cisco Live sessions will be available for viewing on demand after the event at cicolive.cisco.com



Continue Your Education




Demos in the Cisco Showcase



Walk-in self-paced labs



Meet the engineer 1:1 meetings



Related sessions



Thank you



INTUITIVE



INTUITIVE