You make **possible**

Mastering ACI Forwarding Behavior

- A day in the life of a packet -

Takuya Kishida - Technical Marketing, DCBU ACI

BRKACI-3545

Barcelona | January 27-31, 2020

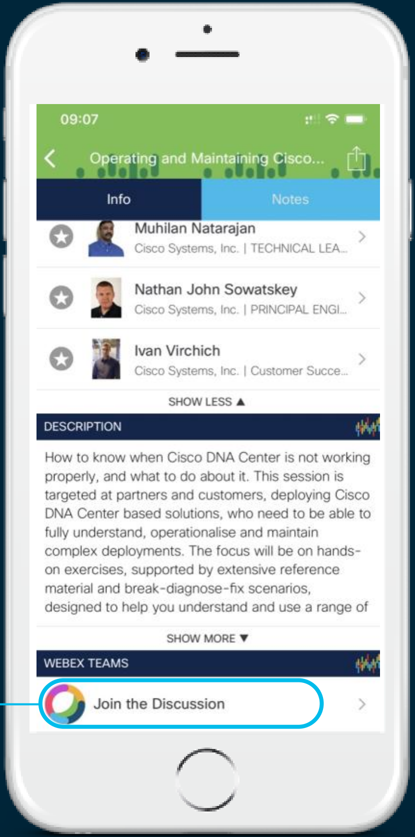
Cisco Webex Teams

Questions?

Use Cisco Webex Teams 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



09:07

Operating and Maintaining Cisco...

Muhilan Natarajan
Cisco Systems, Inc. | TECHNICAL LEA...

Nathan John Sowatskey
Cisco Systems, Inc. | PRINCIPAL ENGI...

Ivan Virchich
Cisco Systems, Inc. | Customer Succe...

DESCRIPTION

How to know when Cisco DNA Center is not working properly, and what to do about it. This session is targeted at partners and customers, deploying Cisco DNA Center based solutions, who need to be able to fully understand, operationalise and maintain complex deployments. The focus will be on hands-on exercises, supported by extensive reference material and break-diagnose-fix scenarios, designed to help you understand and use a range of

WEBEX TEAMS

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
  - Packet Capture in ACI

Basic Acronyms/Definitions

Reference Slide

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

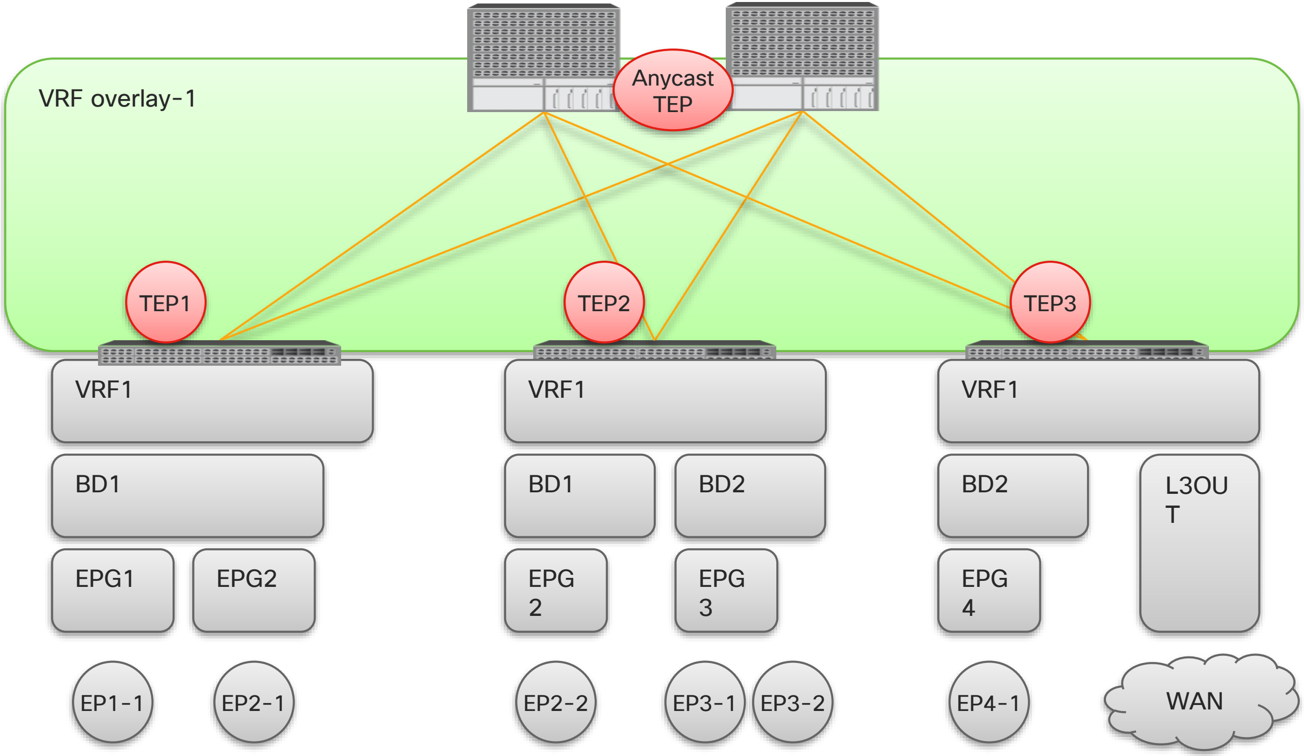
Acronyms	Definitions
d*XXX*o	Outer Destination XXX (dIPo = Outer Destination IP)
s*XXX*o	Outer Source XXX (sIPo = Outer Source IP)
d*XXX*i	Inner Destination XXX (dIPi = Inner Destination IP)
s*XXX*i	Inner Source XXX (sIPi = Inner Source IP)
GIPo	Outer Multicast Group IP
VNID	Virtual Network Identifier

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
  - Packet Capture in ACI

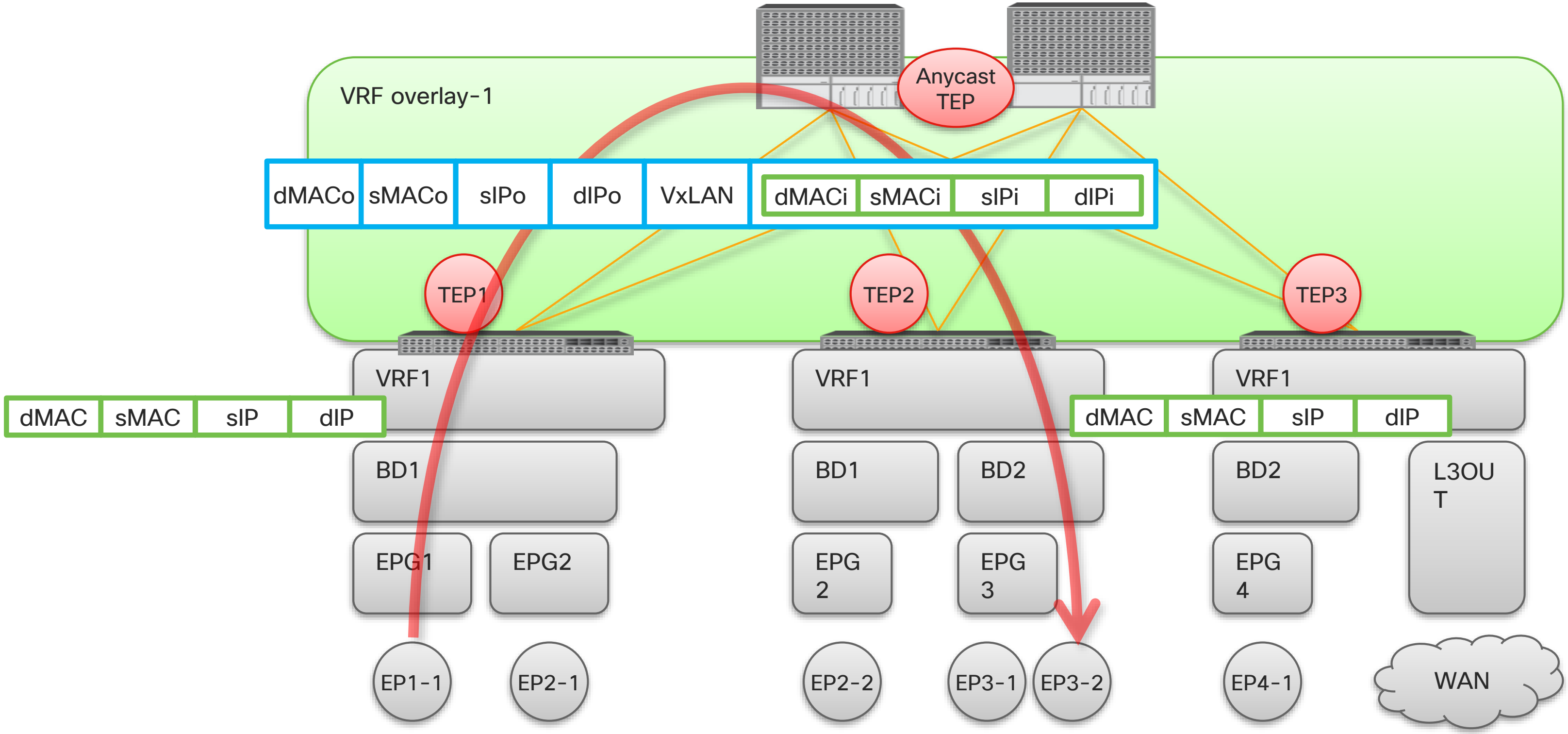
ACI Overlay VxLAN and TEP

※ TEP : Tunnel EndPoint

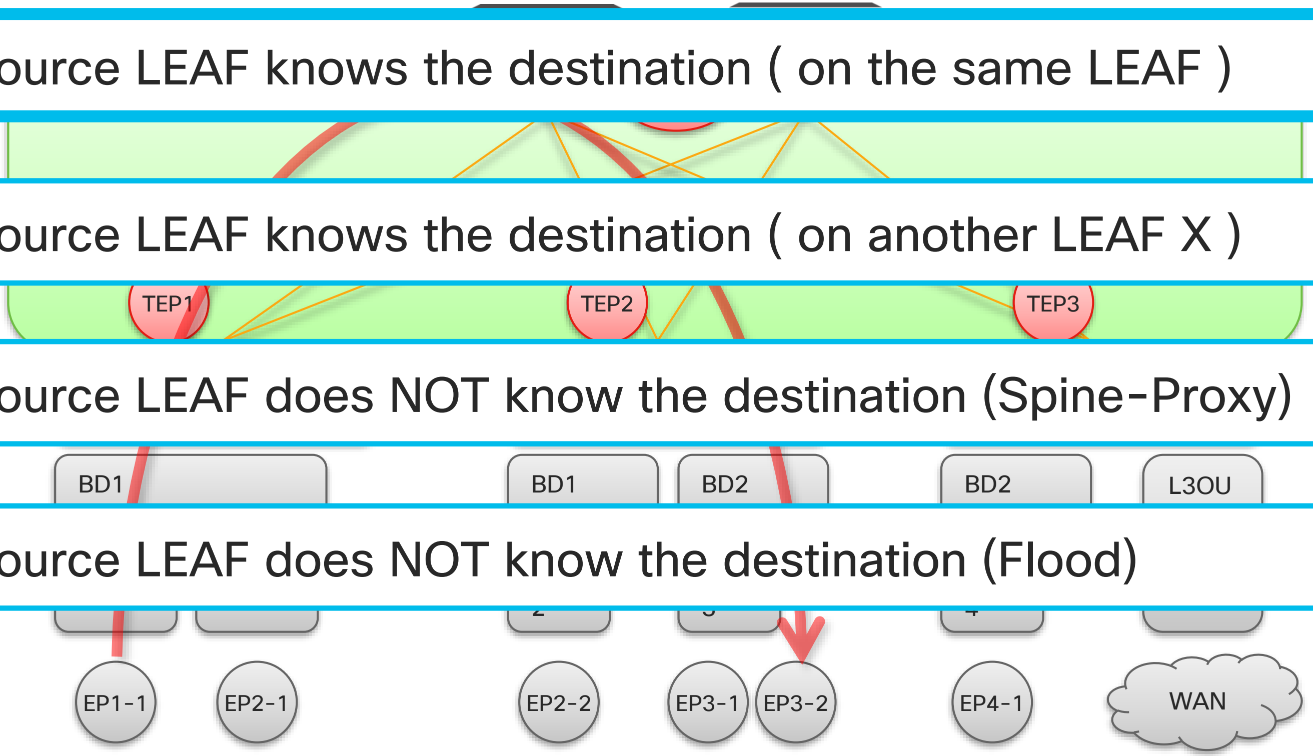


ACI Overlay VxLAN and TEP

※ TEP : Tunnel EndPoint



ACI Overlay VxLAN and TEP



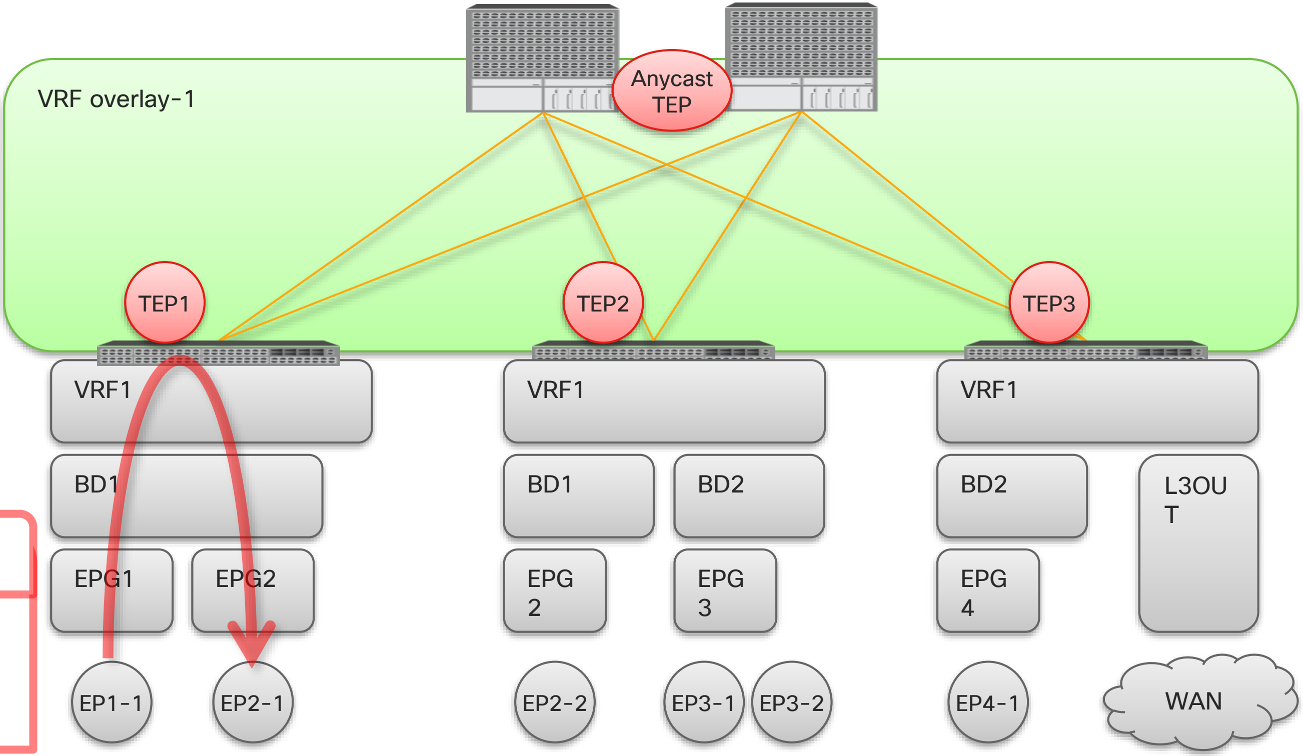
Scenario 1 : source LEAF knows the destination (on the same LEAF)

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

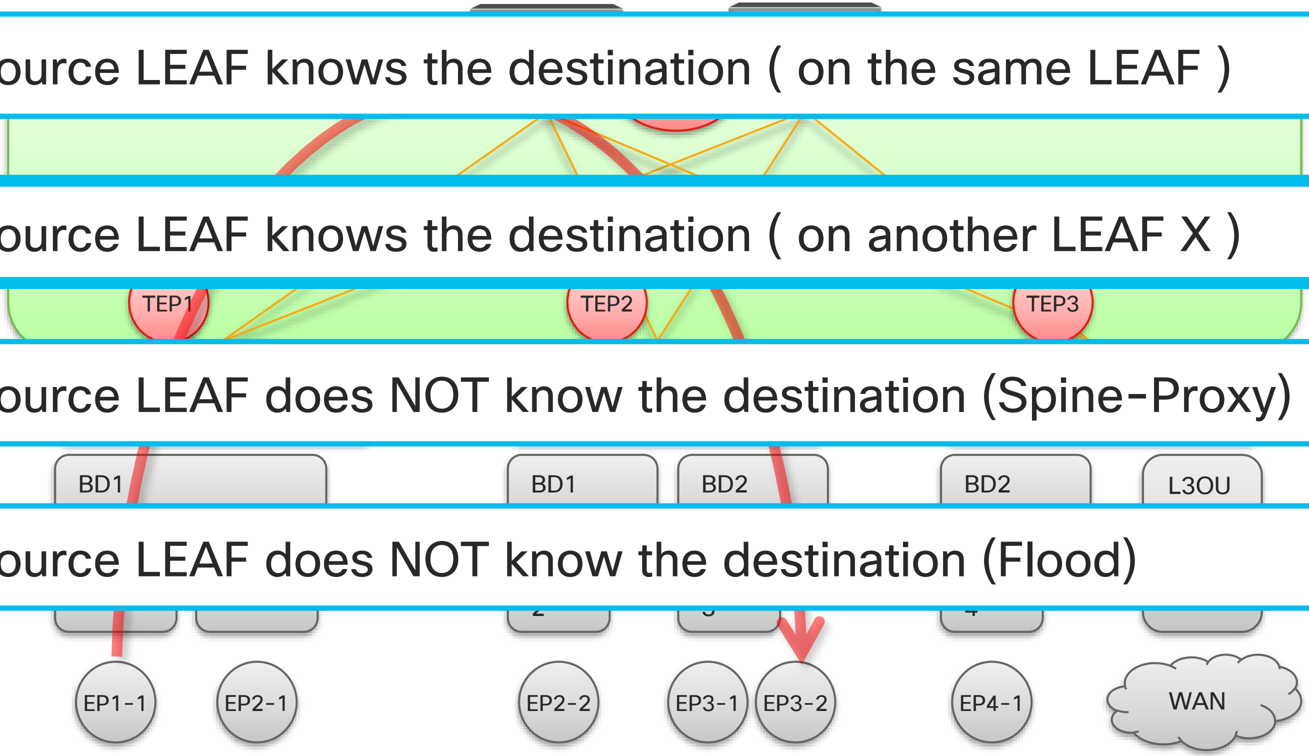
ACI Overlay VxLAN and TEP

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

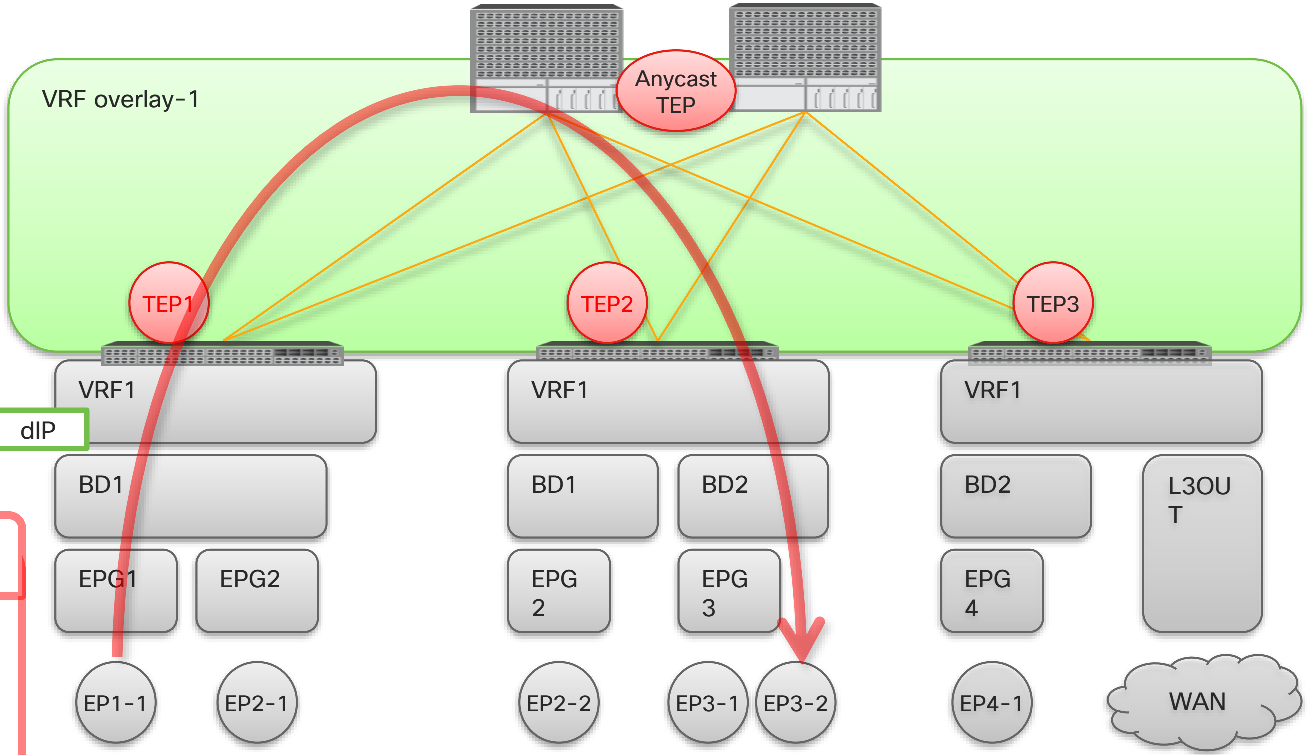
Scenario 2 : source LEAF knows the destination (on another LEAF X)

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)

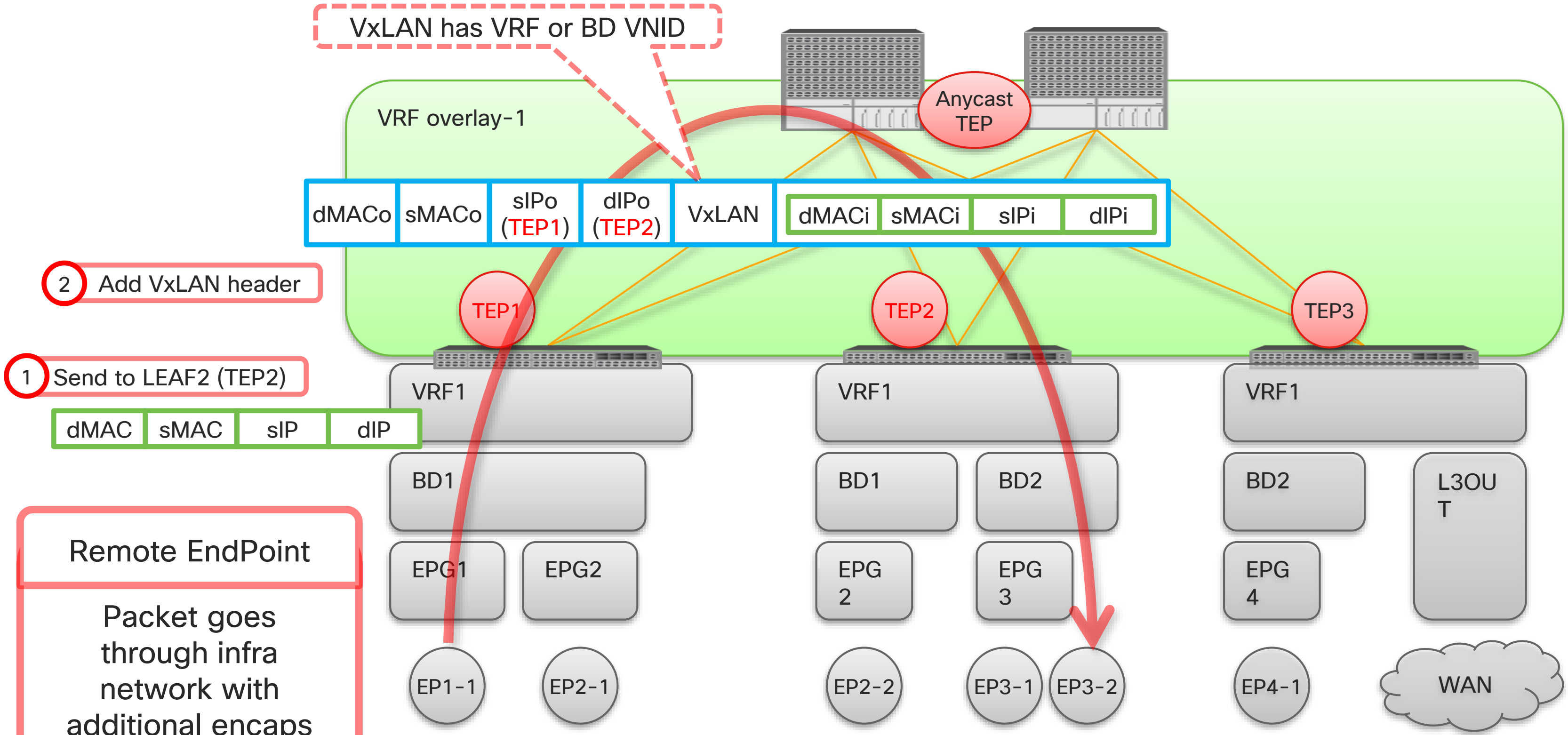
dMAC	sMAC	sIP	dIP

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)

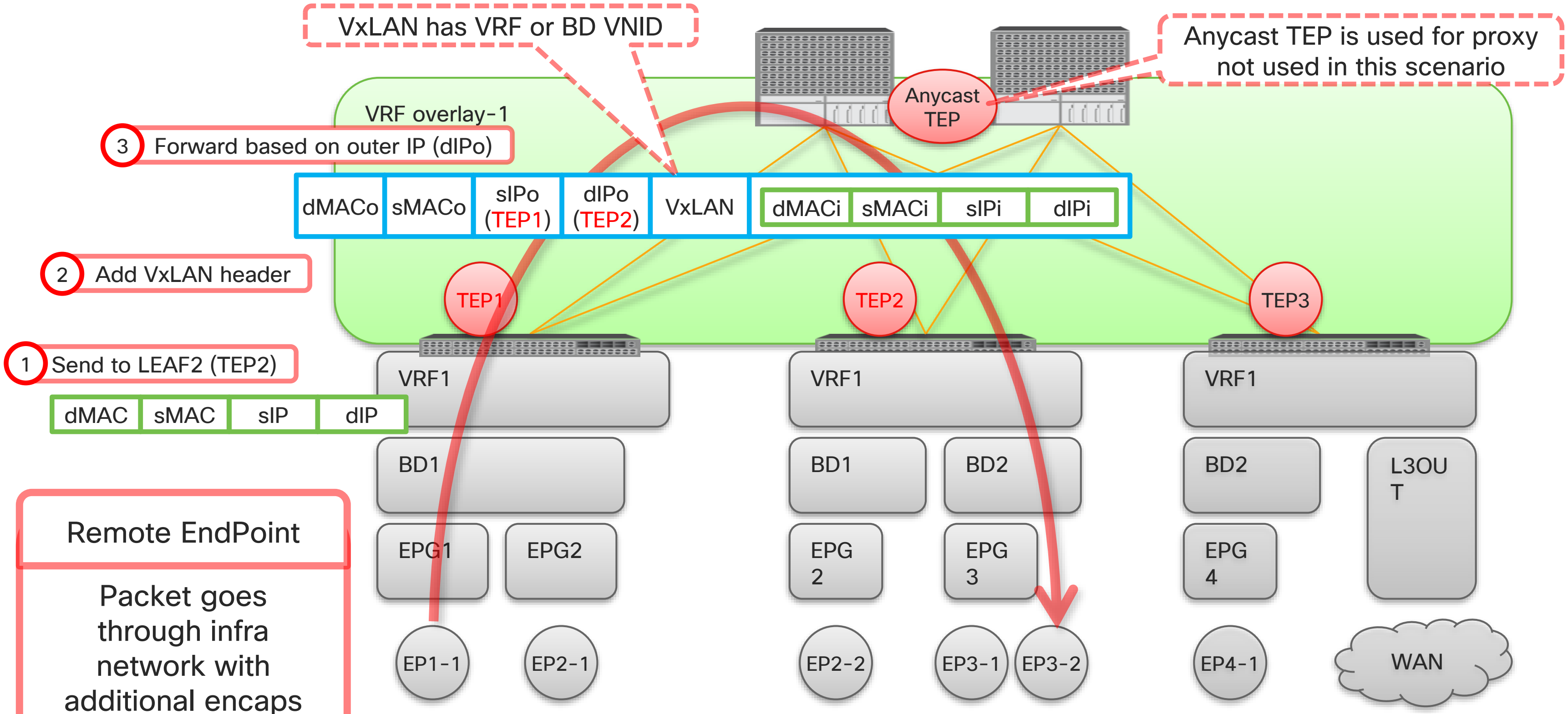


Remote EndPoint

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)

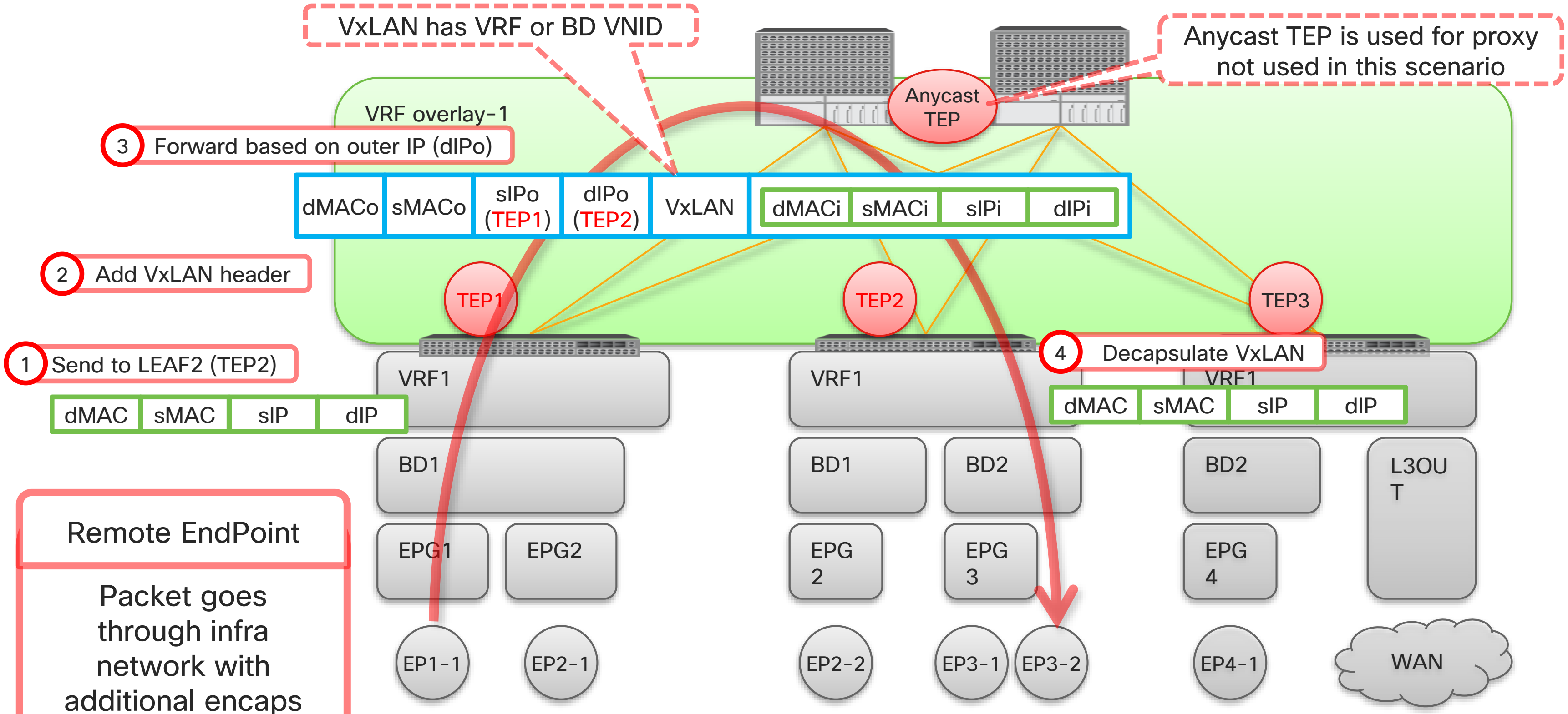


Remote EndPoint

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)



VxLAN has VRF or BD VNID

Anycast TEP is used for proxy not used in this scenario

3 Forward based on outer IP (dIPo)

2 Add VxLAN header

1 Send to LEAF2 (TEP2)

4 Decapsulate VxLAN

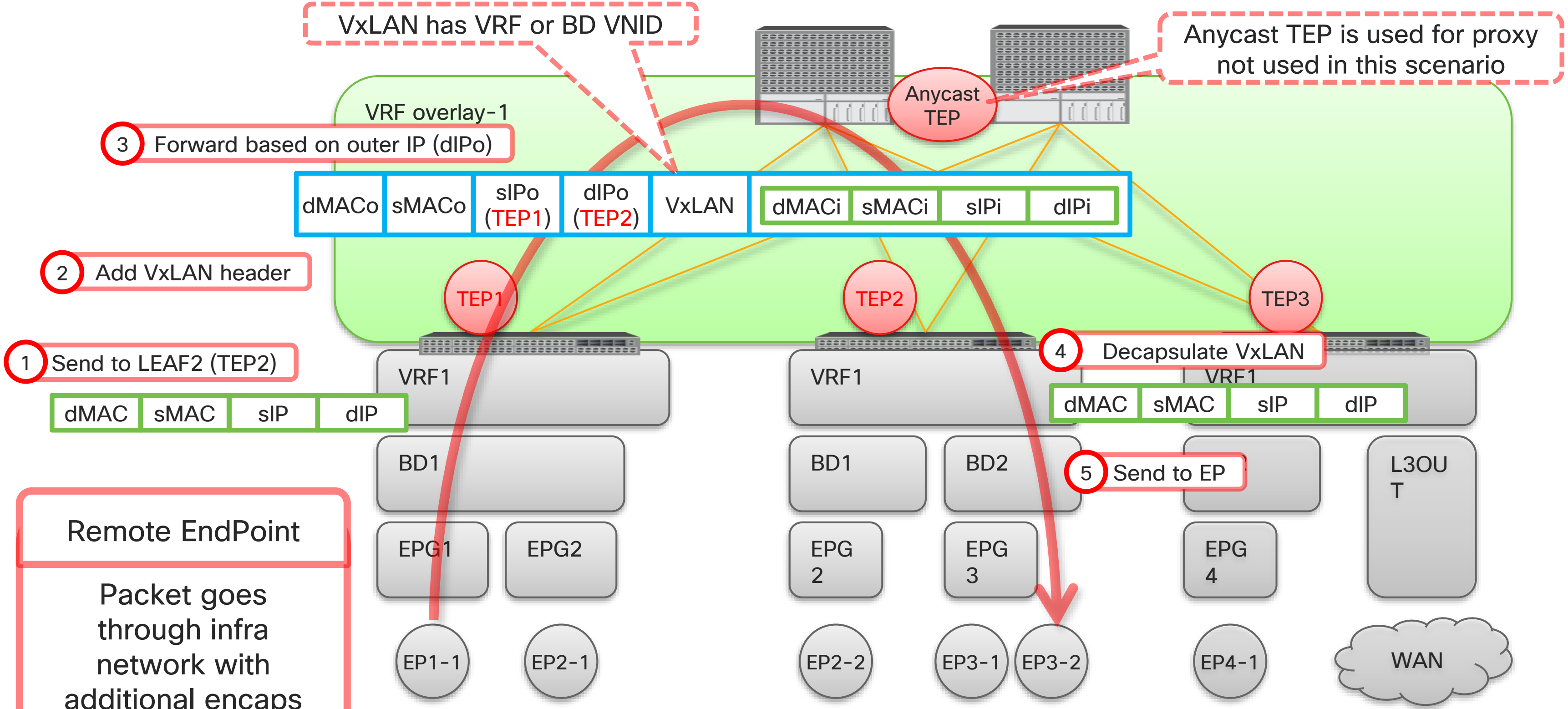
dMACo	sMACo	sIPo (TEP1)	dIPo (TEP2)	VxLAN	dMACi	sMACi	sIPi	dIPi

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)



Remote EndPoint

Packet goes through infra network with additional encaps (iVxLAN)

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

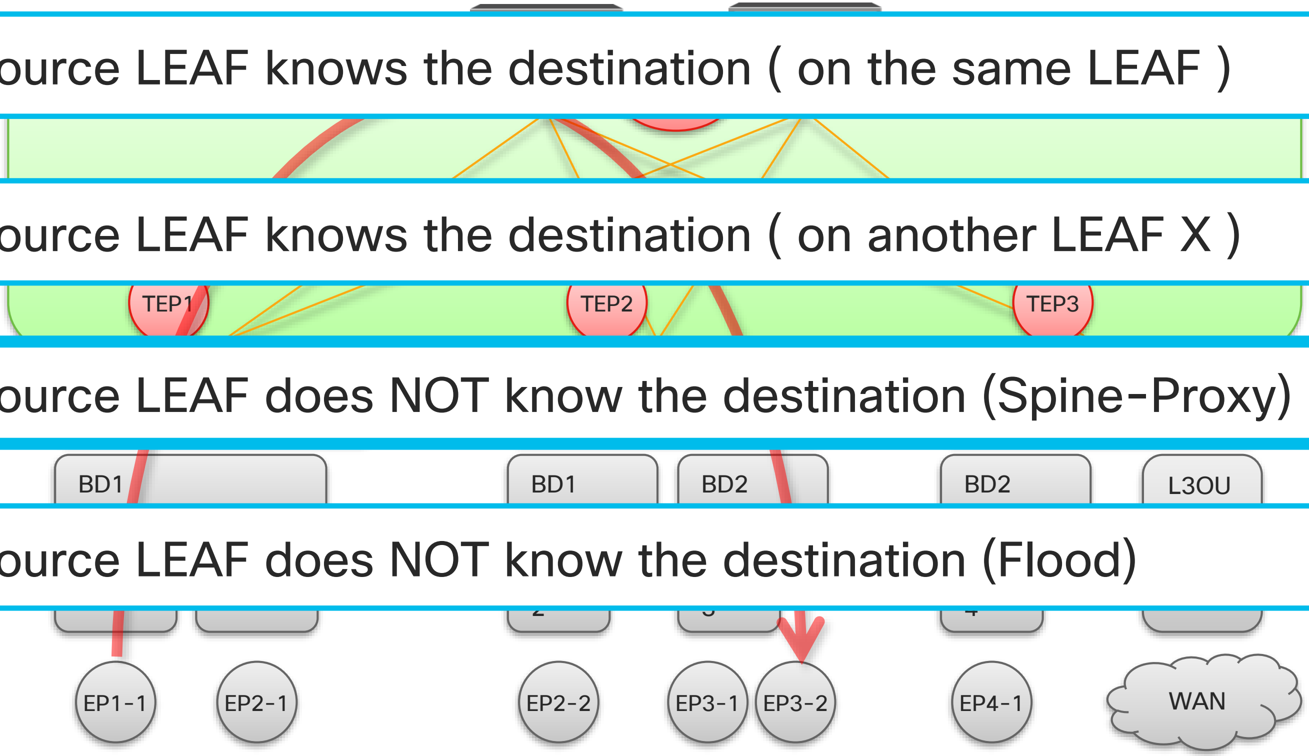
ACI Overlay VxLAN and TEP

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

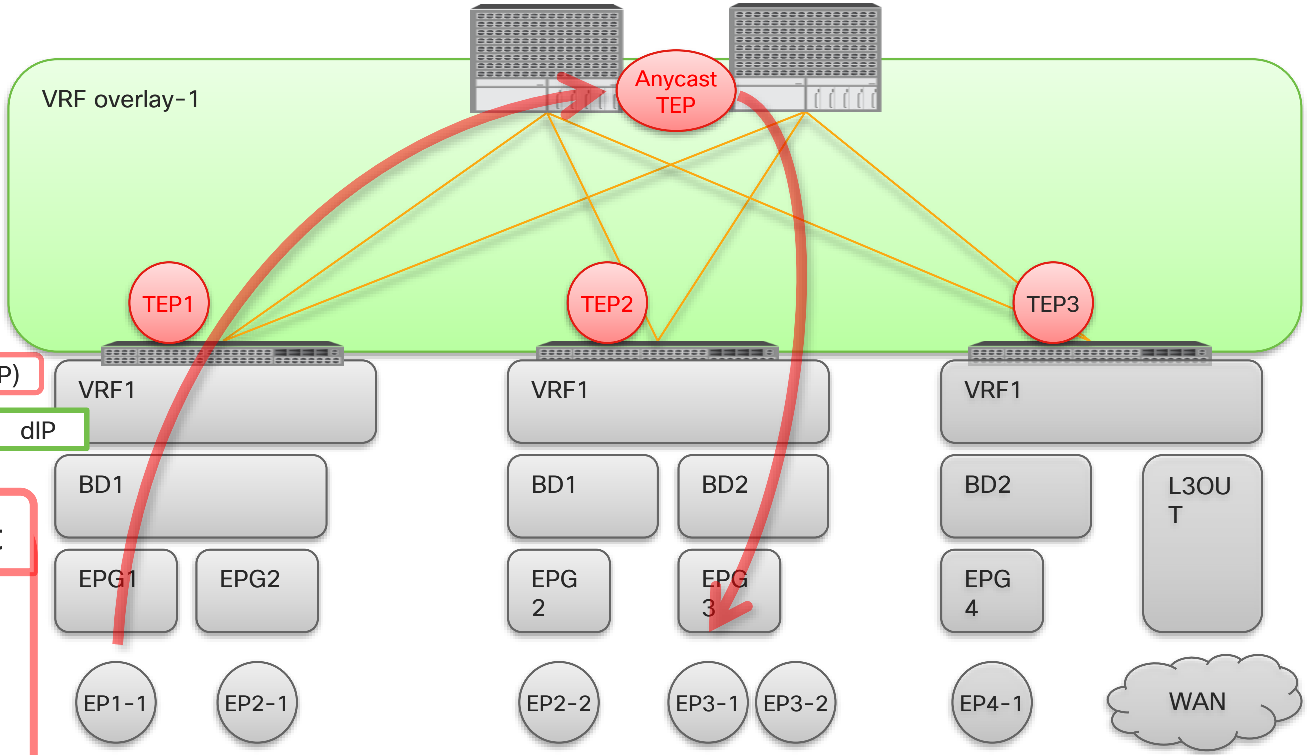
Scenario 2 : source LEAF knows the destination (on another LEAF X)

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)

3 Forward based on outer IP (dIPo)

dMACo	sMACo	sIPo (TEP1)	dIPo (acast TEP)	VxLAN	dMACi	sMACi	sIPi	dIPi

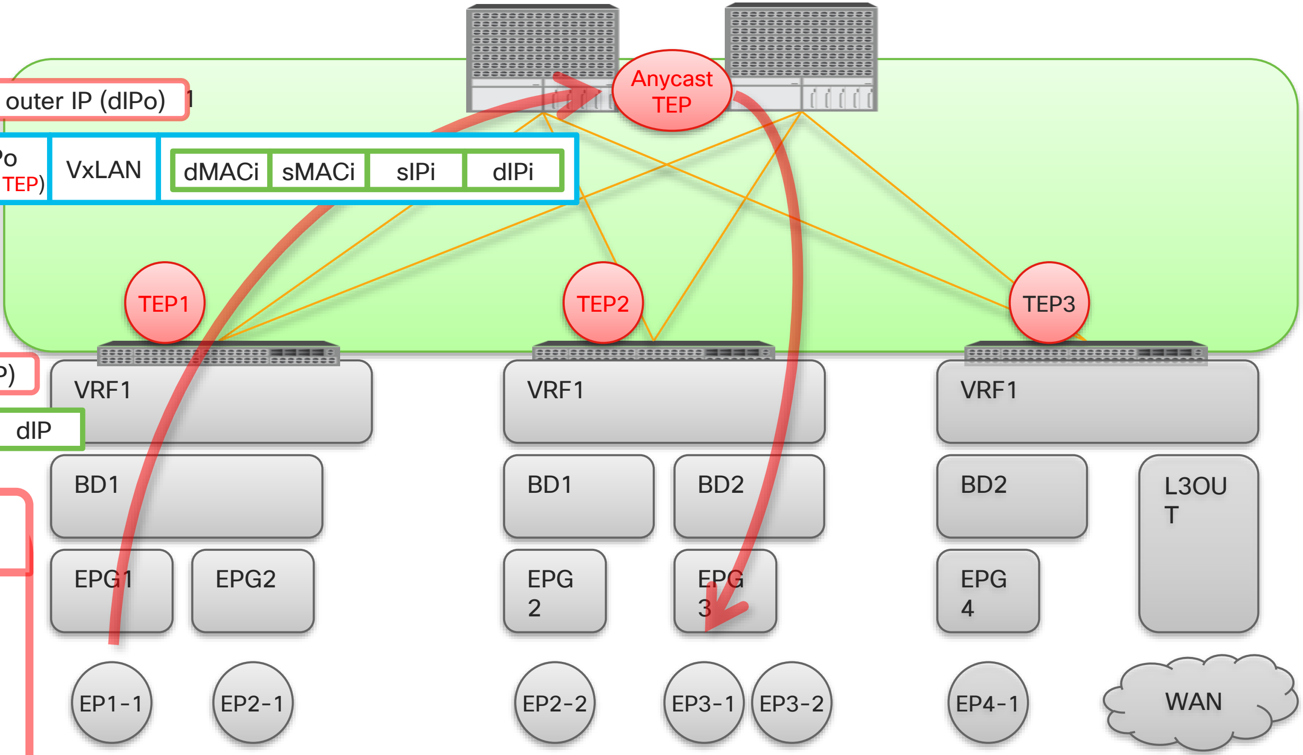
2 Add VxLAN header

1 Send to SPINE (Anycast TEP)

dMAC	sMAC	sIP	dIP

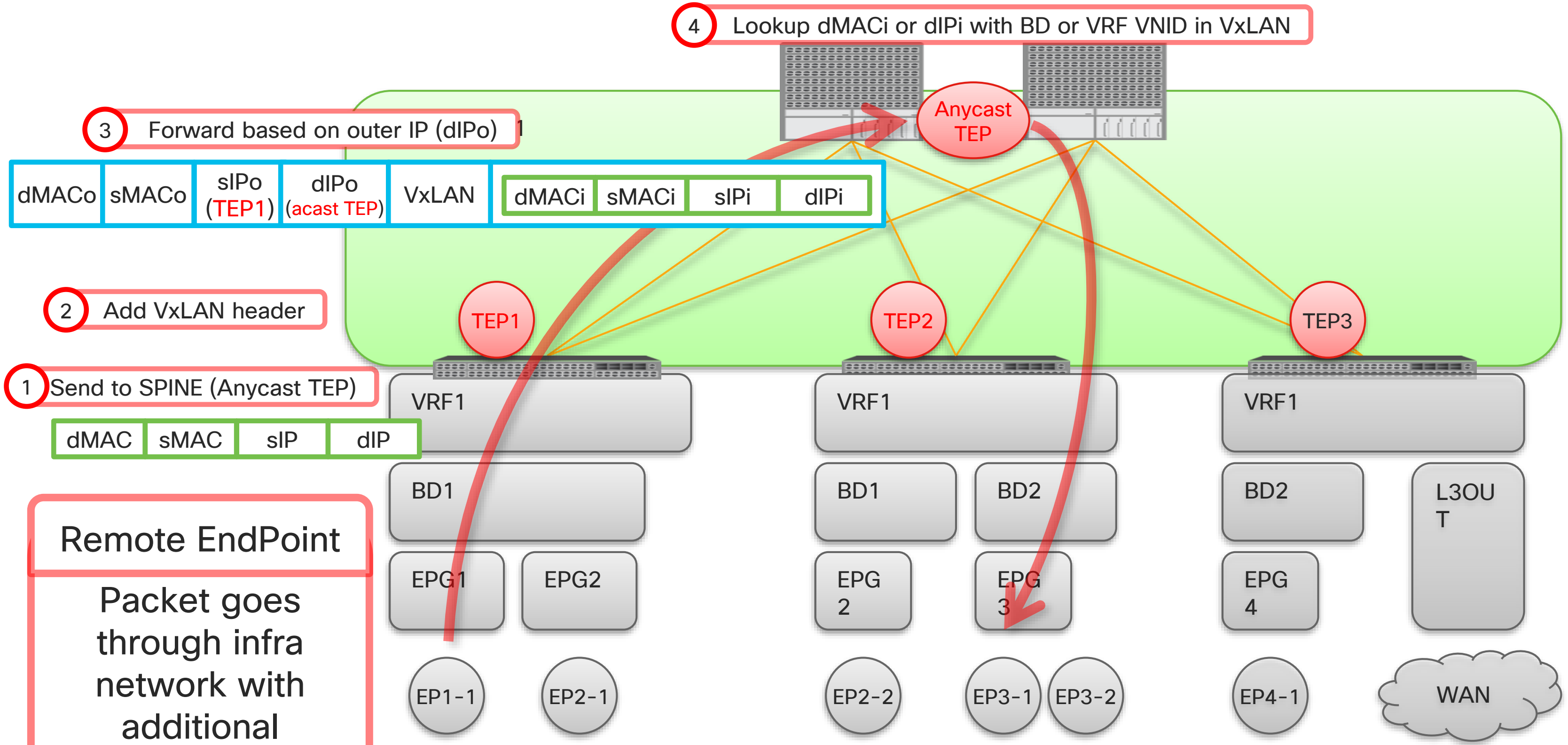
Remote EndPoint

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)

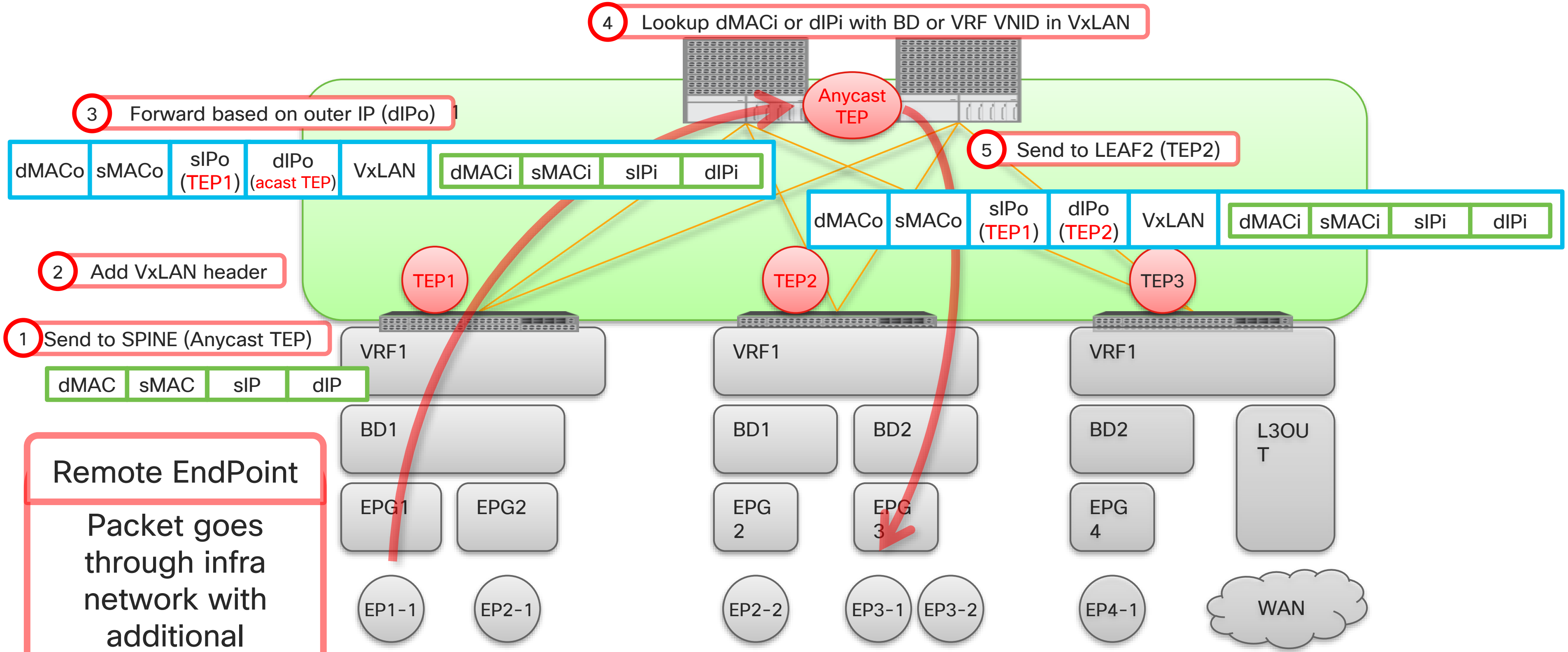


Remote EndPoint

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)

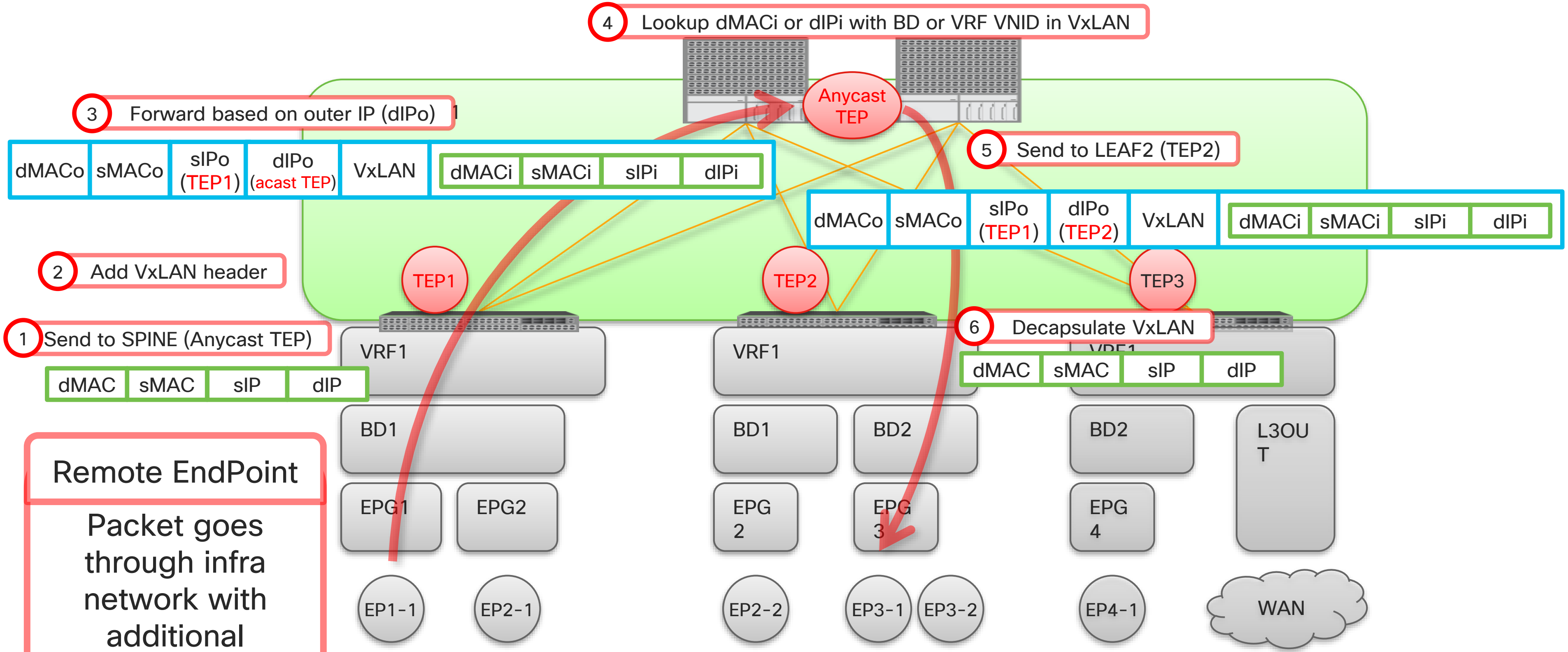


Remote EndPoint

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)

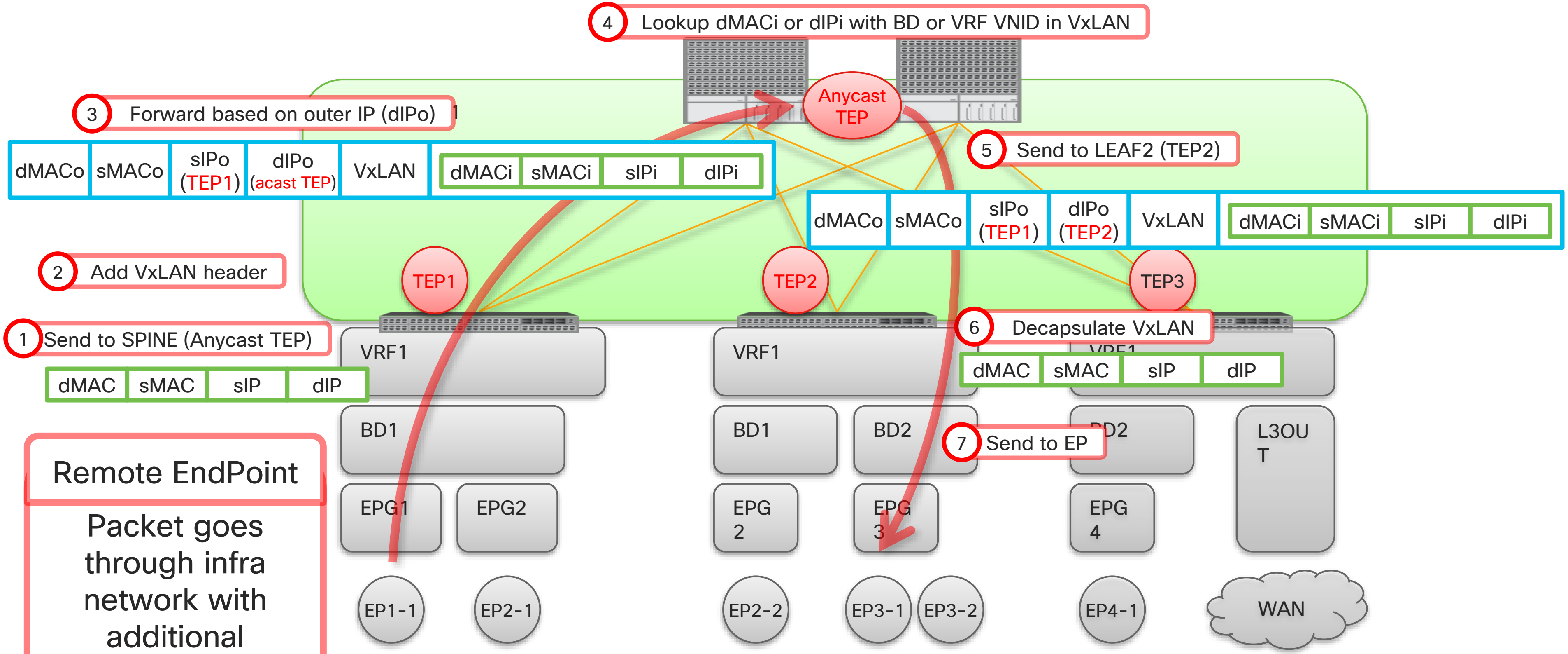


Remote EndPoint

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)

dMACo	sMACo	sIPo (TEP1)	dIPo (acast TEP)	VxLAN	dMACi	sMACi	sIPi	dIPi

5 Send to LEAF2 (TEP2)

dMACo	sMACo	sIPo (TEP1)	dIPo (TEP2)	VxLAN	dMACi	sMACi	sIPi	dIPi

2 Add VxLAN header

1 Send to SPINE (Anycast TEP)

dMAC	sMAC	sIP	dIP

6 Decapsulate VxLAN

dMAC	sMAC	sIP	dIP

7 Send to EP

Remote EndPoint

Packet goes through infra network with additional encaps (iVxLAN)

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

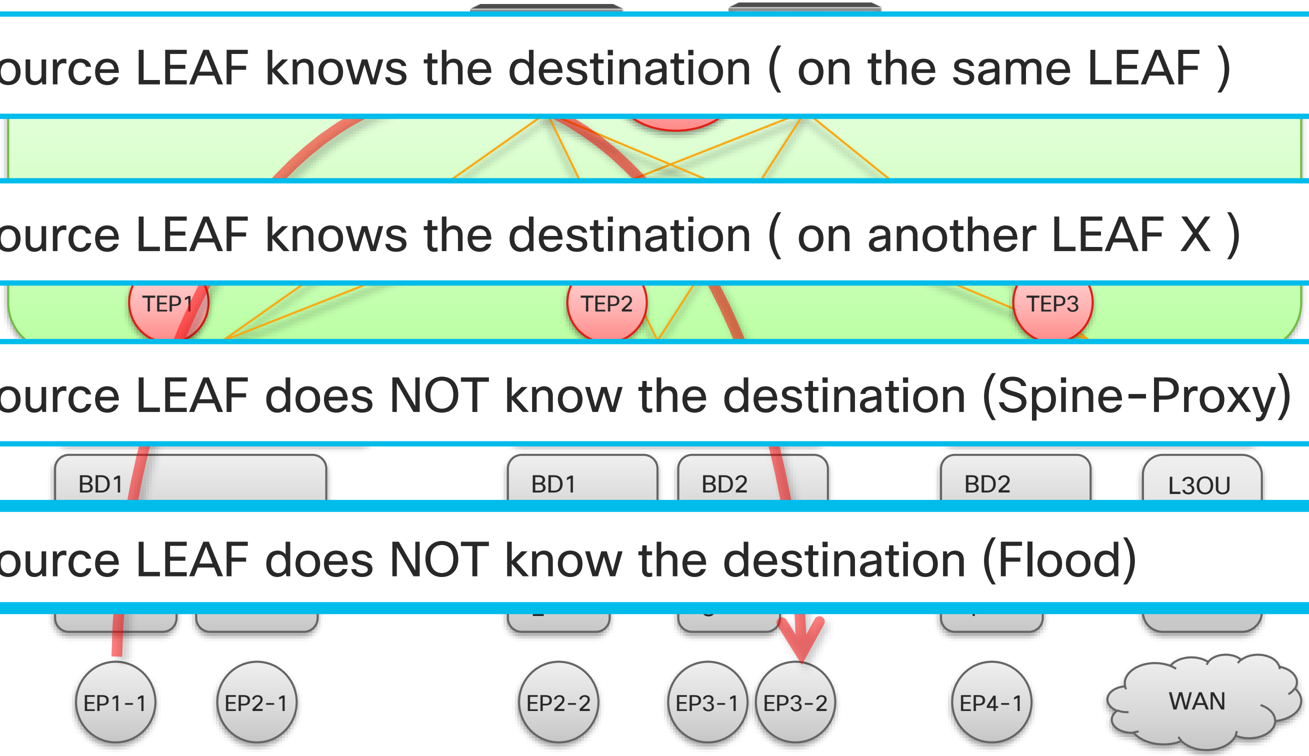
ACI Overlay VxLAN and TEP

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

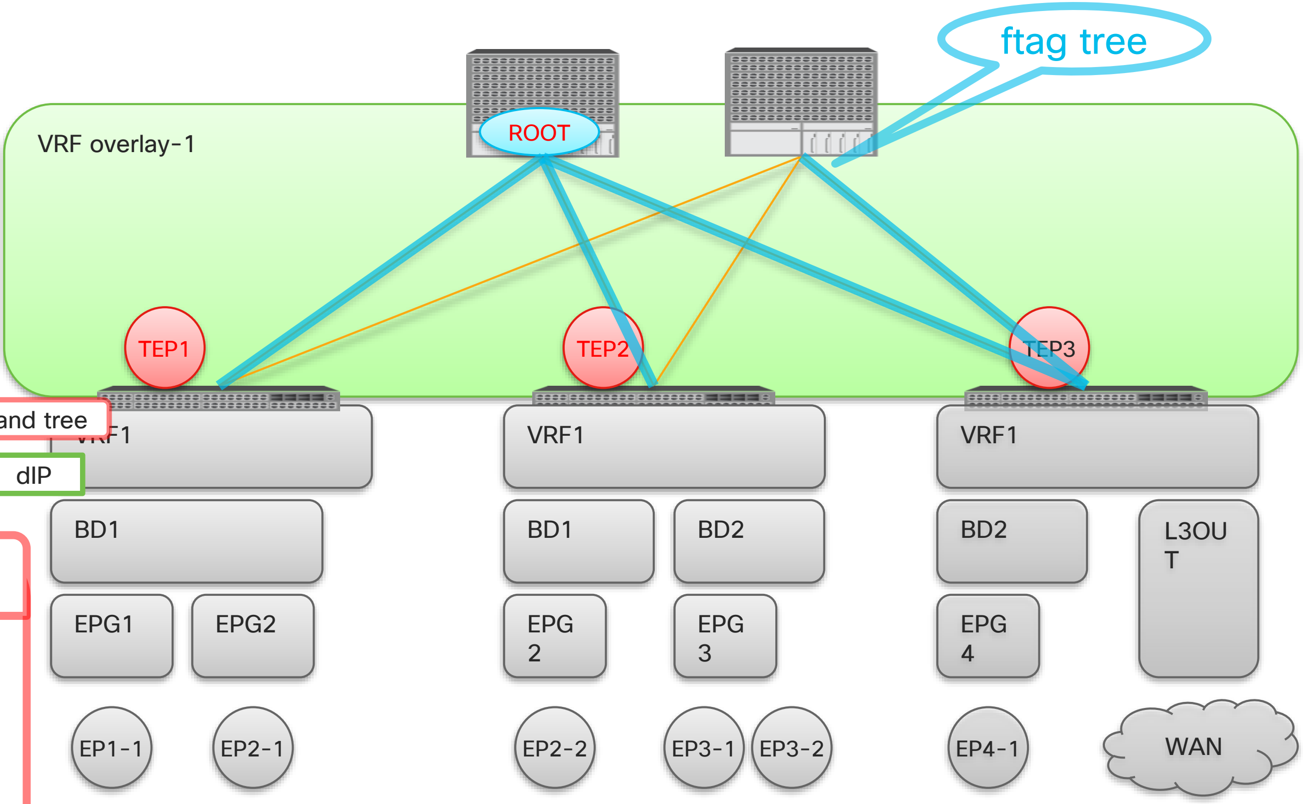
Scenario 2 : source LEAF knows the destination (on another LEAF X)

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)



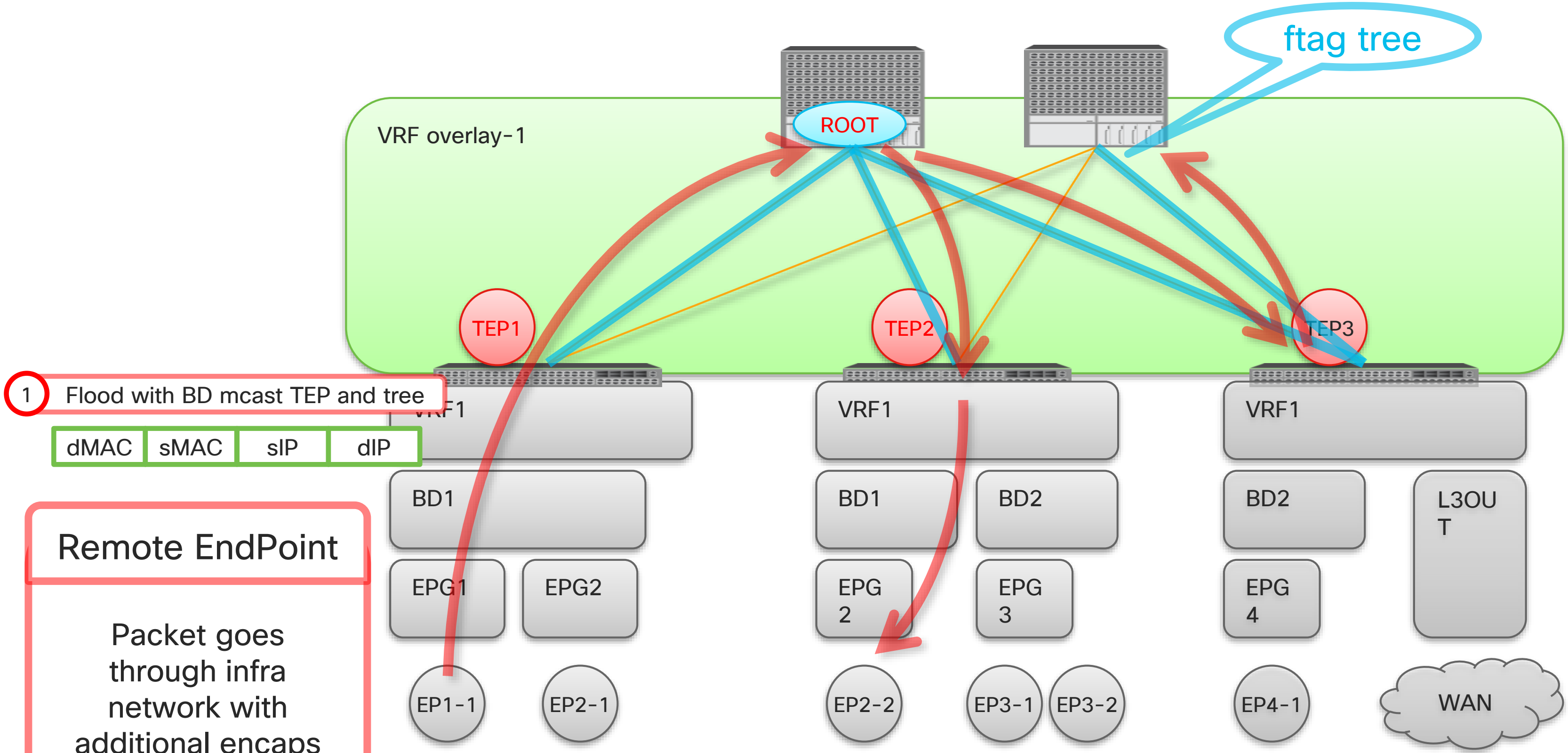
1 Flood with BD mcast TEP and tree

dMAC	sMAC	sIP	dIP

Remote EndPoint

Packet goes through infra network with additional encaps (iVxLAN)

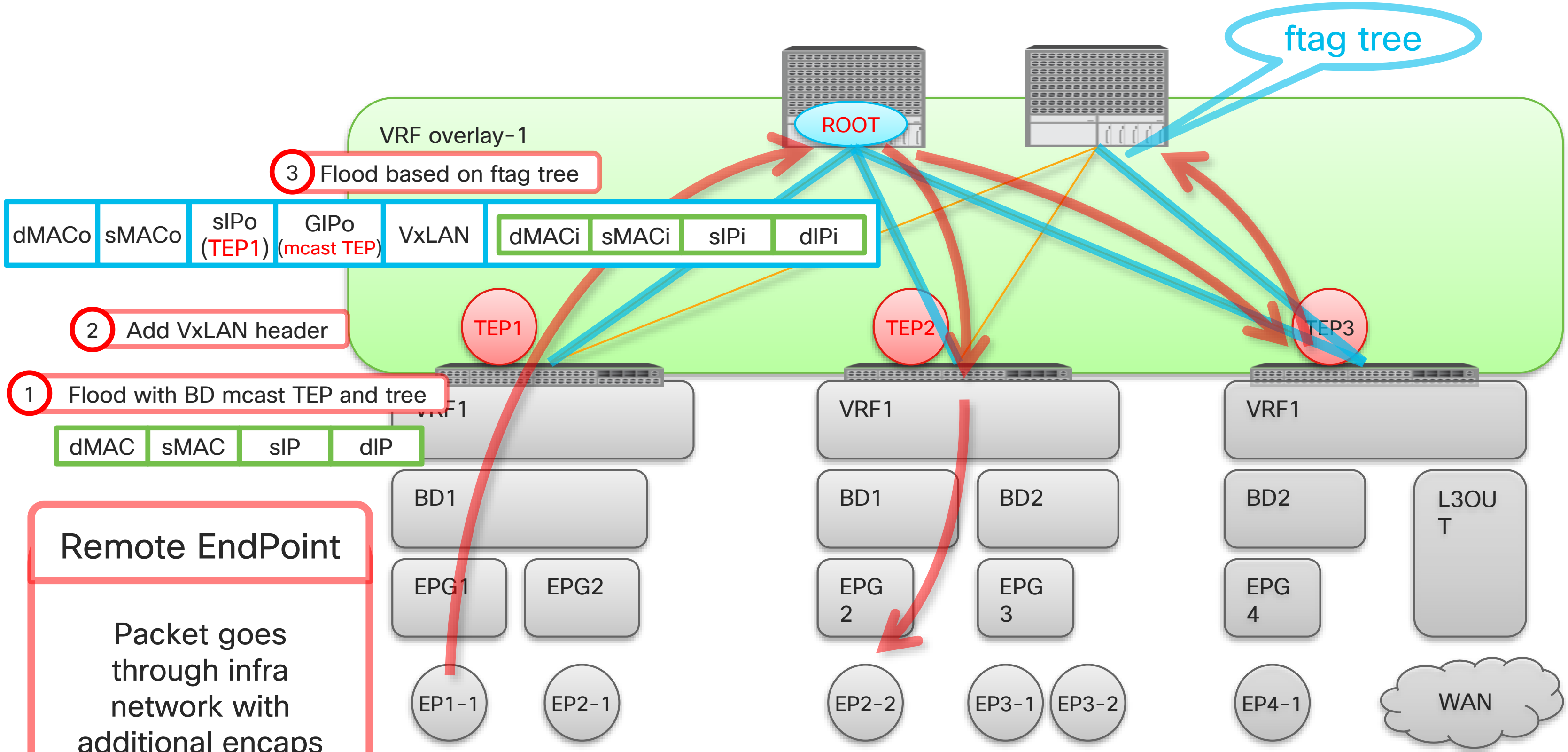
Source LEAF does NOT know the destination (Flood)



Remote EndPoint

Packet goes through infra network with additional encaps (iVxLAN)

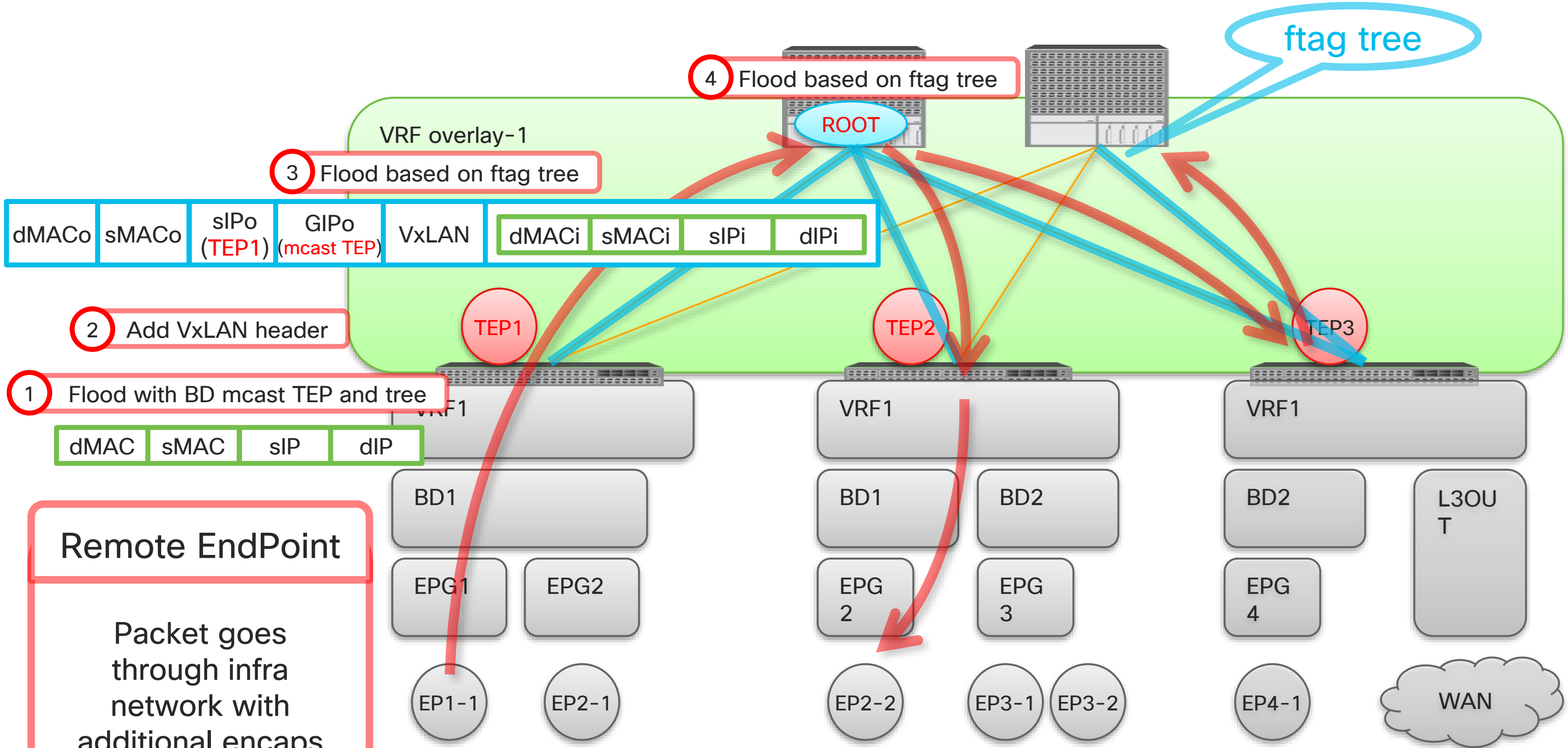
Source LEAF does NOT know the destination (Flood)



Remote EndPoint

Packet goes through infra network with additional encaps (iVxLAN)

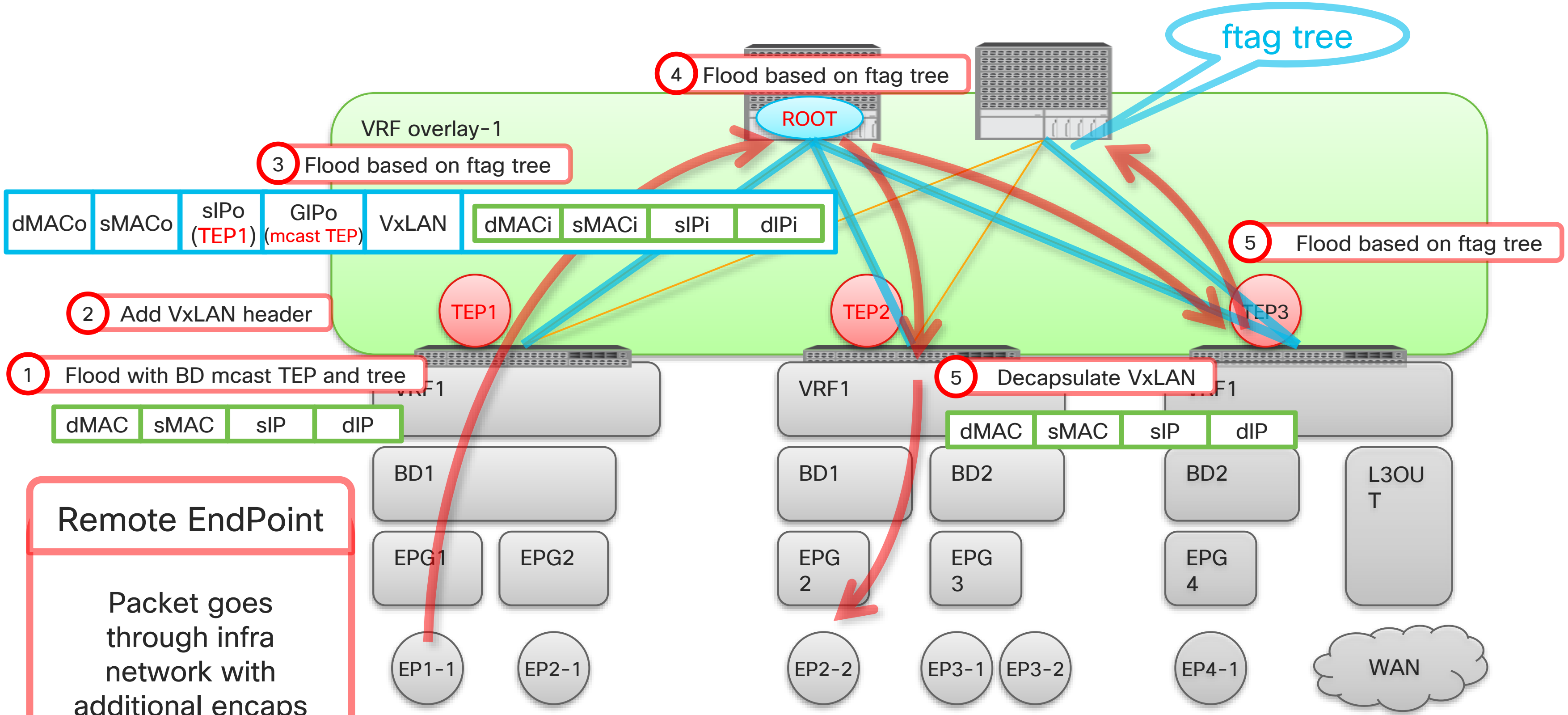
Source LEAF does NOT know the destination (Flood)



Remote EndPoint

Packet goes through infra network with additional encaps (iVxLAN)

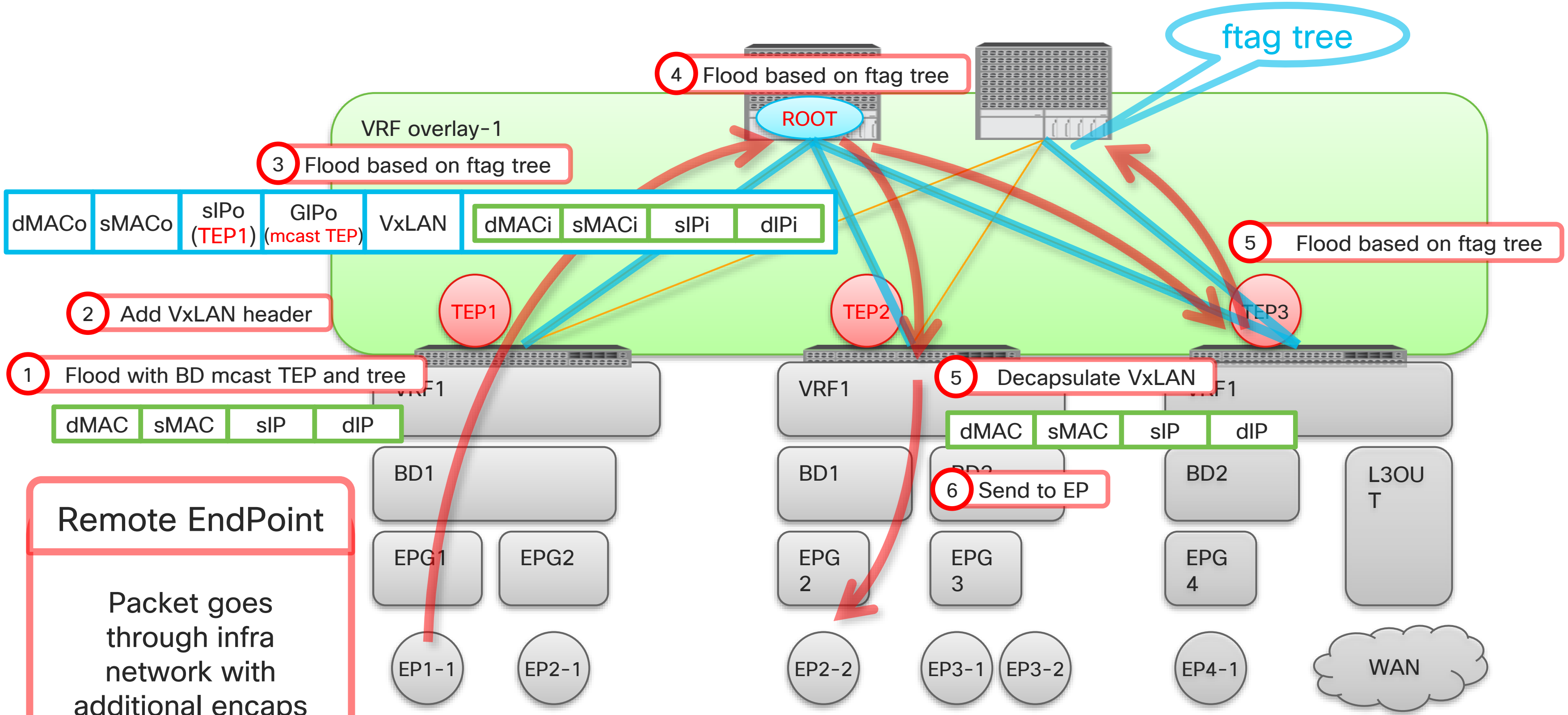
Source LEAF does NOT know the destination (Flood)



Remote EndPoint

Packet goes through infra network with additional encaps (iVxLAN)

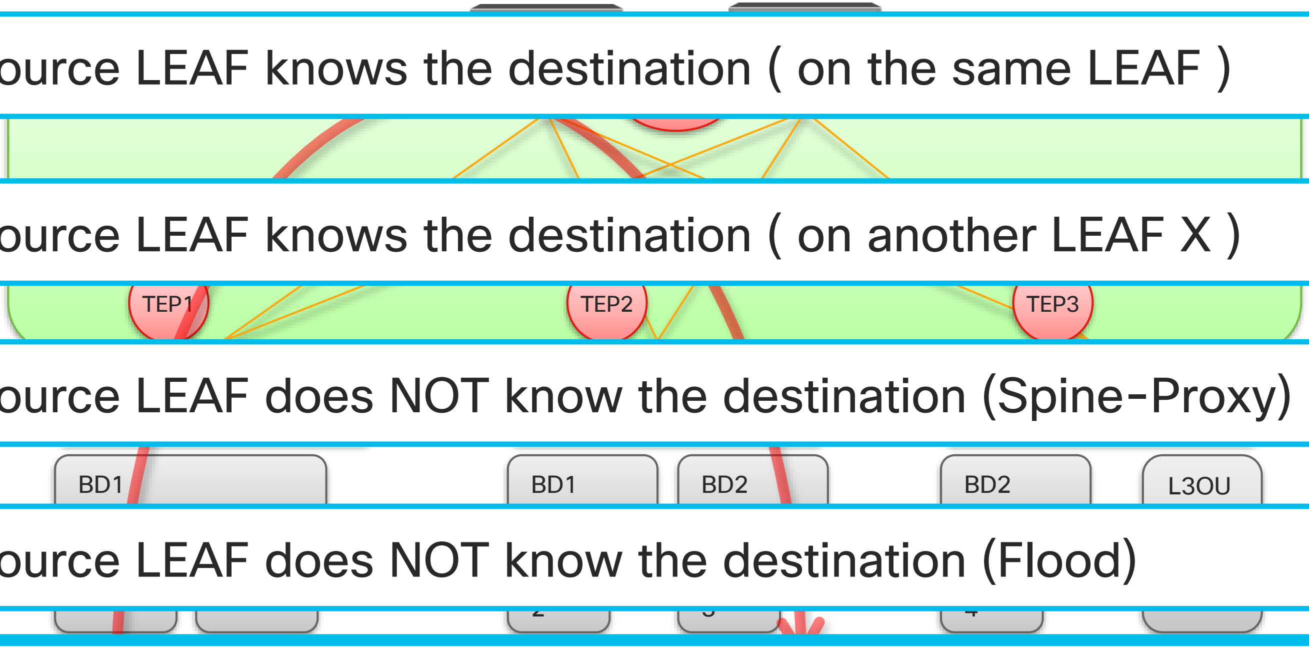
Source LEAF does NOT know the destination (Flood)



Remote EndPoint

Packet goes through infra network with additional encaps (iVxLAN)

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
  - Packet Capture in ACI

ACI Forwarding Component 1

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

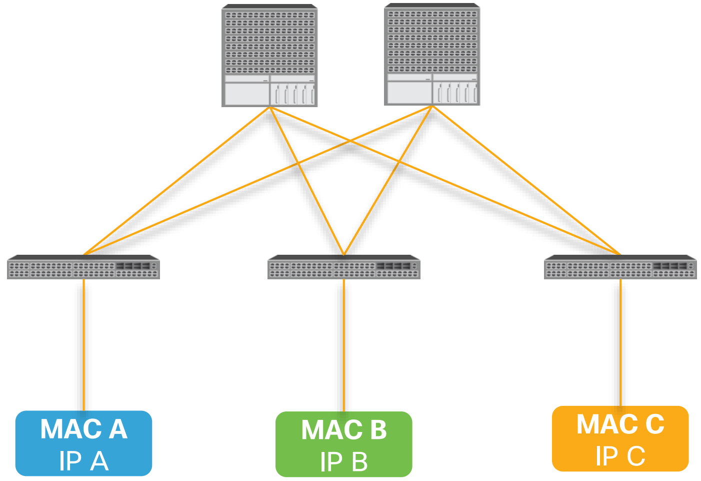
End Point (EP)

What is an EP?

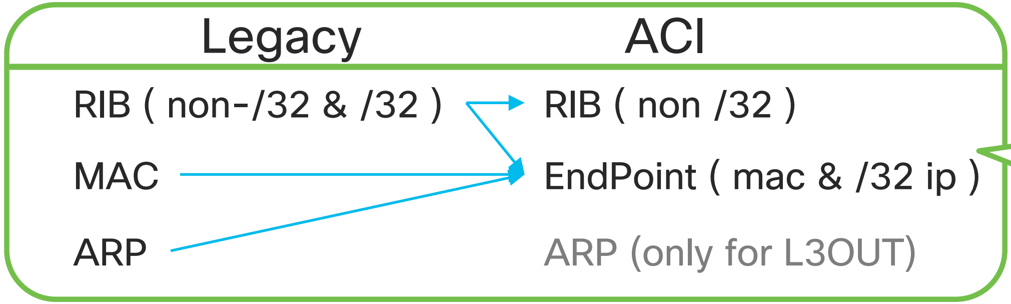
- It stands for hosts, in other words MAC address with IP(s)
  - ➢ sometimes MAC only
  - ➢ IP in EP is always /32

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)



These are End Points



Legacy	ACI
RIB (non-/32 & /32)	RIB (non /32)
MAC	EndPoint (mac & /32 ip)
ARP	ARP (only for L3OUT)

Forwarding table lookup order

1. EndPoint Table (show endpoint)
2. RIB (show ip route)

RIB : Routing Information Base

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?

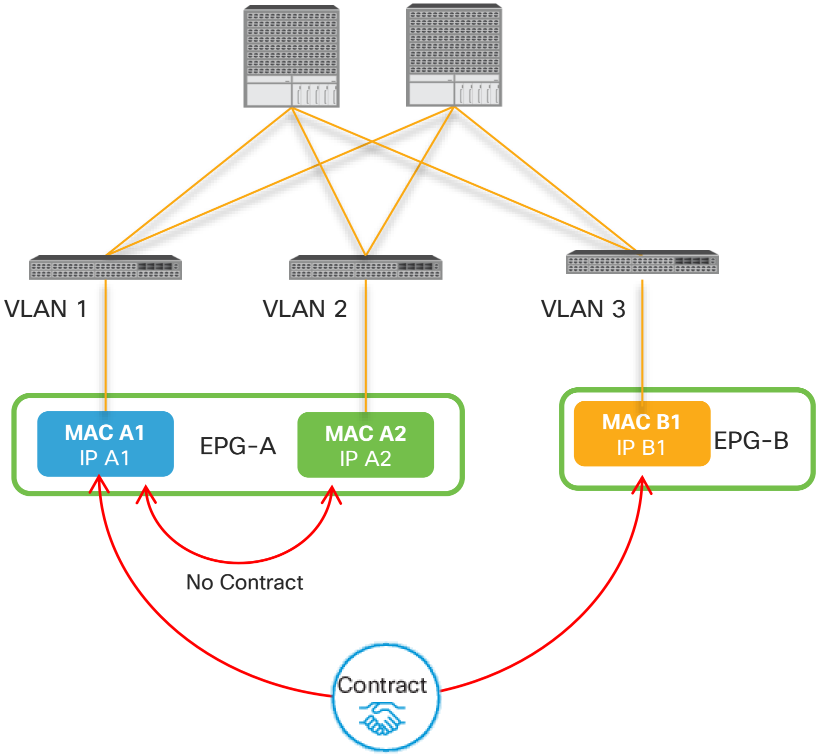
- **To implement traffic filtering**
  - ➢ Traffic within the same EPG doesn't get blocked
  - ➢ Traffic across EPGs always blocked without a contract
- A contract is applied between EPGs to allow traffic

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

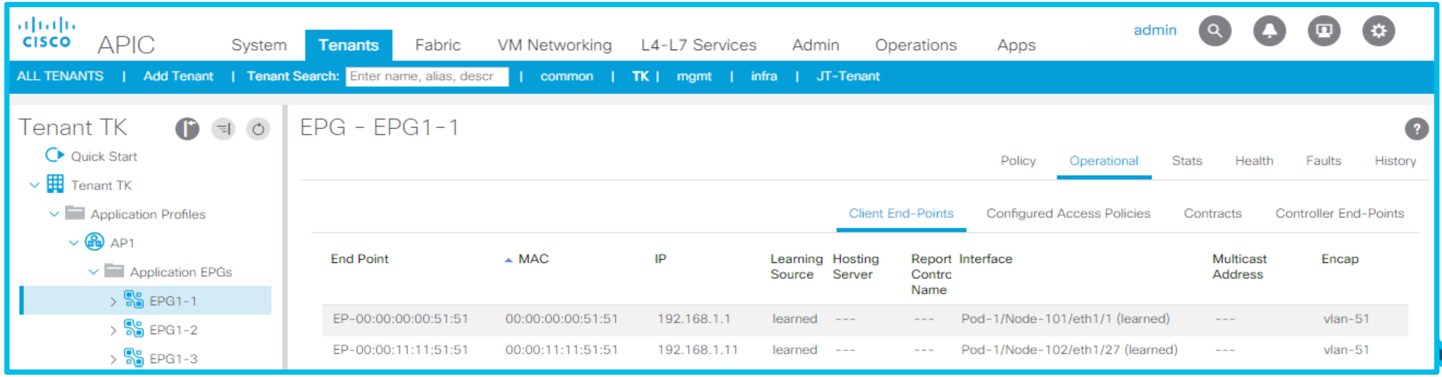
What happens with forwarding?

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



How to check End Points

From APIC GUI (Fabric perspective)

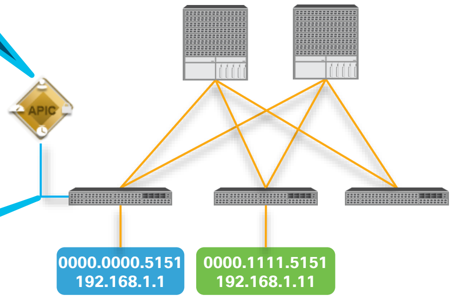


EPG - EPG1-1

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/                              Encap           MAC Address       MAC Info/       Interface
      Domain                             VLAN            IP Address        IP Info
+-----------------------------------+---------------+-----------------+--------------+-------------+
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
```

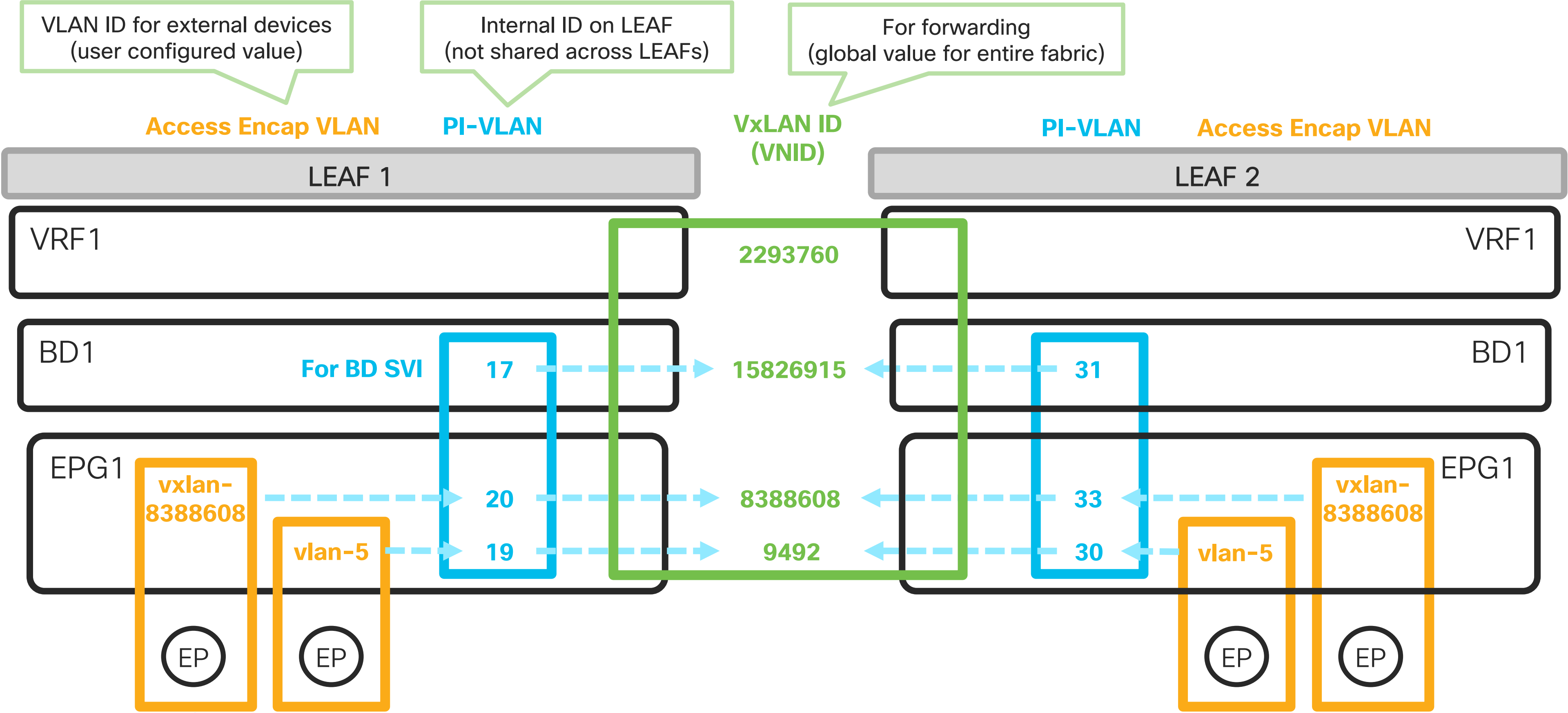
Good for forwarding verification

shows how EPs look from each LEAF

cisco Live!

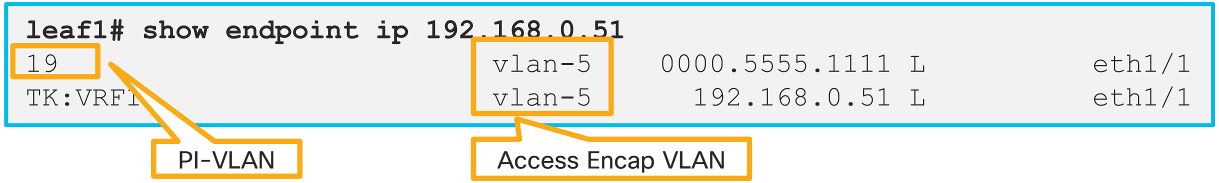
VLAN types in ACI

※ PI-VLAN : Platform Independent VLAN



PI-VLAN for EPG and BD CLI

- Endpoint Table

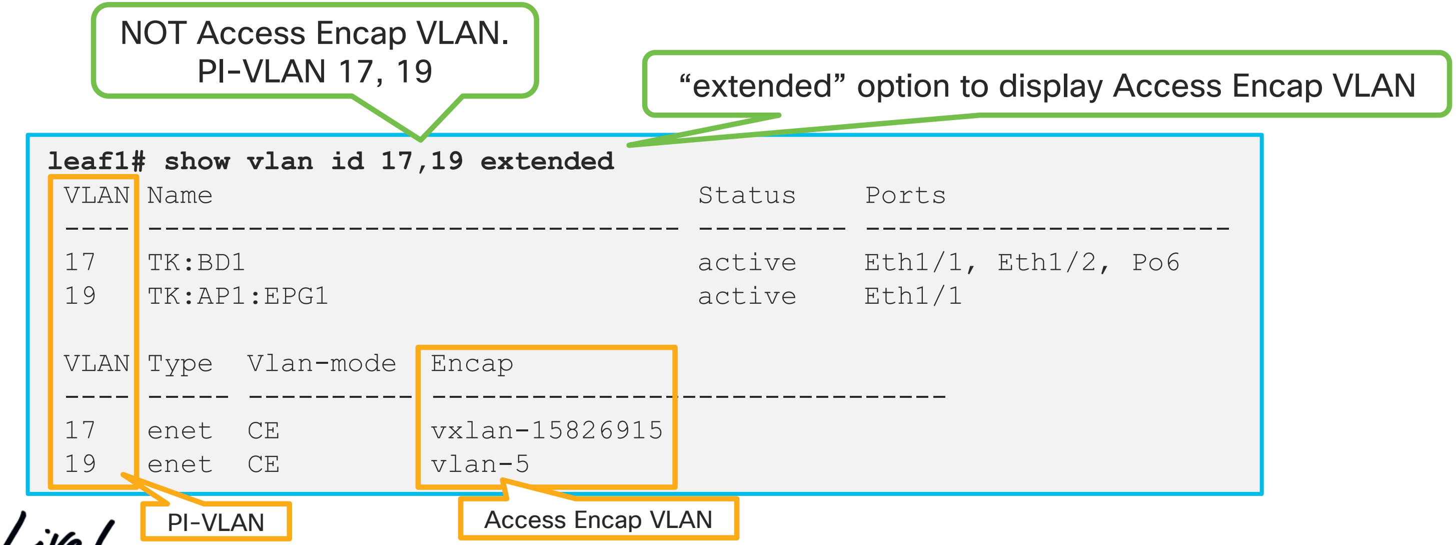


```
leaf1# show endpoint ip 192.168.0.51
19                           vlan-5   0000.5555.1111 L          eth1/1
TK:VRF1                      vlan-5     192.168.0.51 L          eth1/1
```

PI-VLAN

Access Encap VLAN

- VLAN Table



NOT Access Encap VLAN.
PI-VLAN 17, 19

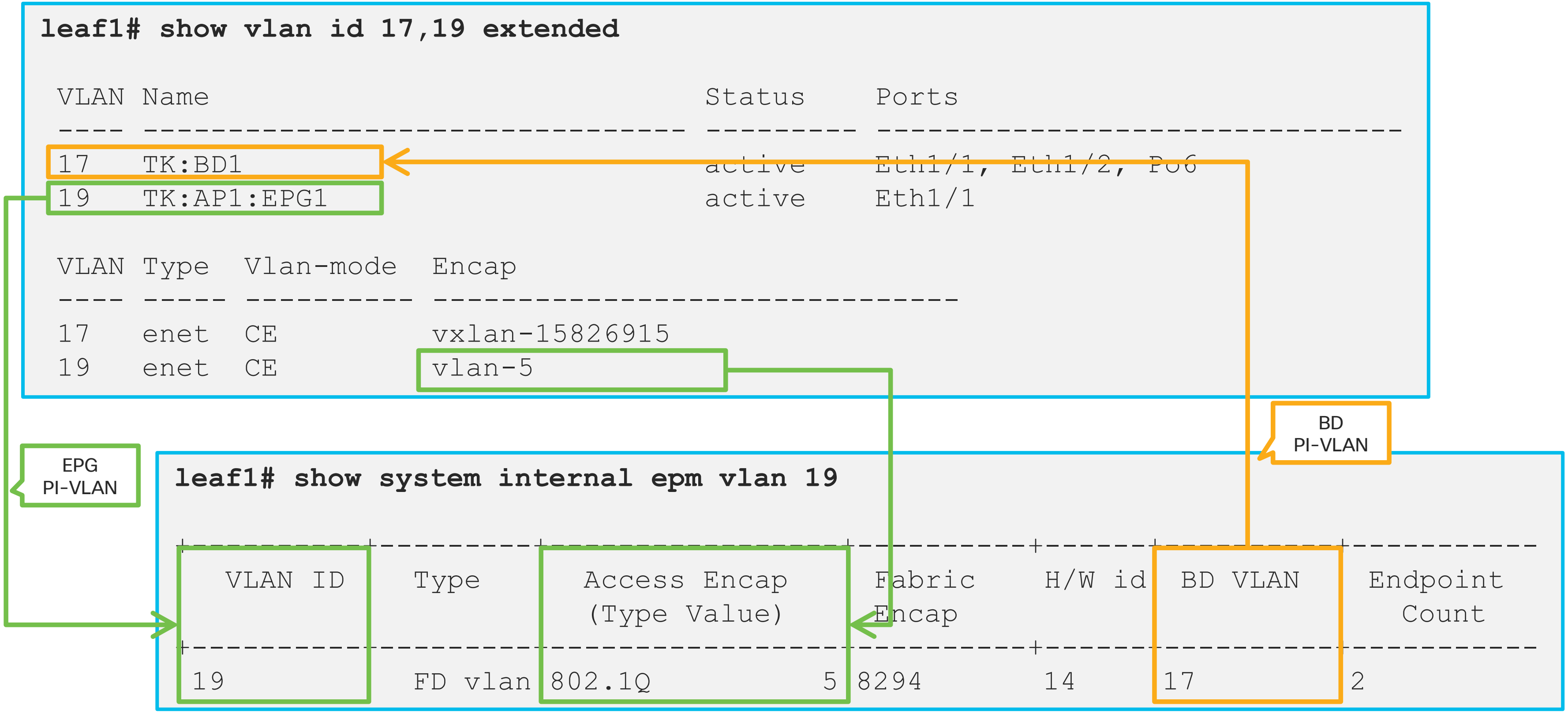
“extended” option to display Access Encap VLAN

```
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

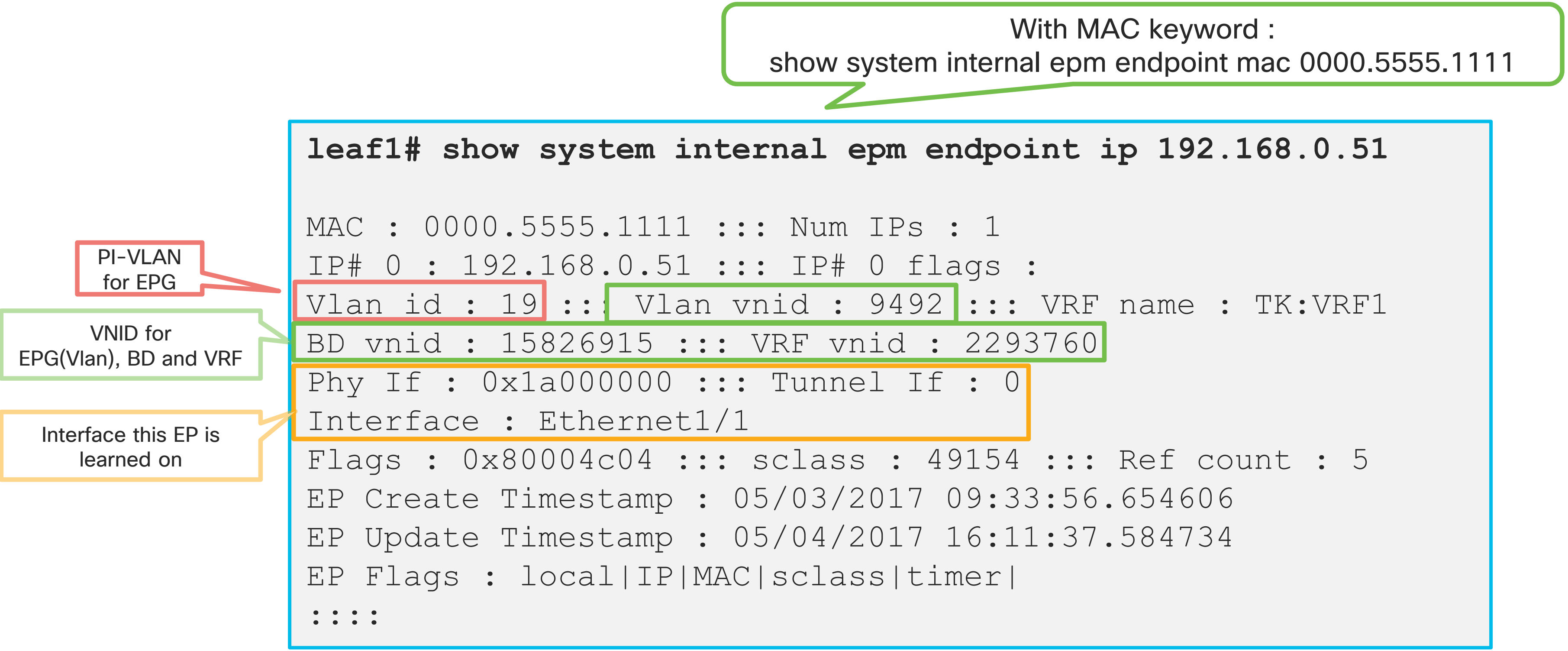
PI-VLAN for EPG and BD CLI



```
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
```

```
leaf1# show system internal epm vlan 19

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

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

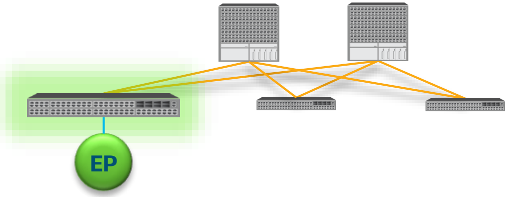
all commands on

```
Legend:
 O - peer-attached
 L - local
```

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
```

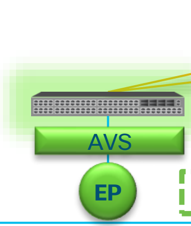


Virtual Local Endpoint (VL)

- 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
```

Access Encap VLAN (VxLAN)



AVS/AVE is treated as virtual LEAF with vTEP

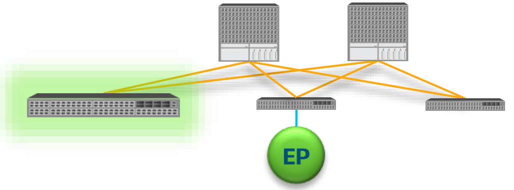
Remote Endpoint (Xr)

- An endpoint on another LEAF

BD VNID (not Access Encap VLAN)

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

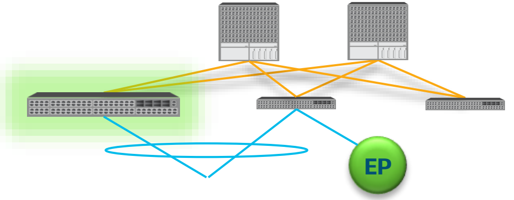
```
fab1-leaf1# show endpoint ip 192.168.0.52
TK:VRF1                                        192.168.0.52             tunnel8
```



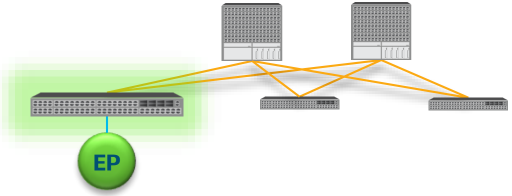
On-Peer Endpoint

- 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
```



End Point Learning (Local EP)



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

VRF to which MAC & IP belong

```
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
```

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)

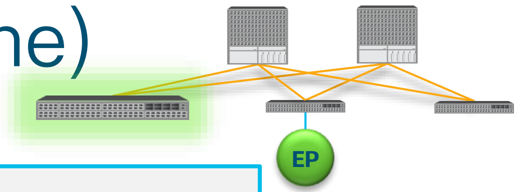
  or

- if **ARP request** with sender IP A comes in from its **front panel port**.

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

BD VNID

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

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)

dMACo	sMACo	sIPo	dIPo	VxLAN	dMACi	sMACi	sIPi	dIPi

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 fnvread
      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/dom-1
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

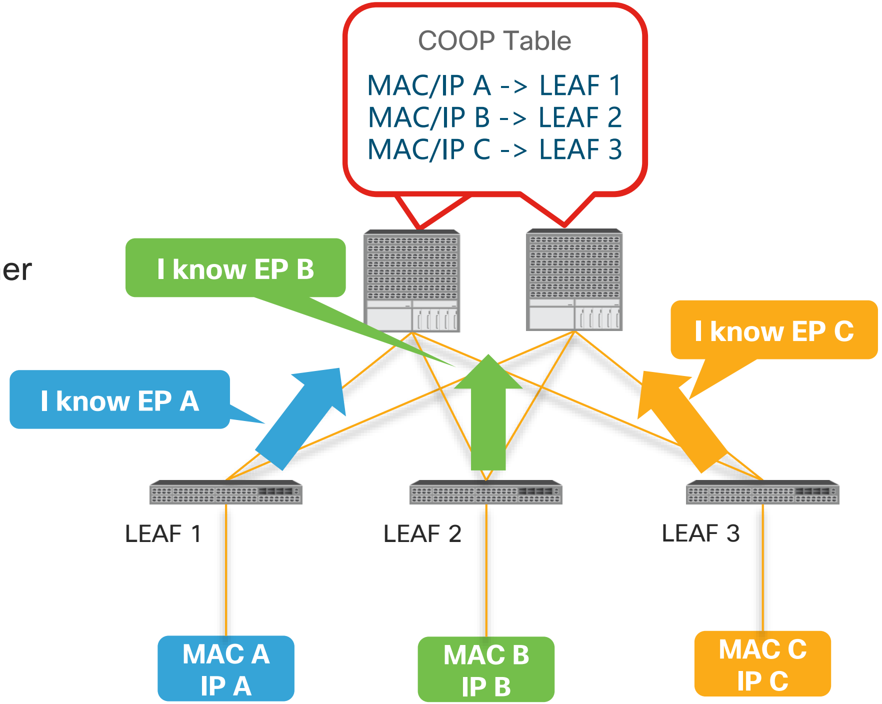
1. LEAF learns EP (either MAC or/and IP) as local
2. LEAF reports local EP to Spine via COOP process
3. SPINE stores these in COOP DB and synchronize with other SPINEs

What is the purpose of COOP?

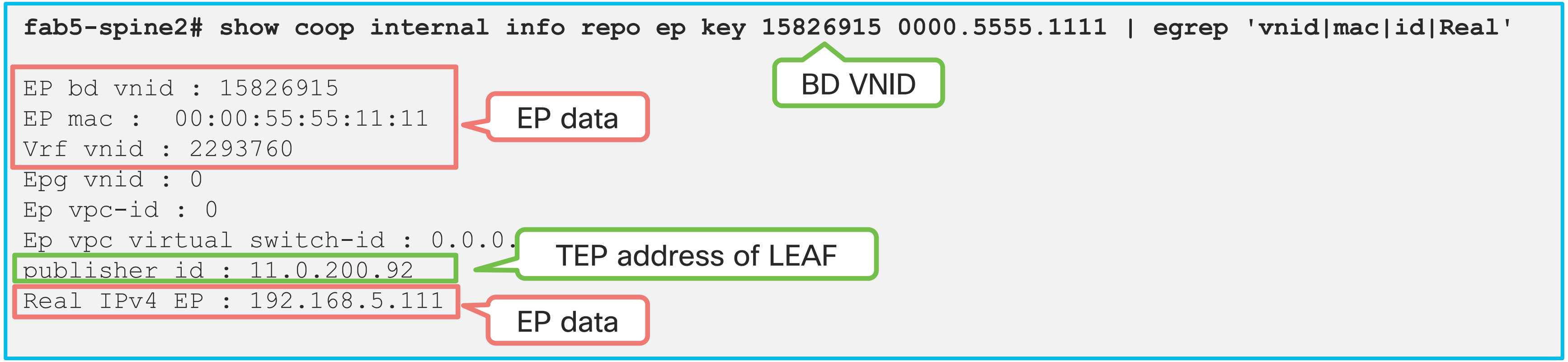
When Leaf doesn't know dst EP, LEAF can forward packet to Spine in order to let Spine decide where to send. This behavior is called Spine-Proxy.

Note :

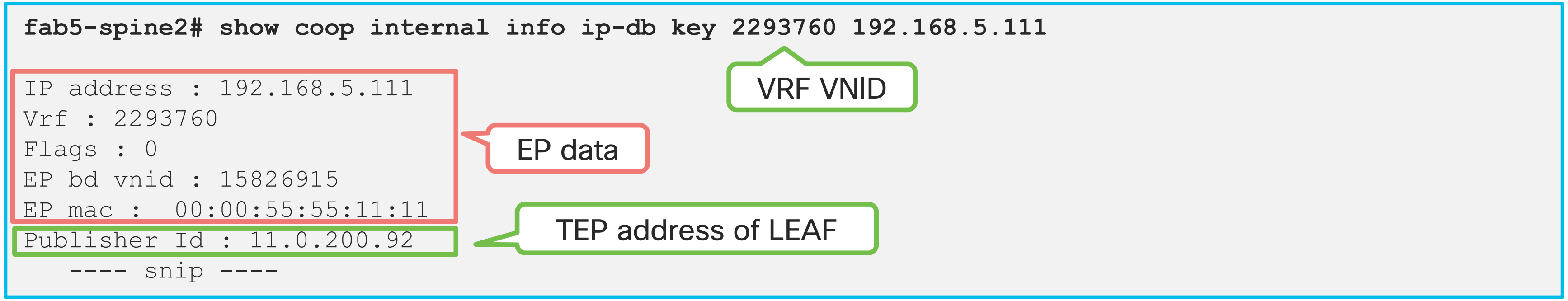
- Normally SPINE doesn't push COOP DB entries to each LEAF. It just receives and stores. The exception is for bounce entries.
- Remote Endpoints are stored on each Leaf nodes as cache. This is not reported to Spine COOP.



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
Epg vnid : 0
Ep vpc-id : 0
Ep vpc virtual switch-id : 0.0.0.0
publisher id : 11.0.200.92
Real IPv4 EP : 192.168.5.111
```



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

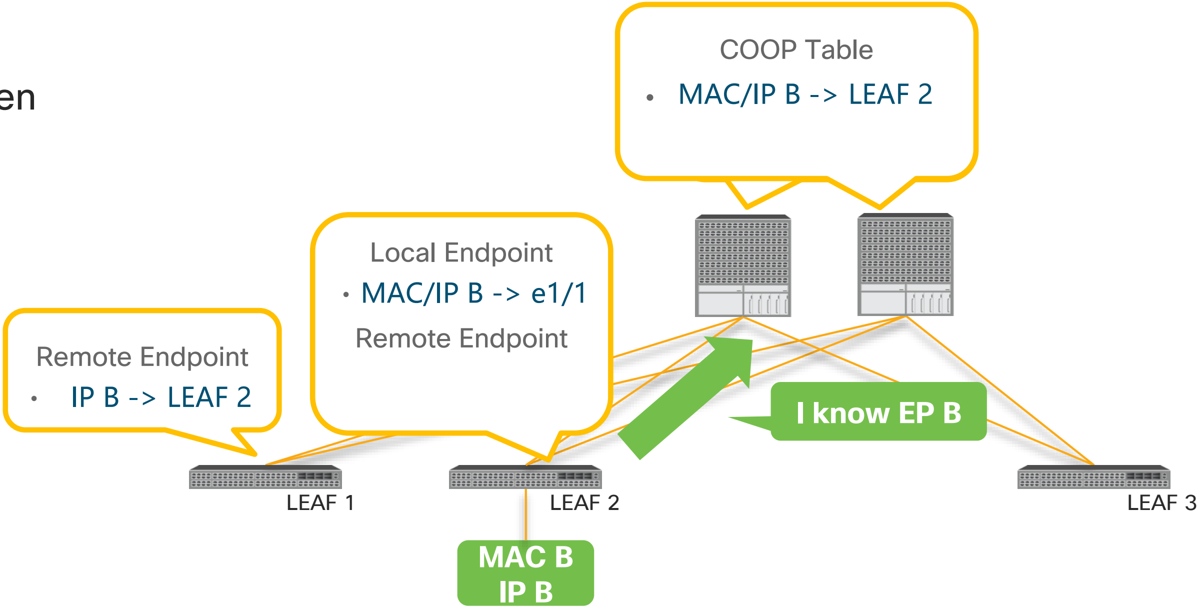
Bounce Entry

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)



For more details : ACI Fabric Endpoint Learning Whitepaper

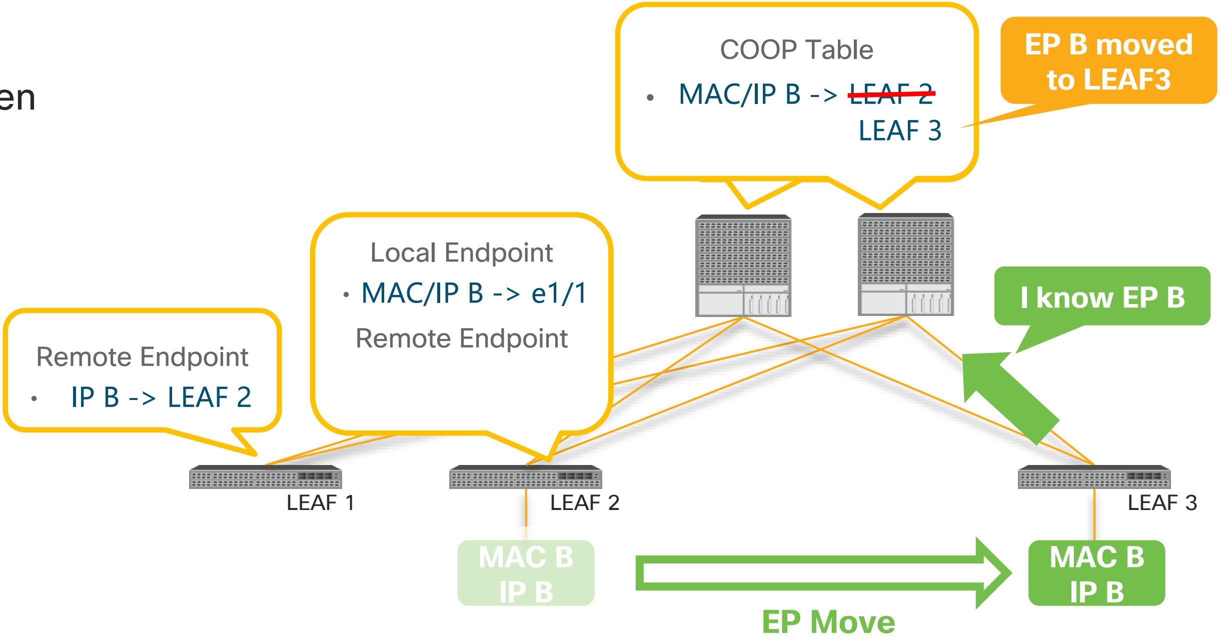
Bounce Entry

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)



For more details : ACI Fabric Endpoint Learning Whitepaper

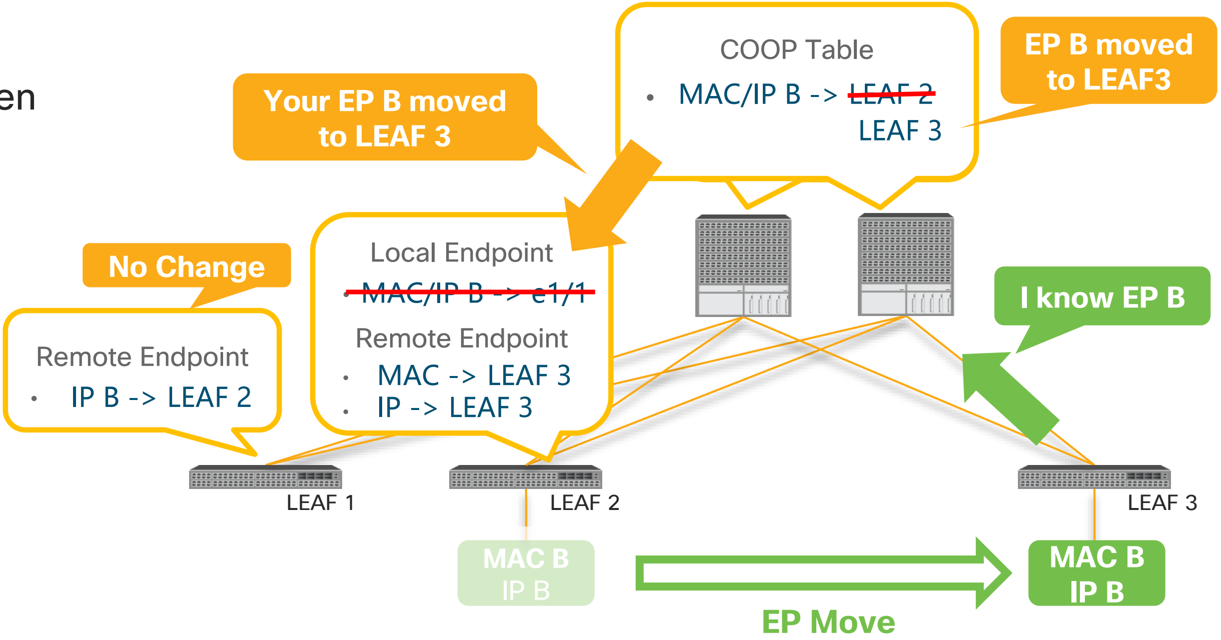
Bounce Entry

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)



For more details : ACI Fabric Endpoint Learning Whitepaper

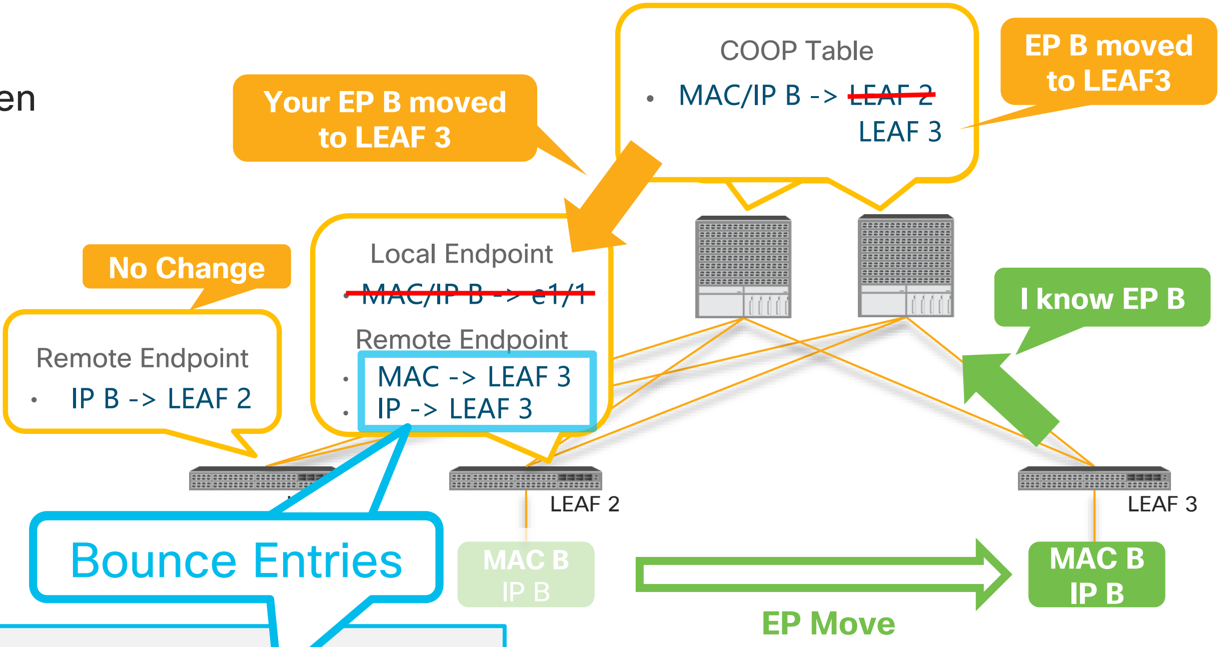
Bounce Entry

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)



COOP Table

- MAC/IP B -> ~~LEAF 2~~ LEAF 3

EP B moved to LEAF3

Your EP B moved to LEAF 3

No Change

Remote Endpoint

- IP B -> LEAF 2

Local Endpoint

- ~~MAC/IP B -> e1/1~~

Remote Endpoint

- MAC -> LEAF 3
- IP -> LEAF 3

I know EP B

Bounce Entries

LEAF 2

LEAF 3

MAC B IP B

EP Move

MAC B IP B

```
leaf2# show end vrf TK:VRF1

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

For more details : ACI Fabric Endpoint Learning Whitepaper

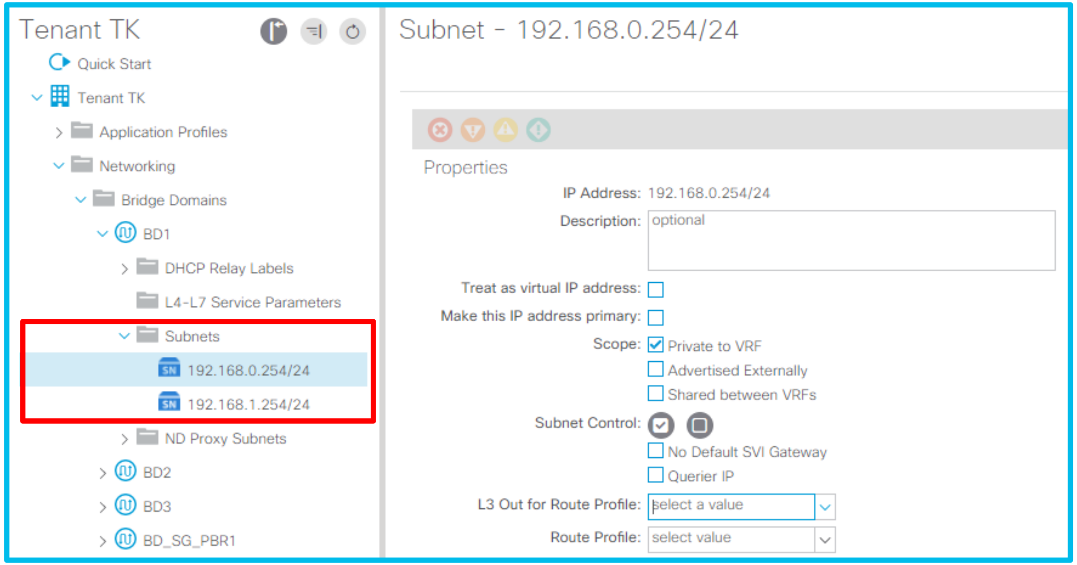
Agenda

- 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
  - Packet Capture in 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 value

```
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

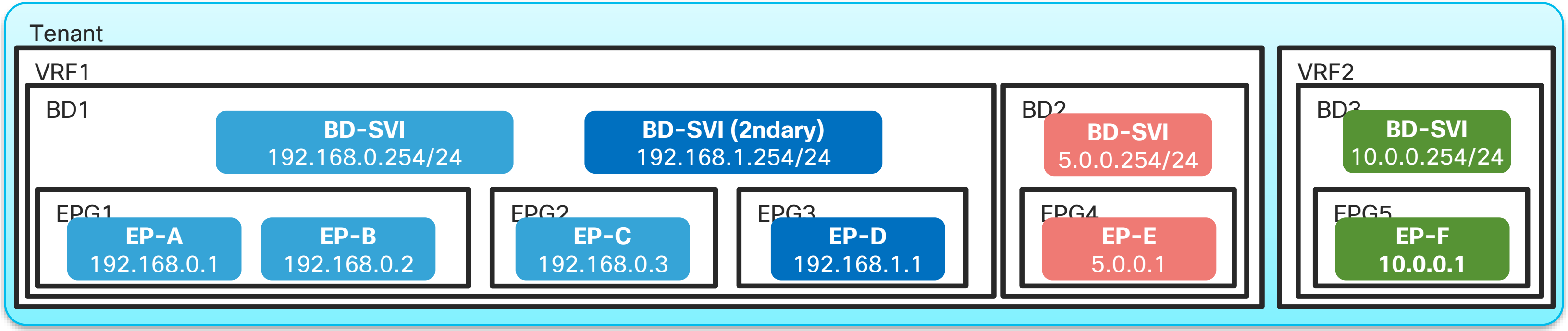
What is pervasive GW for?

- 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

Pervasive Gateway(BD SVI) example



Tenant

VRF1

BD1

BD-SVI
192.168.0.254/24

BD-SVI (2ndary)
192.168.1.254/24

EPG1

EP-A
192.168.0.1

EP-B
192.168.0.2

EPG2

EP-C
192.168.0.3

EPG3

EP-D
192.168.1.1

BD2

BD-SVI
5.0.0.254/24

EPG4

EP-E
5.0.0.1

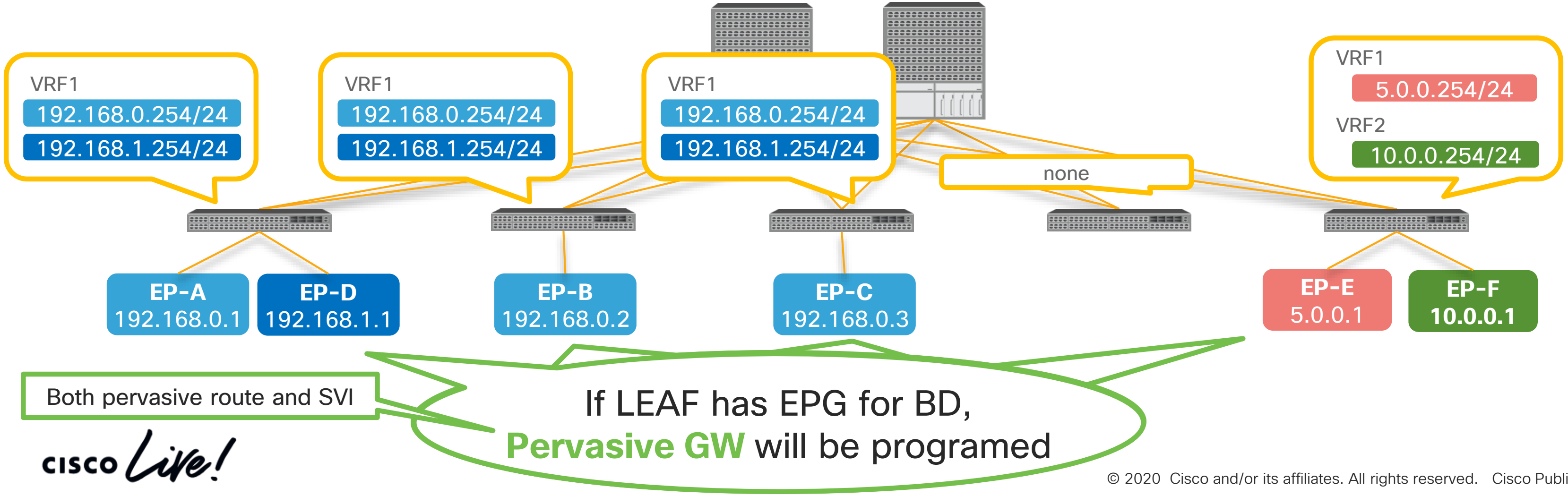
VRF2

BD3

BD-SVI
10.0.0.254/24

EPG5

EP-F
10.0.0.1



VRF1
192.168.0.254/24
192.168.1.254/24

VRF1
192.168.0.254/24
192.168.1.254/24

VRF1
192.168.0.254/24
192.168.1.254/24

none

VRF1
5.0.0.254/24
VRF2
10.0.0.254/24

EP-A
192.168.0.1

EP-D
192.168.1.1

EP-B
192.168.0.2

EP-C
192.168.0.3

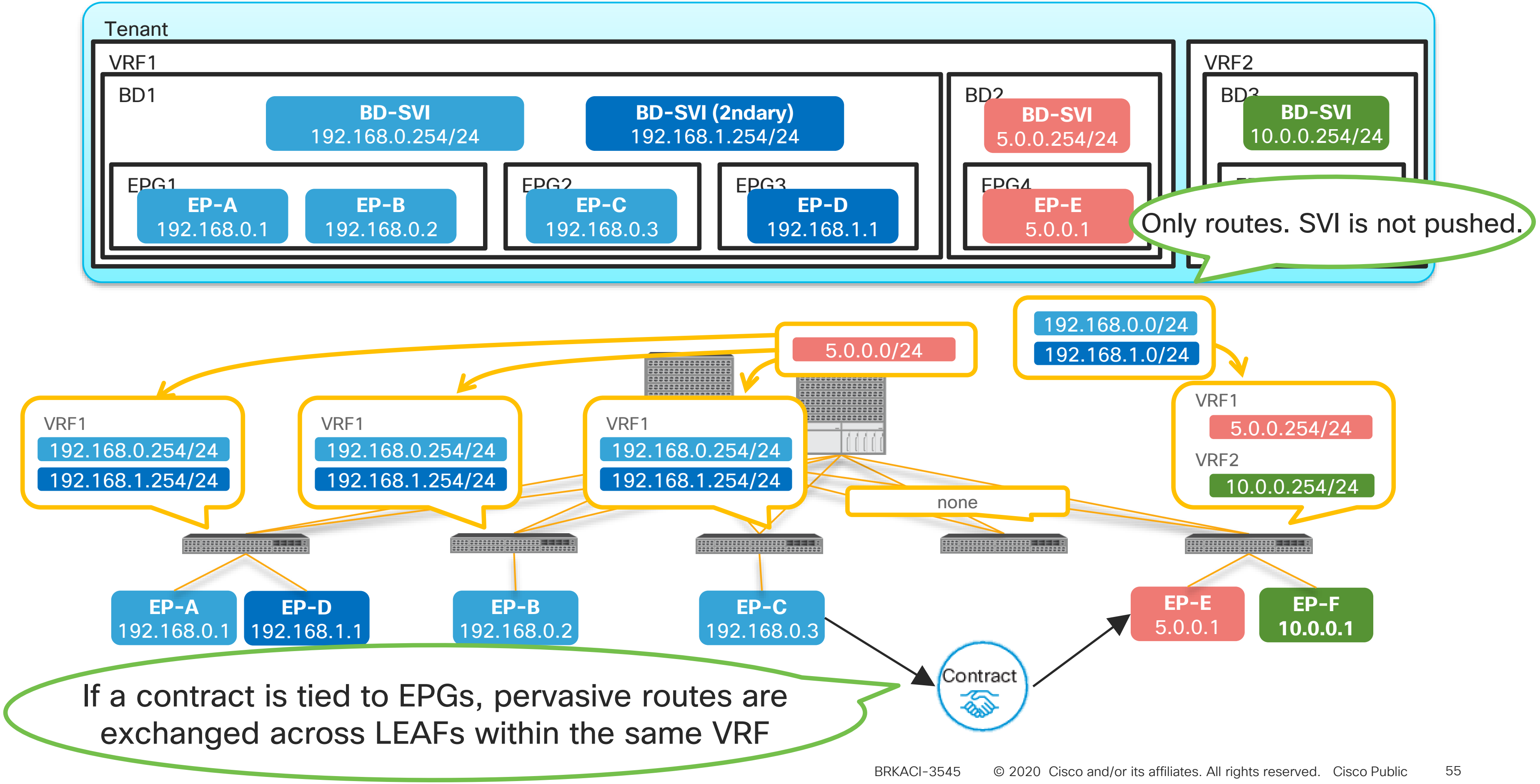
EP-E
5.0.0.1

EP-F
10.0.0.1

Both pervasive route and SVI

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

Pervasive Gateway(BD SVI) example



Tenant

VRF1

BD1

BD-SVI 192.168.0.254/24

BD-SVI (2ndary) 192.168.1.254/24

EPG1: EP-A 192.168.0.1, EP-B 192.168.0.2

EPG2: EP-C 192.168.0.3

EPG3: EP-D 192.168.1.1

BD2

BD-SVI 5.0.0.254/24

EPG4: EP-E 5.0.0.1

VRF2

BD3

BD-SVI 10.0.0.254/24

Only routes. SVI is not pushed.

5.0.0.0/24

192.168.0.0/24
192.168.1.0/24

VRF1
192.168.0.254/24
192.168.1.254/24

VRF1
192.168.0.254/24
192.168.1.254/24

VRF1
192.168.0.254/24
192.168.1.254/24

none

VRF1
5.0.0.254/24
VRF2
10.0.0.254/24

EP-A 192.168.0.1

EP-D 192.168.1.1

EP-B 192.168.0.2

EP-C 192.168.0.3

EP-E 5.0.0.1

EP-F 10.0.0.1

Contract

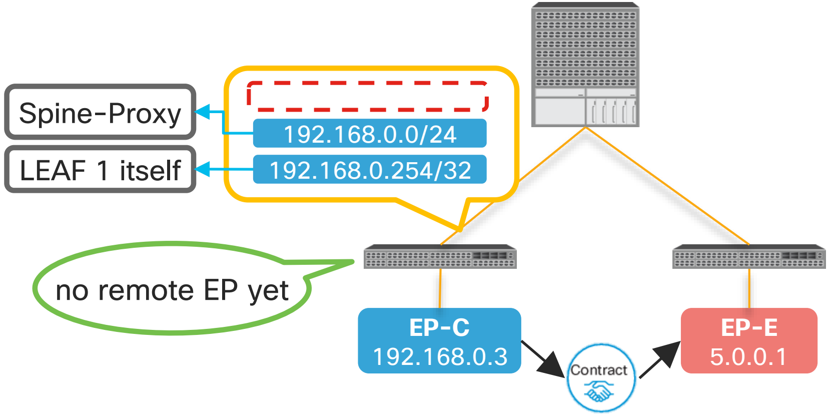
If a contract is tied to EPGs, pervasive routes are exchanged across LEAFs within the same VRF

Pervasive Gateway(BD SVI) cont.

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

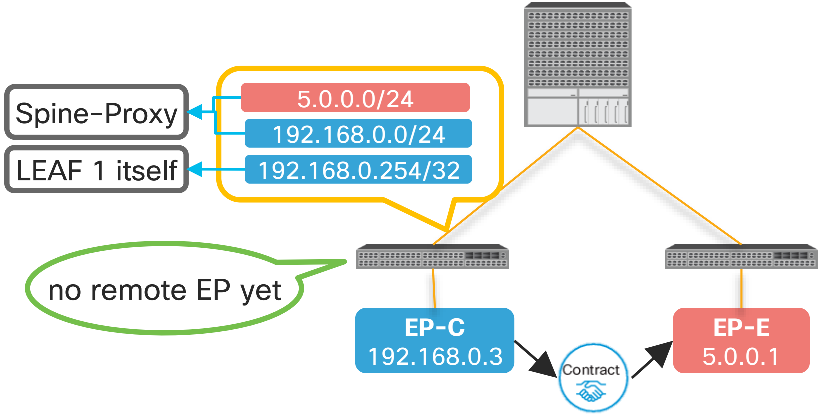
- **Pervasive routes are required for Spine-Proxy**

✗ ***what if no pervasive route, no remote EP?***



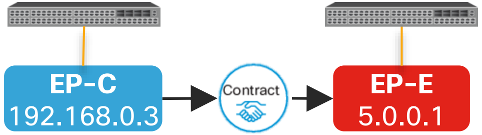
No Spine-Proxy for 5.0.0.1
It may be either dropped or forwarded to L3OUT if a default route exists

✓ ***with pervasive route and no remote EP?***



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
```

Exchange pervasive route

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
```

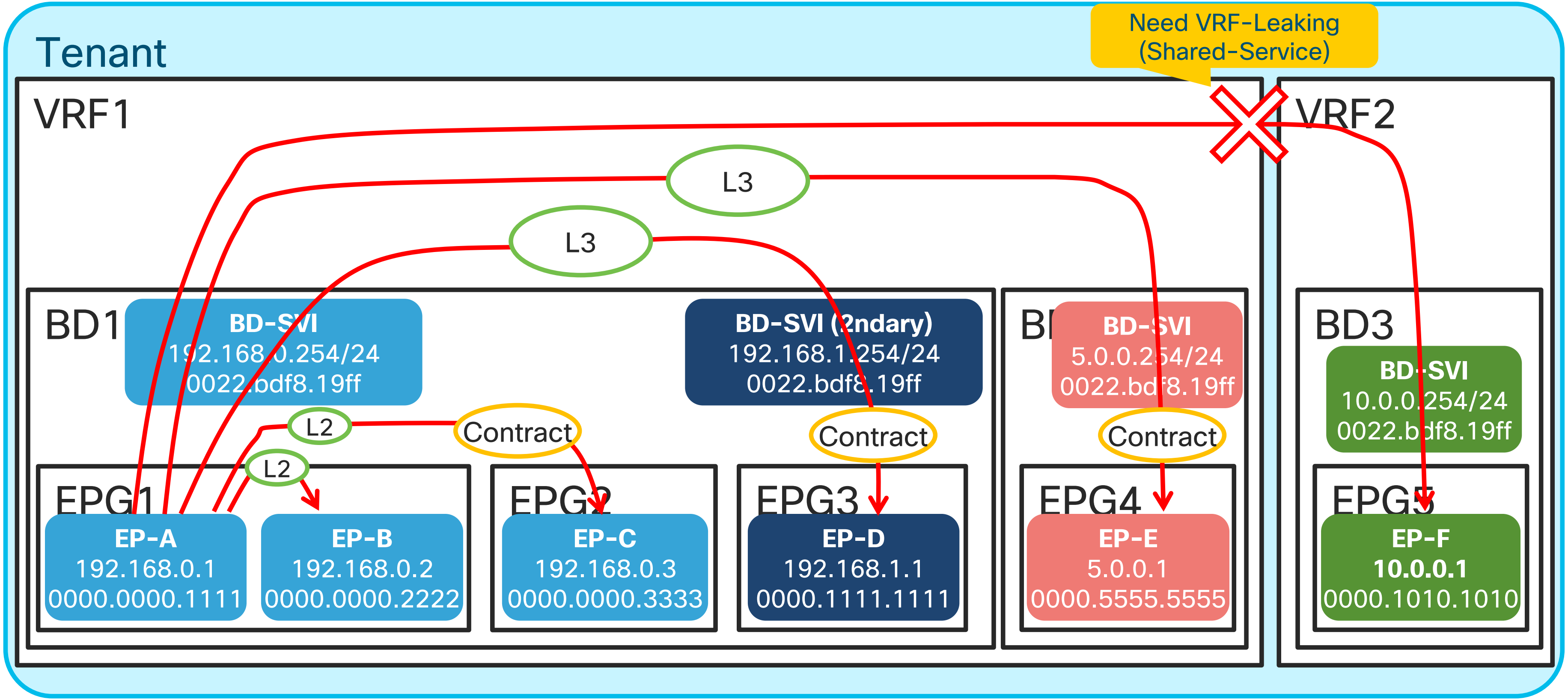
```
L103# show ip route vrf TK:VRF1

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
```

- **Pervasive routes are pushed to other LEAFs with contracts**

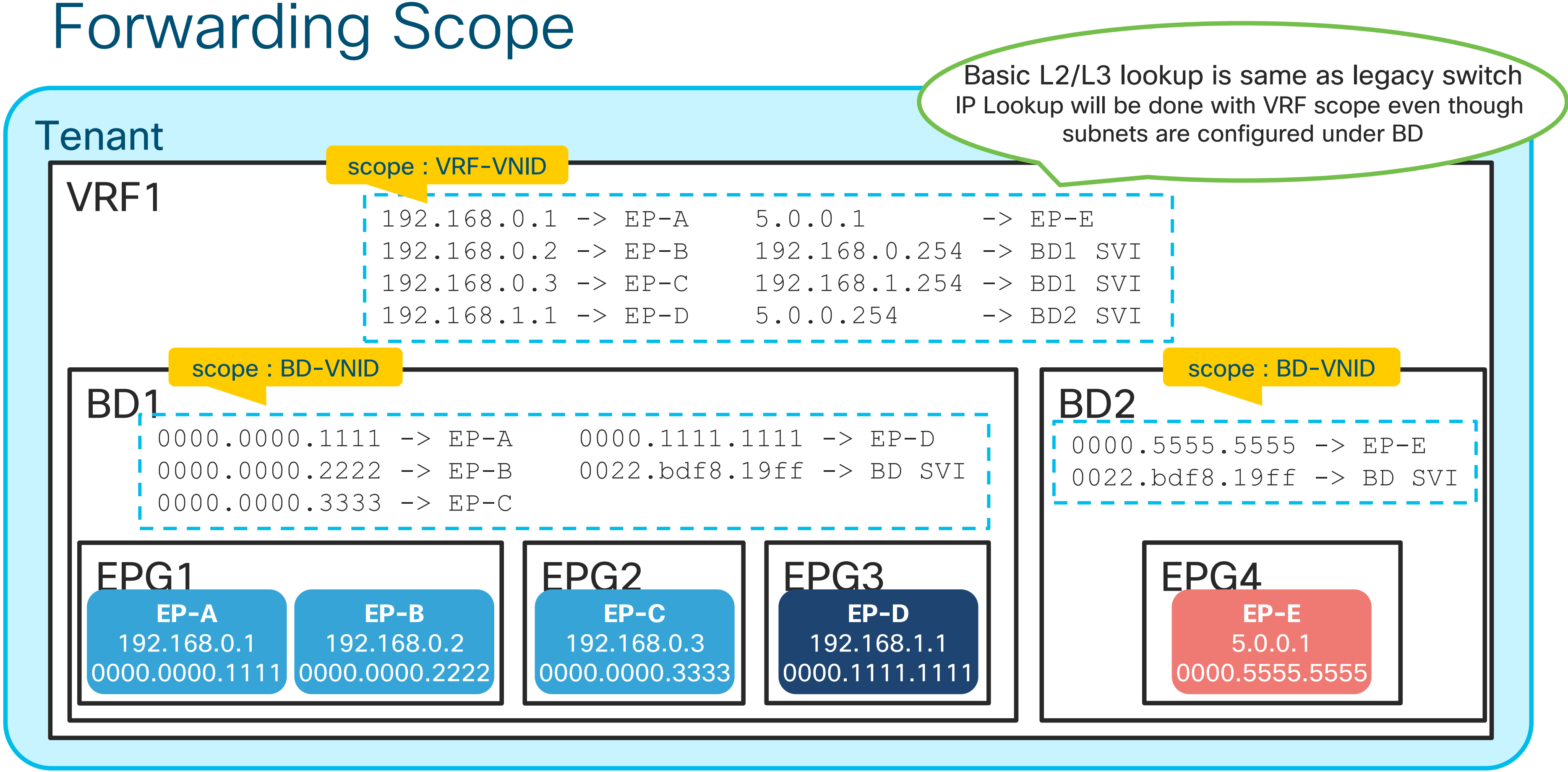
ACI Forwarding Component 2

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

ACI Forwarding Scope Concepts



Forwarding Scope



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

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

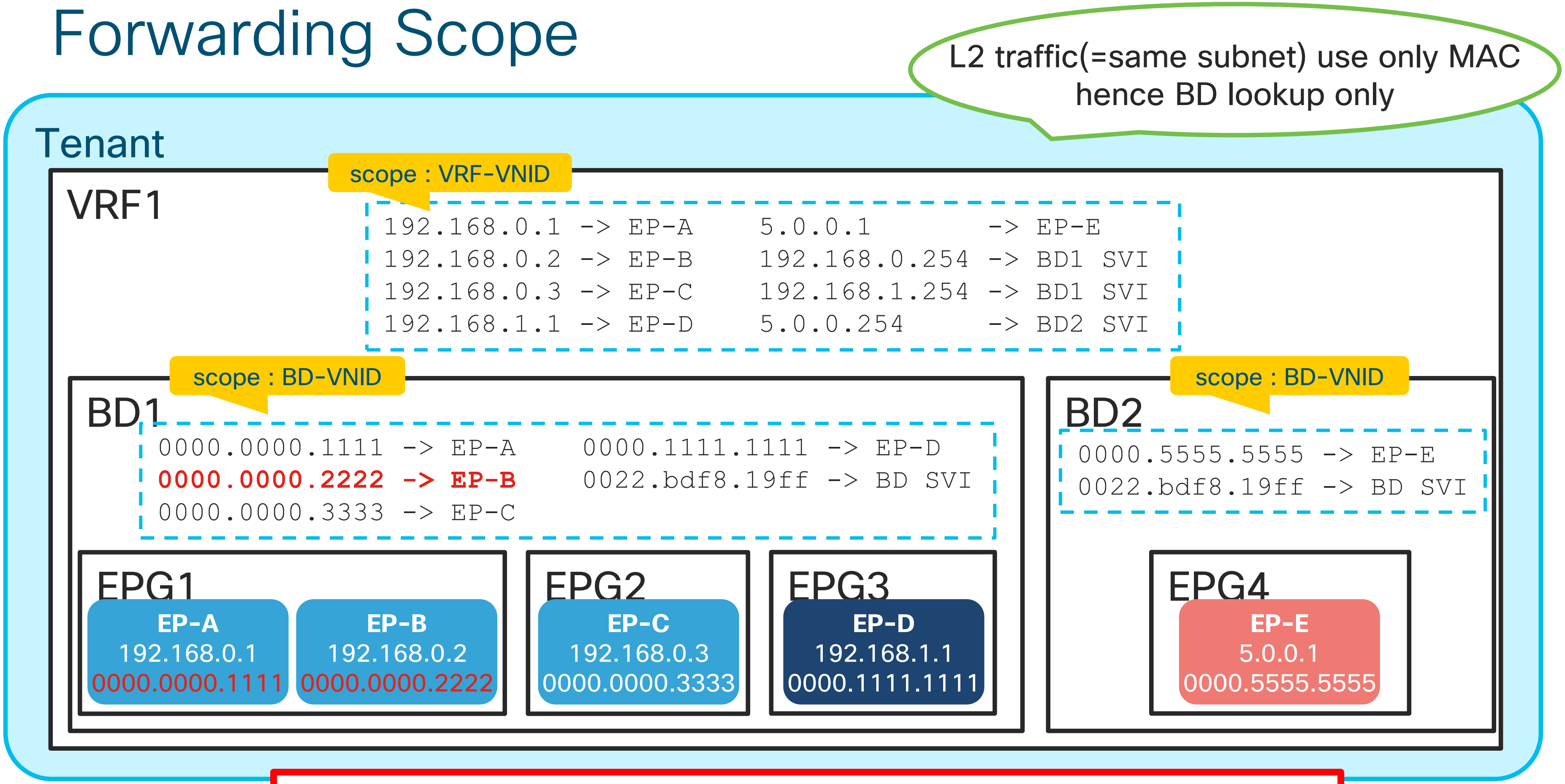
EPG3

EP-D
192.168.1.1
0000.1111.1111

EPG4

EP-E
5.0.0.1
0000.5555.5555

Forwarding Scope



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
```

BD2

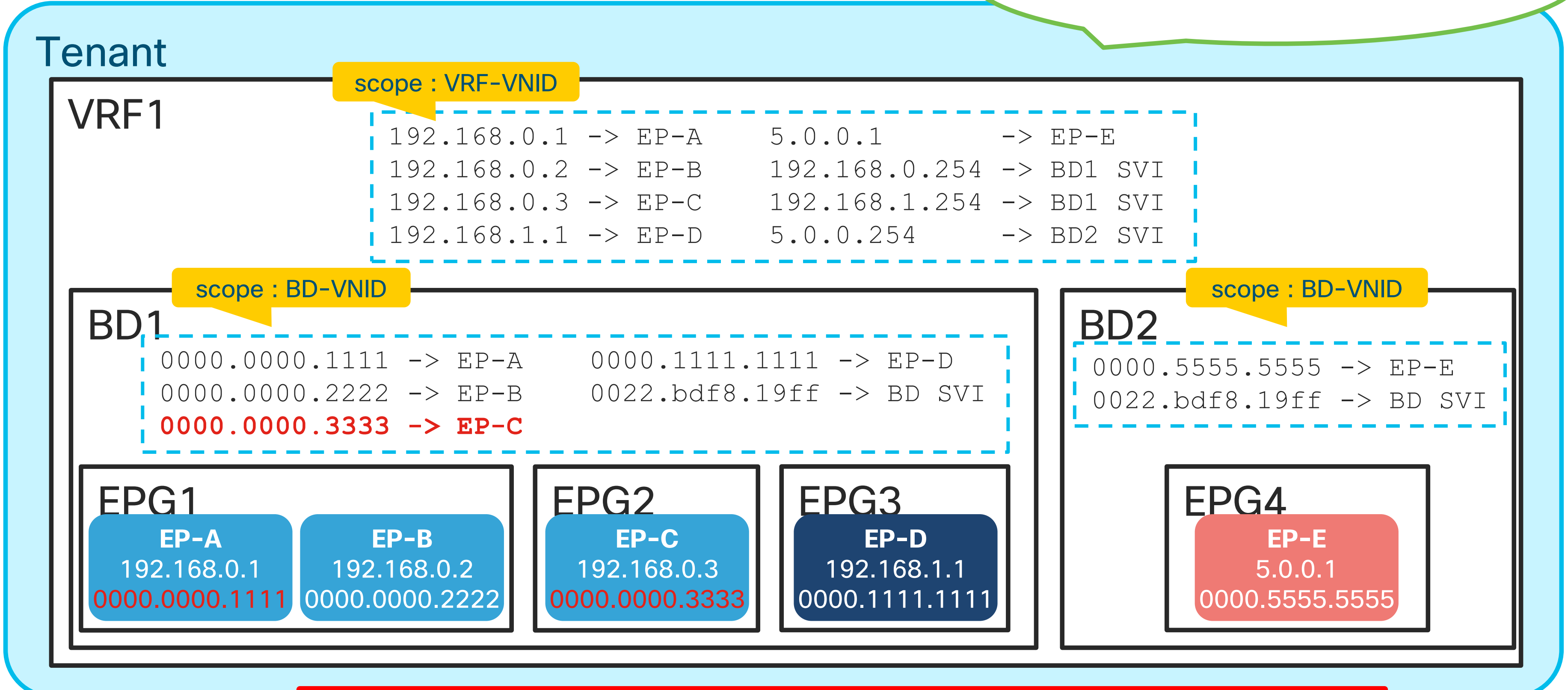
scope : BD-VNID

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

EP-A (0000.0000.1111) -> EP-B (0000.0000.2222)

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

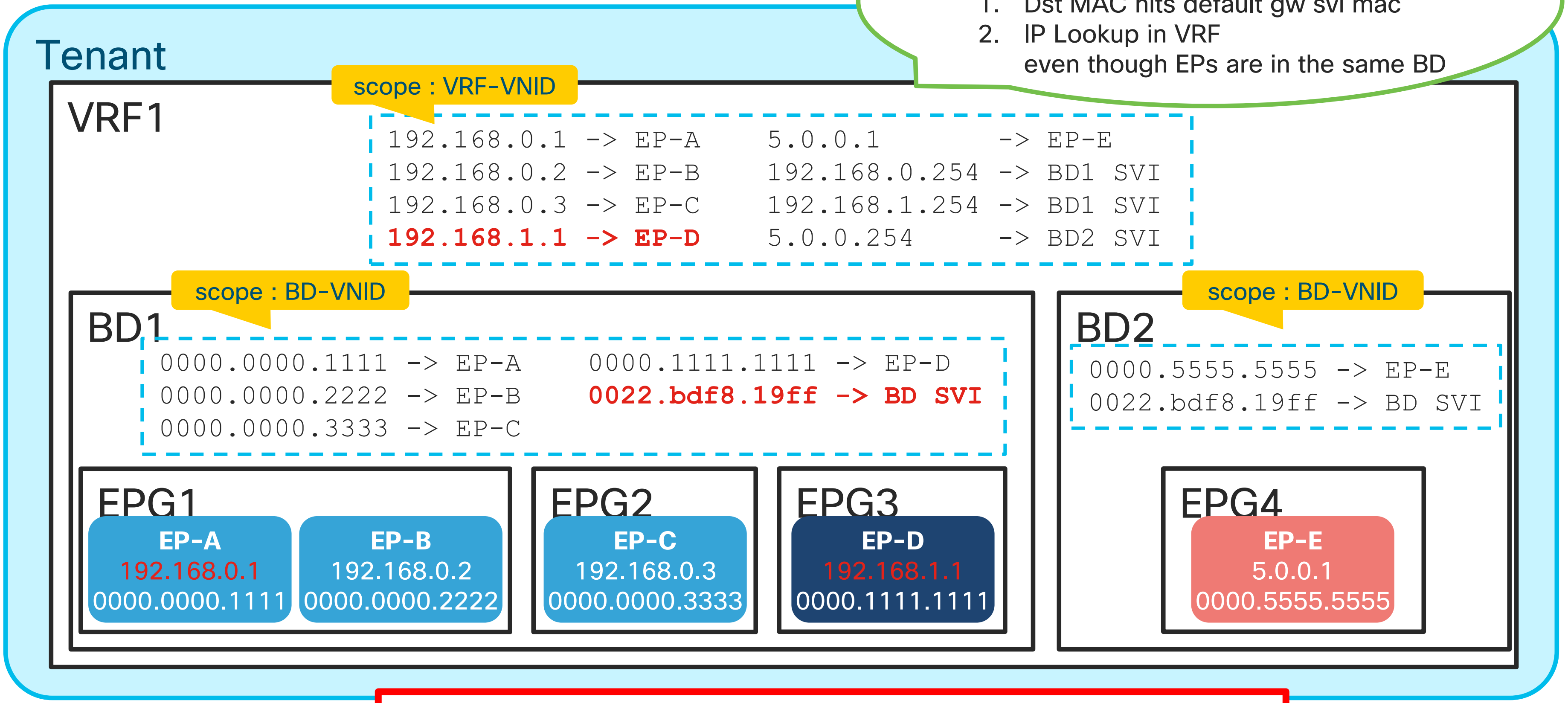
```
0000.5555.5555 -> EP-E
0022.bdf8.19ff -> BD SVI
```

EP-A (0000.0000.1111) -> EP-C (0000.0000.3333)

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
```

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

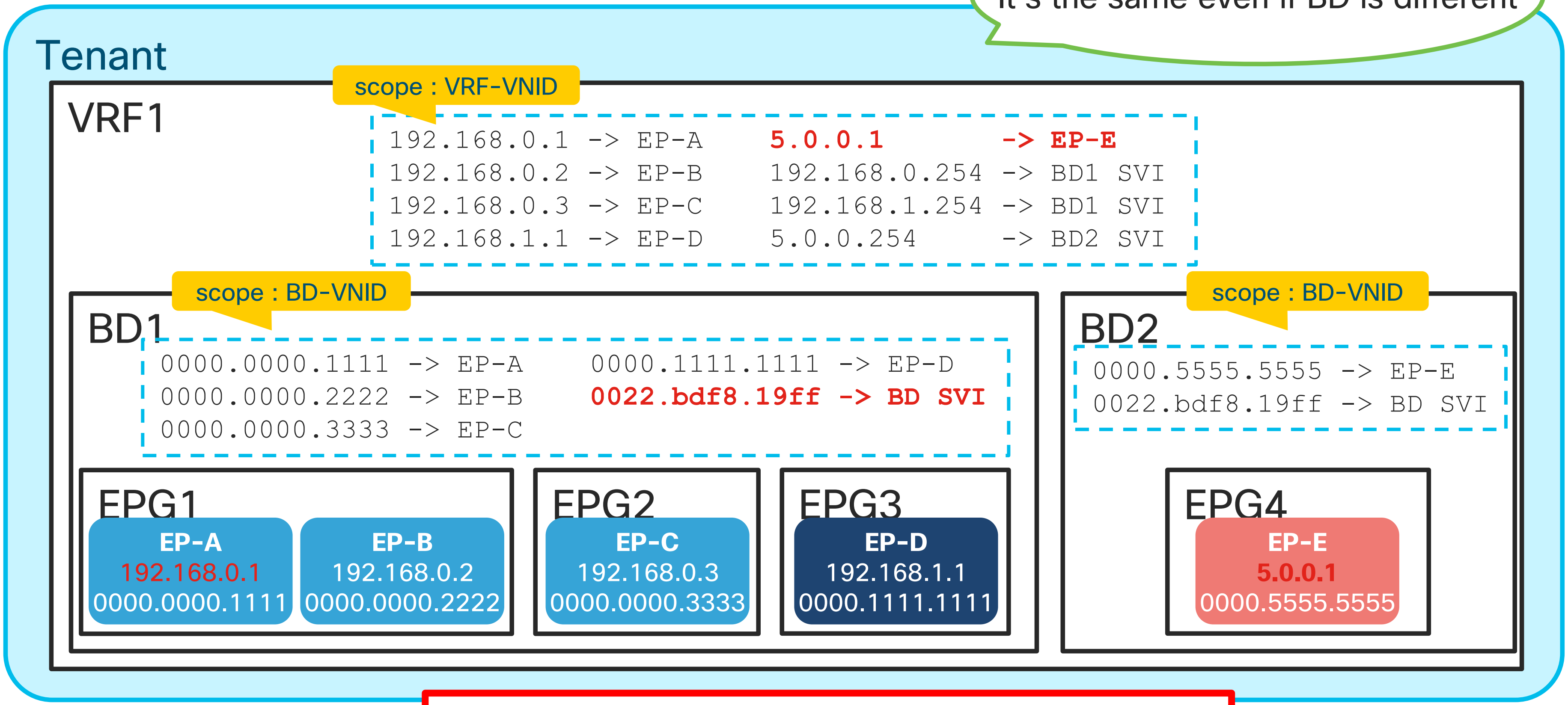
scope : BD-VNID

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

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

Forwarding Scope

It’s the same even if BD is different



```
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
```

```
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
```

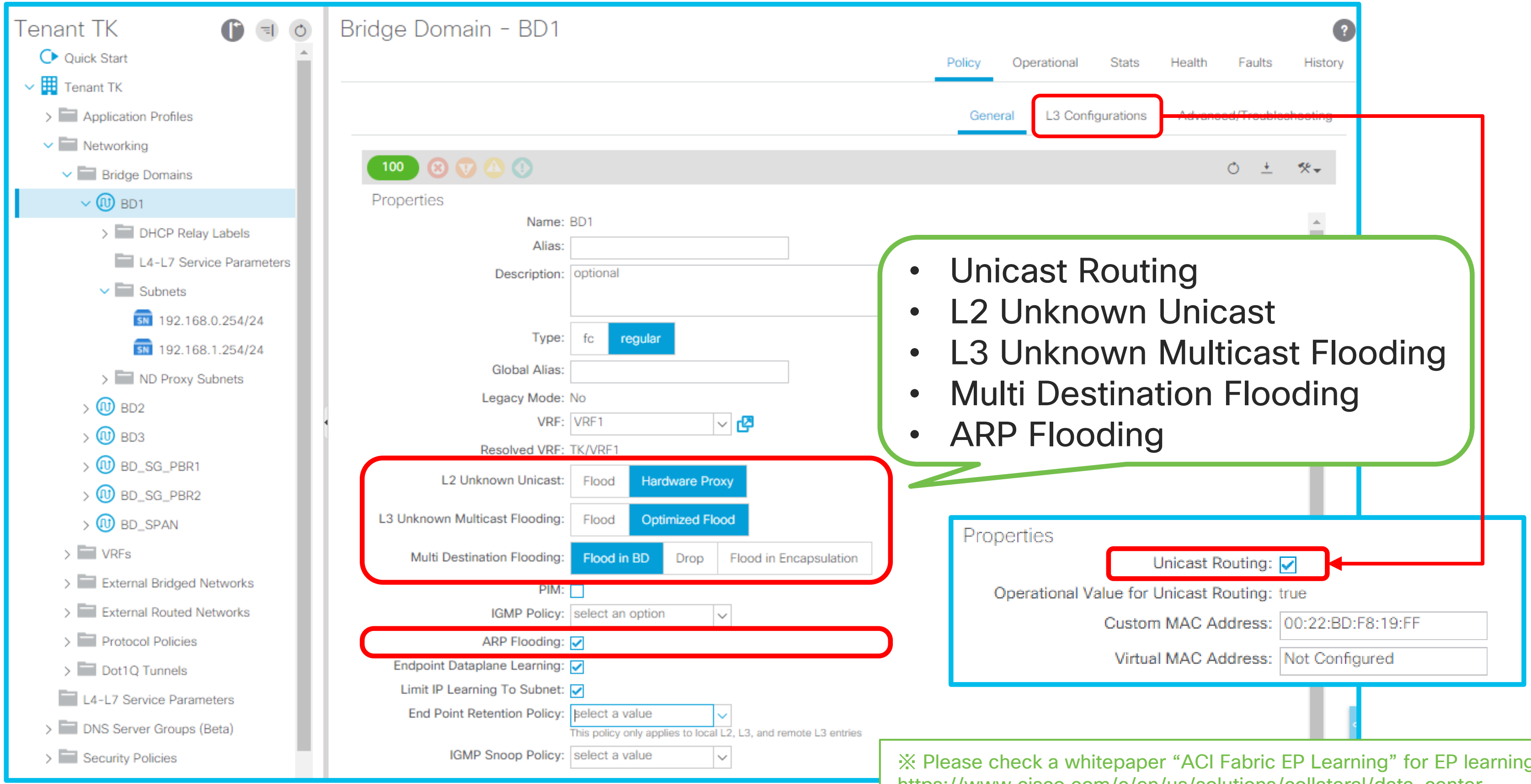
```
0000.5555.5555 -> EP-E
0022.bdf8.19ff -> BD SVI
```

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



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

This policy only applies to local L2, L3, and remote L3 entries

IGMP Snoop Policy: select a value

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

Properties

Unicast Routing: ☑

Operational Value for Unicast Routing: true

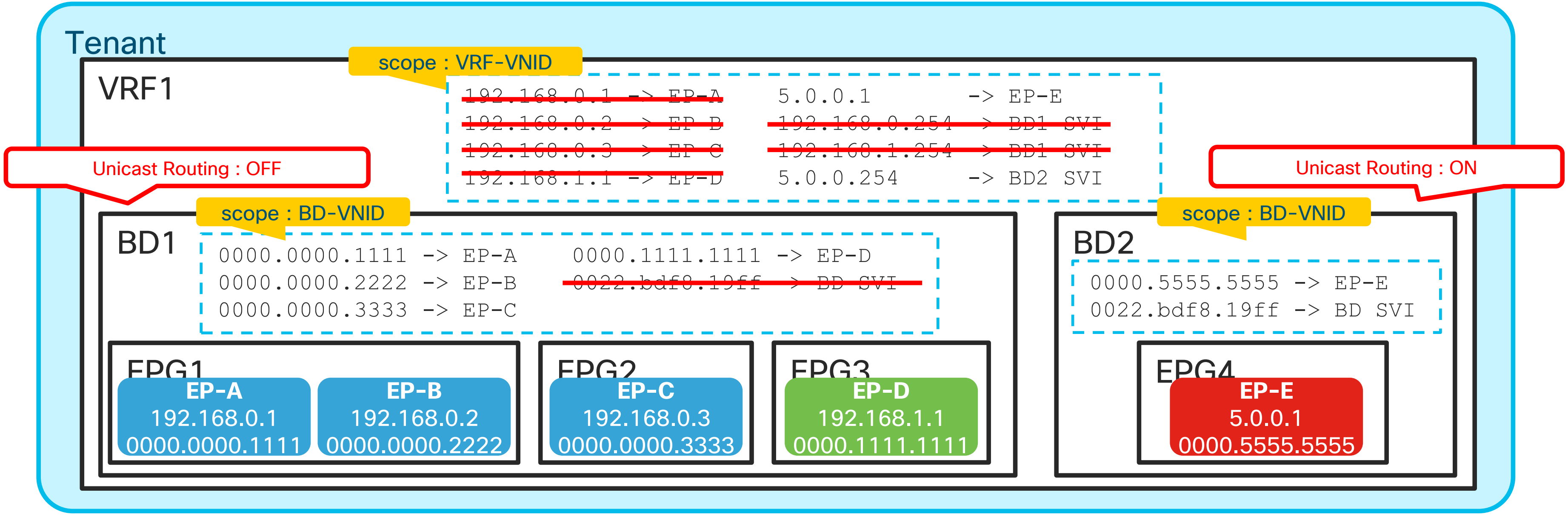
Custom MAC Address: 00:22:BD:F8:19:FF

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

Unicast Routing : OFF

BD1

scope : BD-VNID

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

Unicast Routing : ON

BD2

scope : BD-VNID

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

If Unicast Routing is disabled, (BD1 in above)

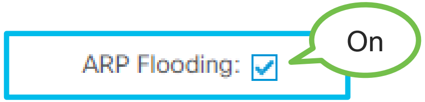
- IP Learning is disabled on BD
- BD SVI is disabled

=> Only L2 Forwarding is available

In above example :

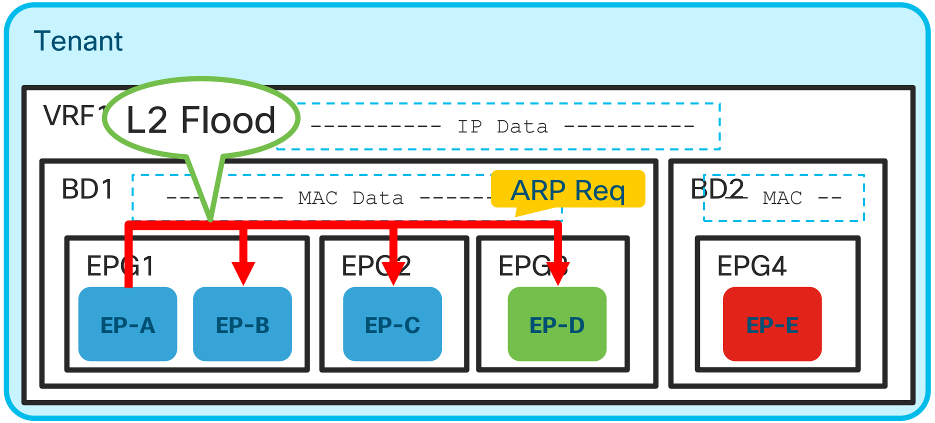
- EP-A <-> EP-B : GOOD (L2 forwarding)
- EP-A <-> EP-C : GOOD (L2 forwarding)
- EP-A <-> EP-D : FAIL (L3 forwarding)
- EP-A <-> EP-E : FAIL (L3 forwarding)

ARP Flooding



ARP Flooding: ☑

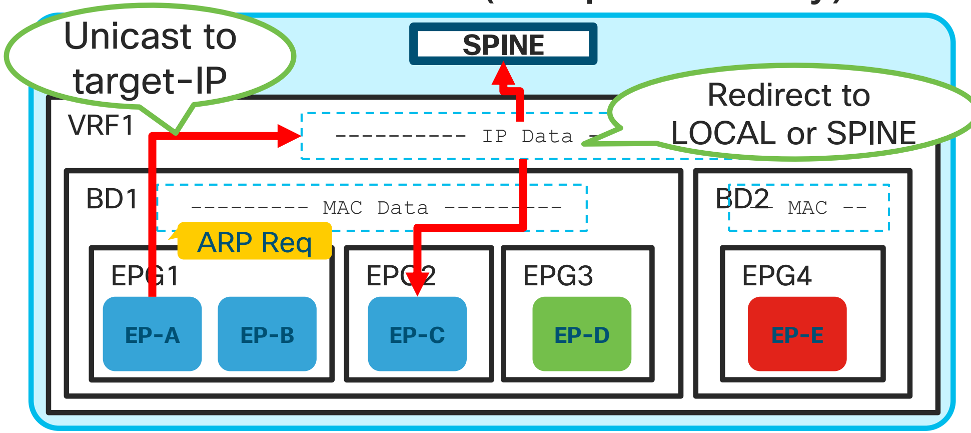
- ARP Flood On



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)

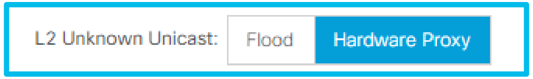


ARP Request is handled as L3 Unicast with Target-IP

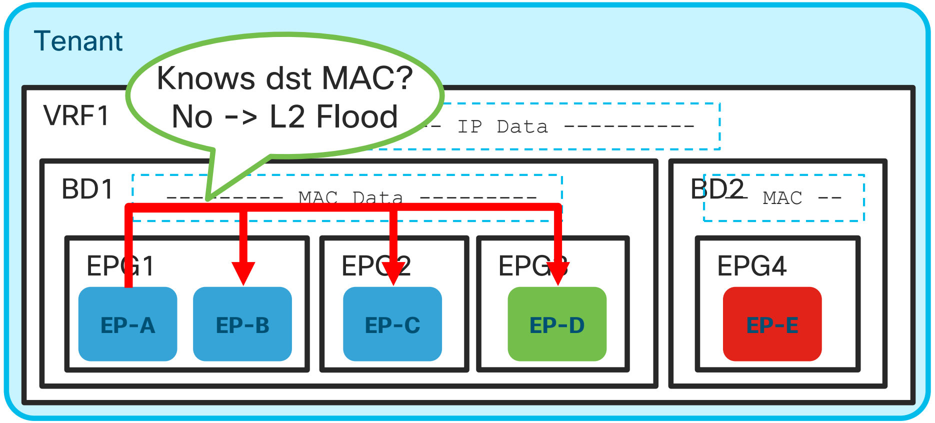
- If IP is learned on ingress Leaf,
  - ➢ Ingress Leaf forwards ARP Req directly to dest
- If IP is not learned on ingress Leaf,
  - ➢ Ingress Leaf forwards ARP Req to Spine (Spine-Proxy)
    Spine will forward it to Leaf on which DstIP resides
- If IP is not learned even on Spine,
  - ➢ Drop and ARP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

※ ARP is not filtered by a contract by default
※ if dst mac is not bcast, ARP request is bridged based on dst mac regardless of ARP Flooding mode

L2 Unknown Unicast


L2 Unknown Unicast: Flood [✓ Hardware Proxy]

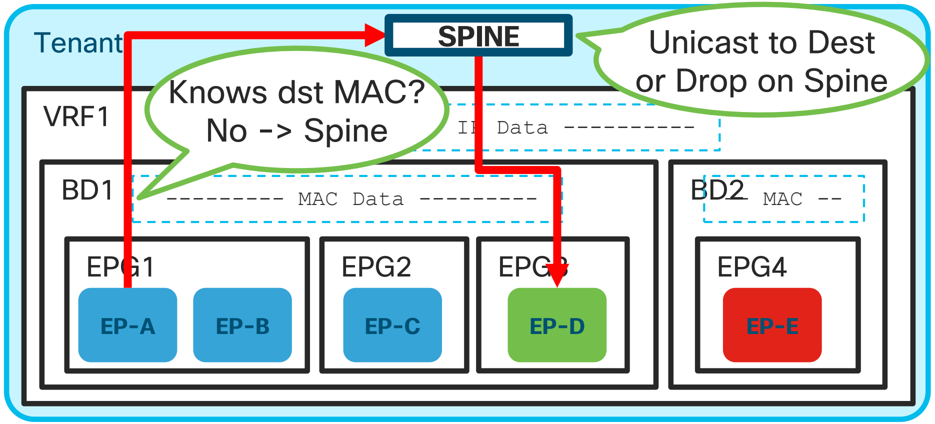
- Flood



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

- Hardware Proxy (= Spine-Proxy)

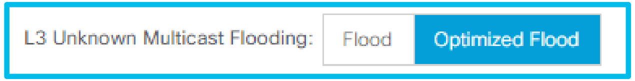


L2 Unknown Unicast is sent to Spine-Proxy

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

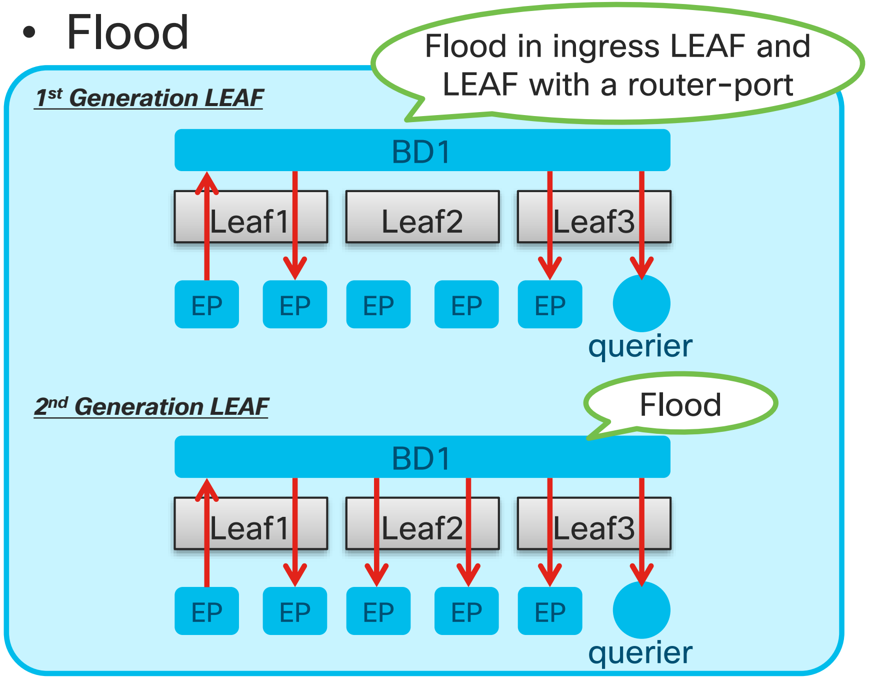
※EP IP Data is not used even if Leaf knows DST-IP. L2 Unknown is still L2 traffic.

L3 Unknown Multicast Flooding

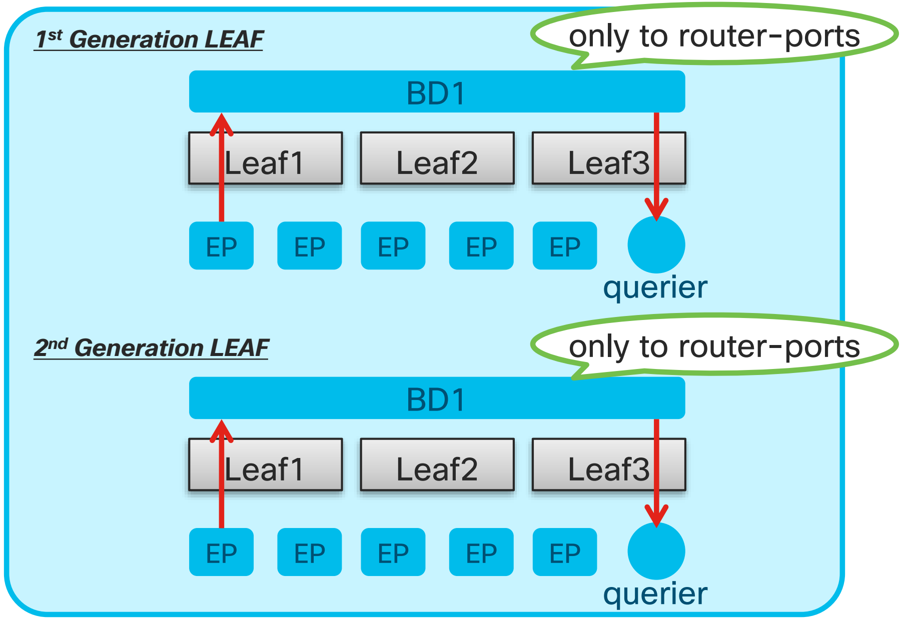


L3 Unknown Multicast Flooding: Flood [✓ Optimized Flood]

- Flood



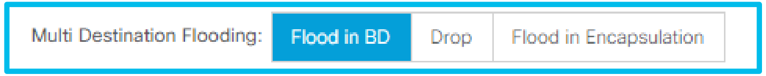
- OMF (Optimized Multicast Flood)



L3 Unknown Multicast = IP multicast group unknown to LEAF IGMP snooping

➢ Controls flooding ***unknown*** IGMP snooping groups

Multi Destination Flooding

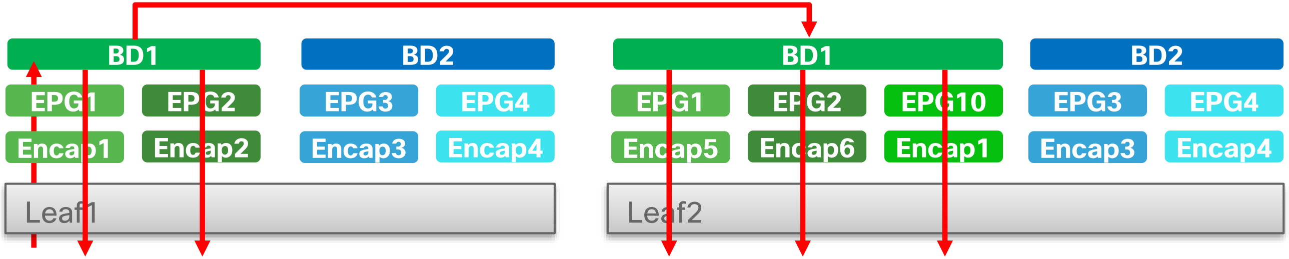


Multi Destination Flooding: Flood in BD | Drop | Flood in Encapsulation

Flooding mode for L2 multicast, Broadcast and link-local

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

- **Flood in BD**
  Flood within the same BD regardless of EPG or VLAN.

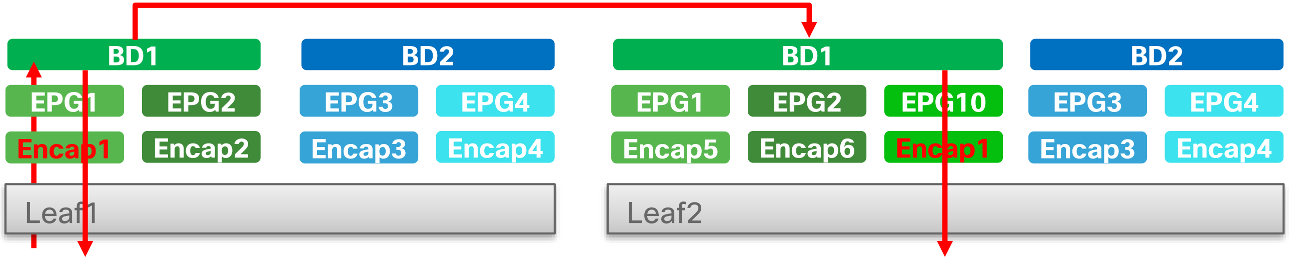


Leaf1: BD1 (EPG1 Encap1, EPG2 Encap2), BD2 (EPG3 Encap3, EPG4 Encap4)

Leaf2: BD1 (EPG1 Encap5, EPG2 Encap6, EPG10 Encap1), BD2 (EPG3 Encap3, EPG4 Encap4)

Behavior change from 3.1

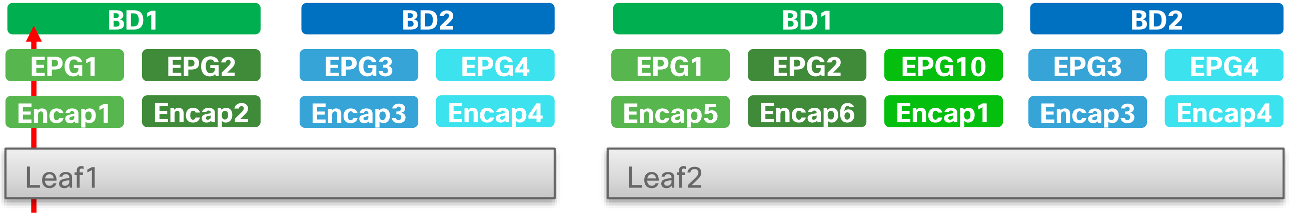
- **Flood in Encapsulation**
  Flood within the same access encap VLAN and BD regardless of EPG.



Leaf1: BD1 (EPG1 Encap1, EPG2 Encap2), BD2 (EPG3 Encap3, EPG4 Encap4)

Leaf2: BD1 (EPG1 Encap5, EPG2 Encap6, EPG10 Encap1), BD2 (EPG3 Encap3, EPG4 Encap4)

- **Drop**
  No Flood. Just drop.

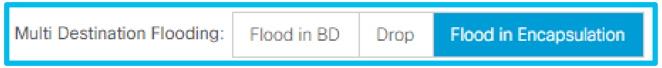


Leaf1: BD1 (EPG1 Encap1, EPG2 Encap2), BD2 (EPG3 Encap3, EPG4 Encap4)

Leaf2: BD1 (EPG1 Encap5, EPG2 Encap6, EPG10 Encap1), BD2 (EPG3 Encap3, EPG4 Encap4)

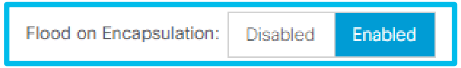
Flood in Encapsulation

BD



Multi Destination Flooding: Flood in BD | Drop | [✓ Flood in Encapsulation]

EPG

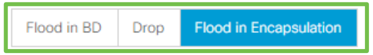


Flood on Encapsulation: Disabled | [✓ Enabled]

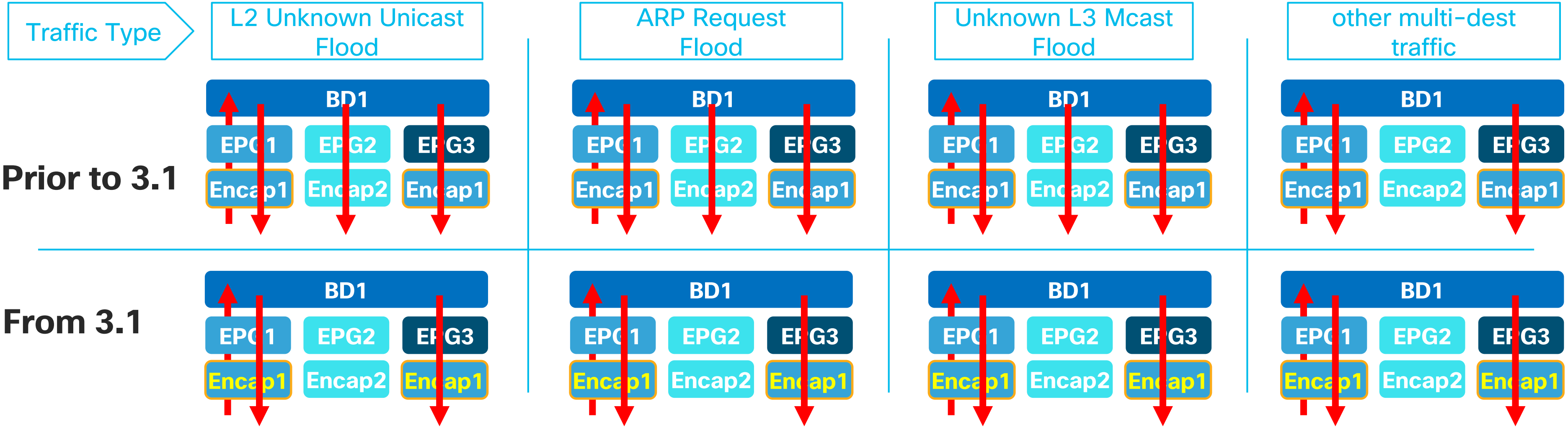
L2 Unknown Unicast: [✓ Flood] | Hardware Proxy

ARP Flooding: ☑

L3 Unknown Multicast Flooding: [✓ Flood] | Optimized Flood



Flood in BD | Drop | [✓ Flood in Encapsulation]

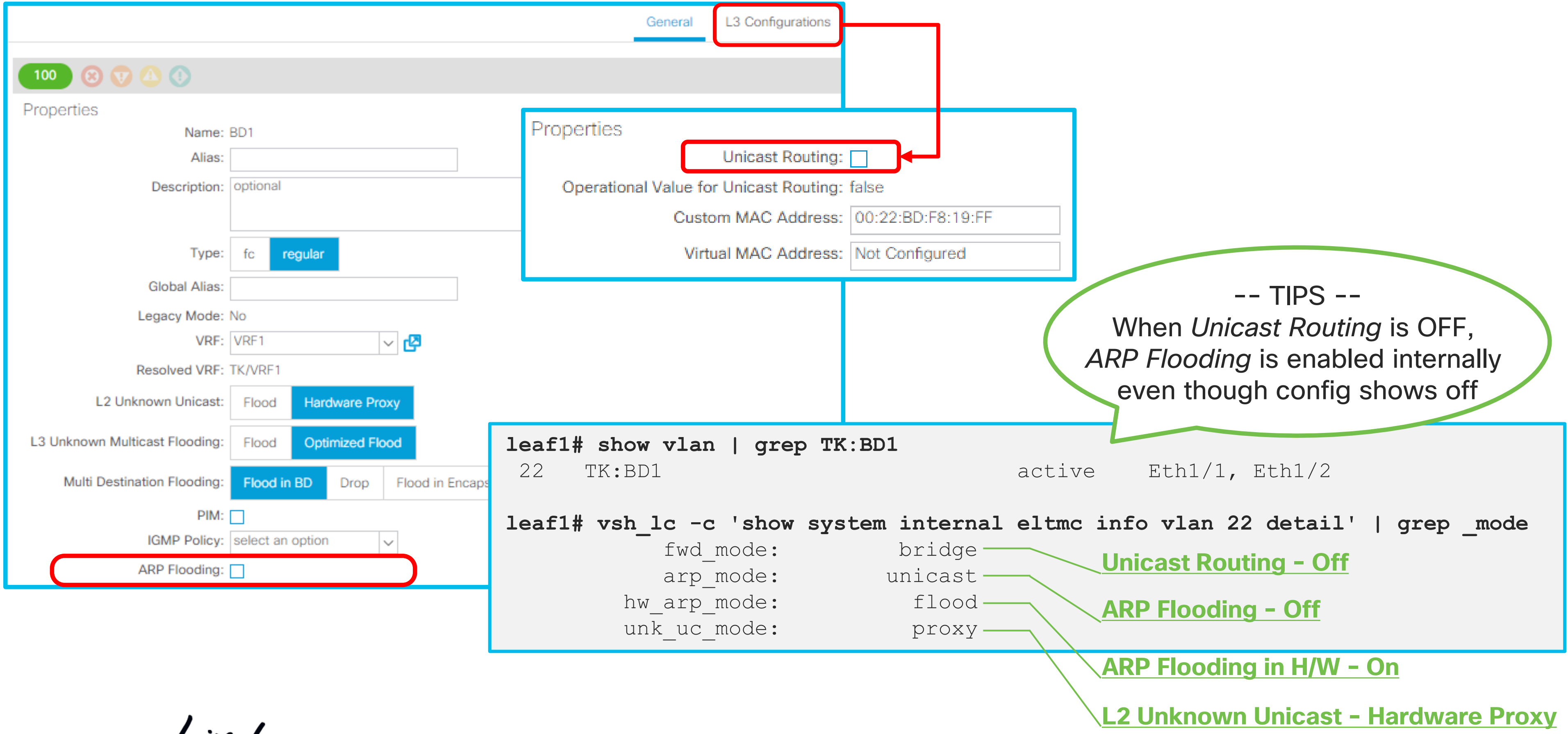


From 3.1 and 2nd generation LEAF,
all packets are flooded within encapsulation without exceptions

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.)



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 Encaps

PIM: □

IGMP Policy: select an option

ARP Flooding: □

Properties

Unicast Routing: □

Operational Value for Unicast Routing: false

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

Virtual MAC Address: Not Configured

-- 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 eltmc 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

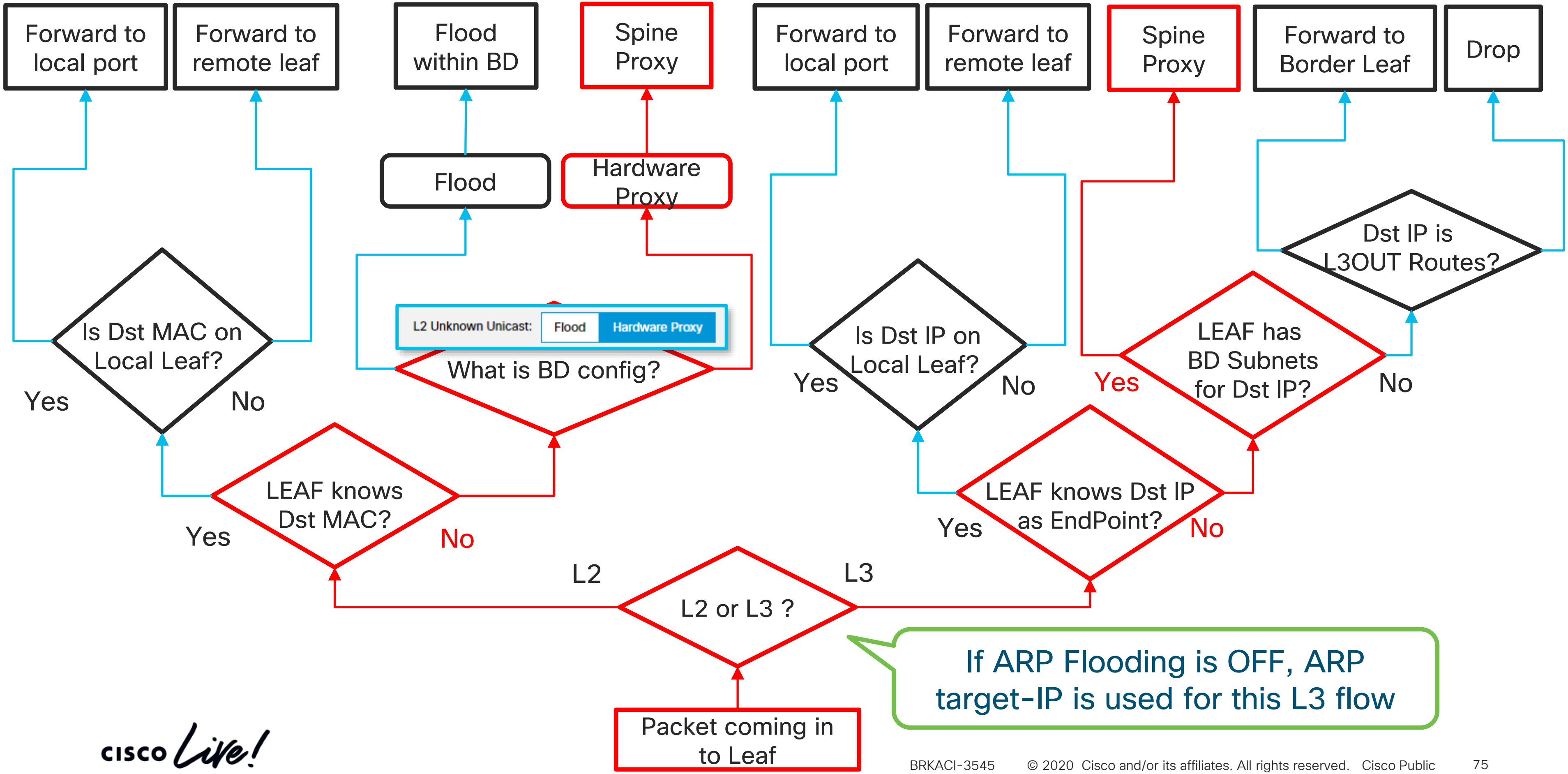
ARP Flooding in H/W - On

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
  - Packet Capture in ACI

Spine Proxy Summary



L2 Unknown Unicast: Flood Hardware Proxy

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

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 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

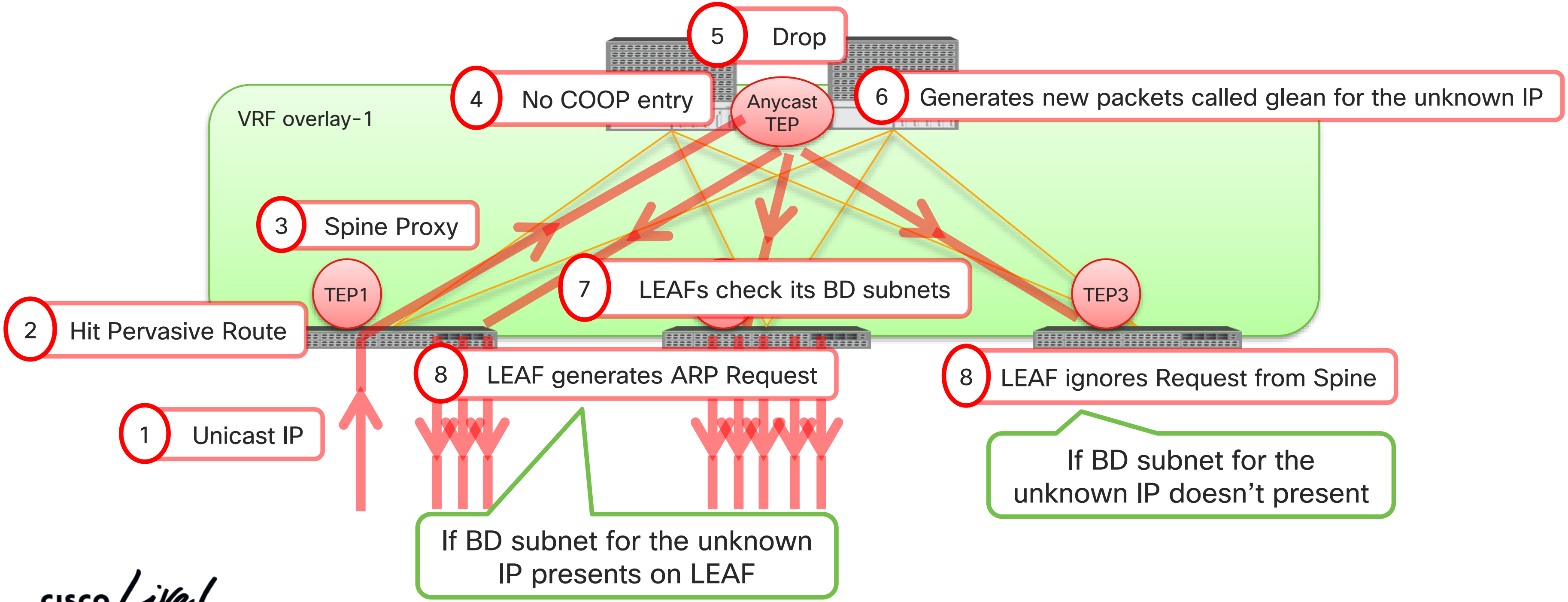
- Proxy-Acast-MAC
  - ✓ Spine-Proxy for L2 traffic (L2 Unknown Unicast mode “Hardware Proxy”)
- Proxy-Acast-V4
  - ✓ Spine-Proxy for IPv4 traffic (includes ARP Request with ARP Flooding mode “OFF”)
- Proxy-Acast-V6
  - ✓ Spine-Proxy for IPv6 traffic

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



5 Drop
4 No COOP entry
Anycast TEP
6 Generates new packets called glean for the unknown IP
VRF overlay-1
3 Spine Proxy
TEP1
7 LEAFs check its BD subnets
TEP3
2 Hit Pervasive Route
8 LEAF generates ARP Request
8 LEAF ignores Request from Spine
1 Unicast IP
If BD subnet for the unknown IP doesn't present
If BD subnet for the unknown IP presents on LEAF

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
  - Packet Capture in 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.

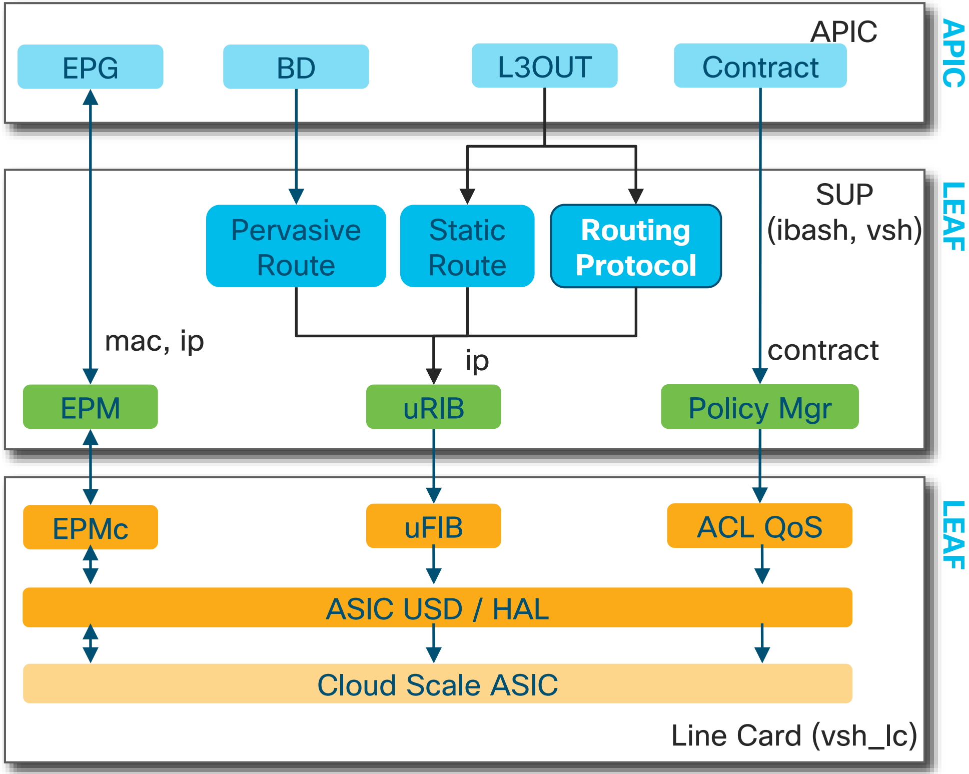
on the **Linecards**:

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 ....

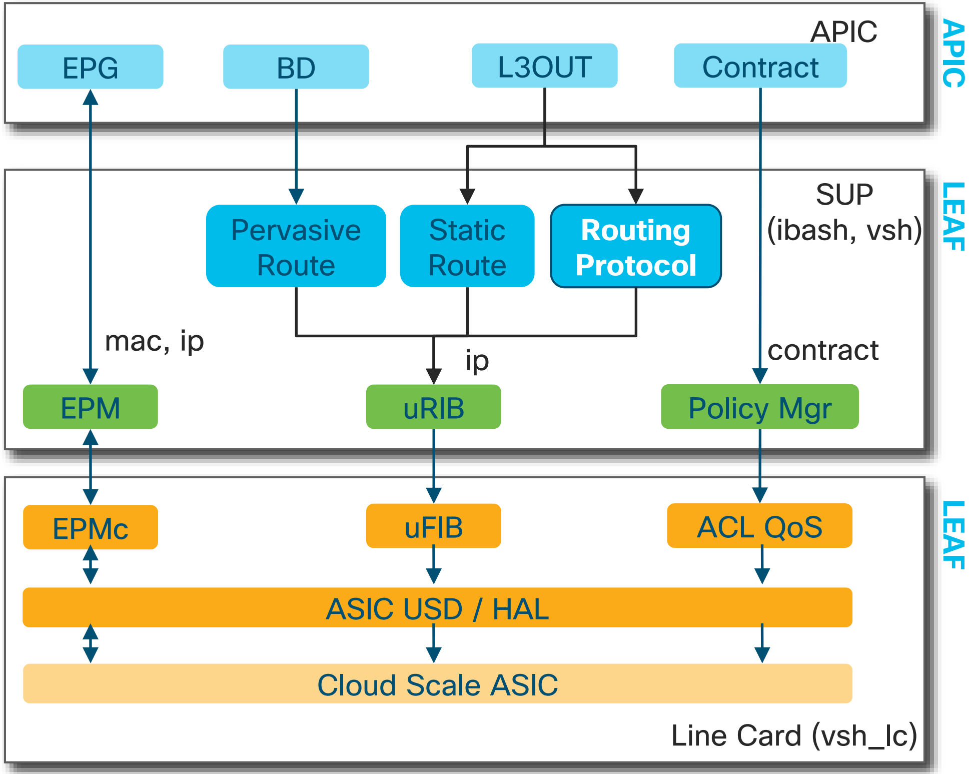
on the **Linecards**: **vsh_lc**

EPMc show system internal epmc ...

uFIB show forwarding ...

ACL QoS show system internal aclqos ...

HAL show platform internal hal ...

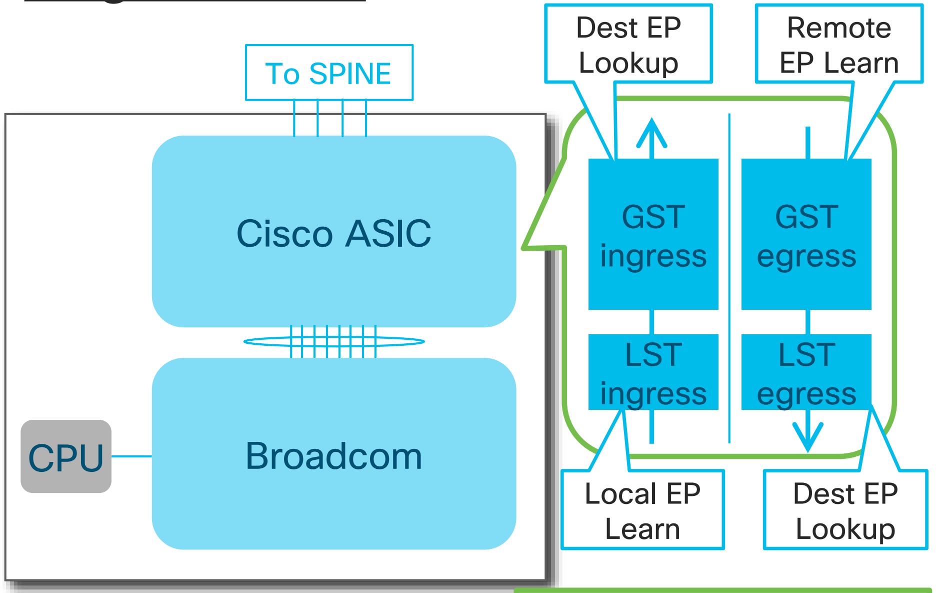


※ ASIC USD (User Space Driver) is only for 1st generation ASIC

LEAF ASIC Generations

※ LST: Local Station Table
※ GST: Global Station Table
※ FP Tile: Forwarding and Policy Tile
※ HAL: Hardware Abstraction Layer

1st generation



To SPINE

Cisco ASIC

CPU

Broadcom

Dest EP Lookup

Remote EP Learn

GST ingress

GST egress

LST ingress

LST egress

Local EP Learn

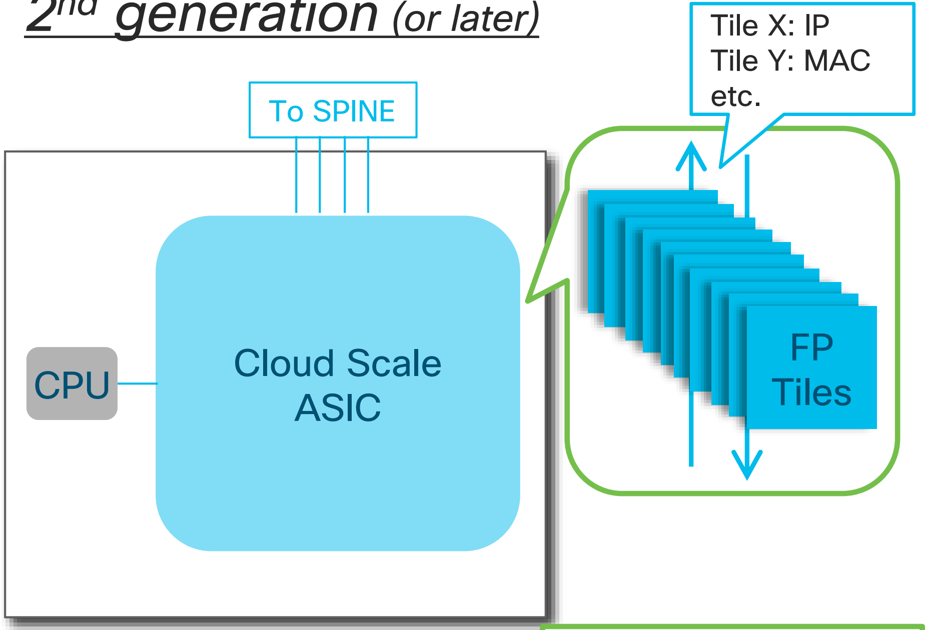
Dest EP Lookup

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)



To SPINE

Tile X: IP
Tile Y: MAC
etc.

CPU

Cloud Scale ASIC

FP Tiles

N9K-C93180YC-EX
N9K-C93108TC-EX
N9K-C93180LC-EX

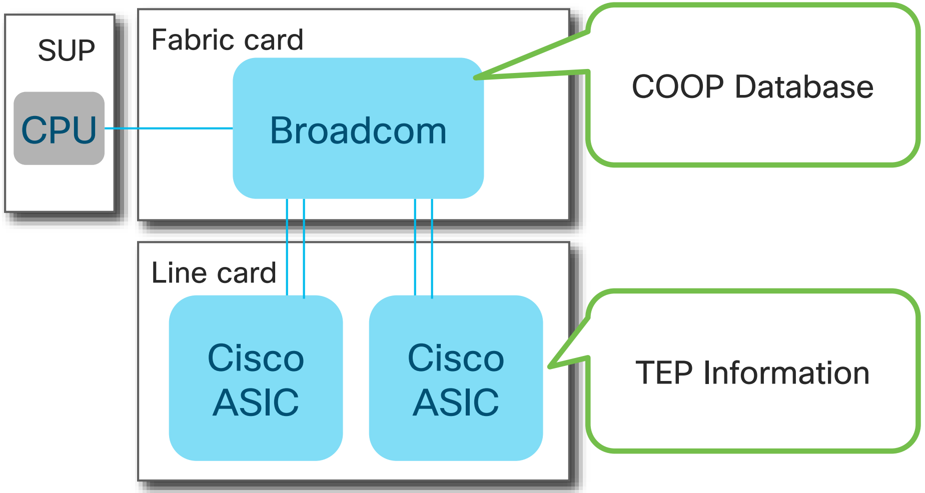
N9K-C93180YC-FX
N9K-C93108TC-FX
N9K-C9348GC-FXP
N9K-C9336C-FX2
N9K-C93240YC-FX2
....

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

※ number of ASIC per card depends on model

SPINE ASIC Generations

1st generation

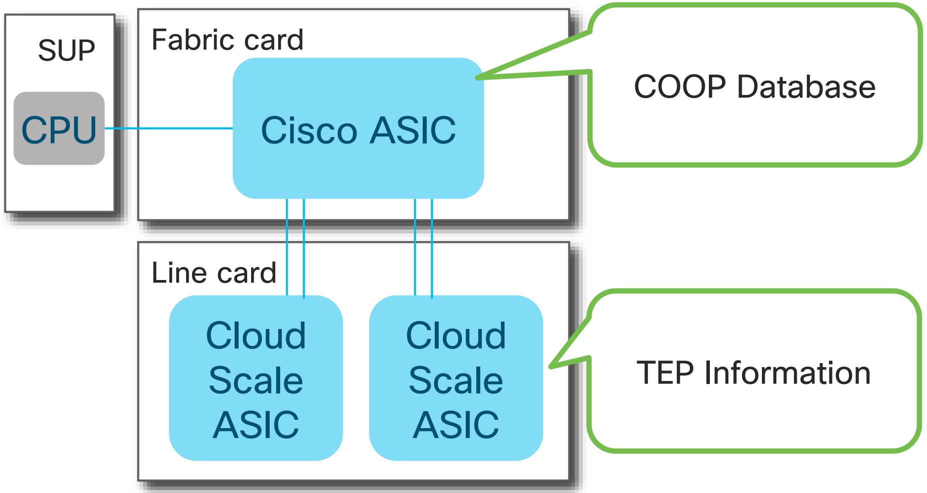

SUP
CPU
Fabric card
Broadcom
COOP Database
Line card
Cisco ASIC
Cisco ASIC
TEP Information

Line card
N9K-X9736PQ

Box spine
N9K-C9336PQ

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

2nd generation (or later)


SUP
CPU
Fabric card
Cisco ASIC
COOP Database
Line card
Cloud Scale ASIC
Cloud Scale ASIC
TEP Information

Line card
N9K-X9732C-EX
N9K-X9736C-FX

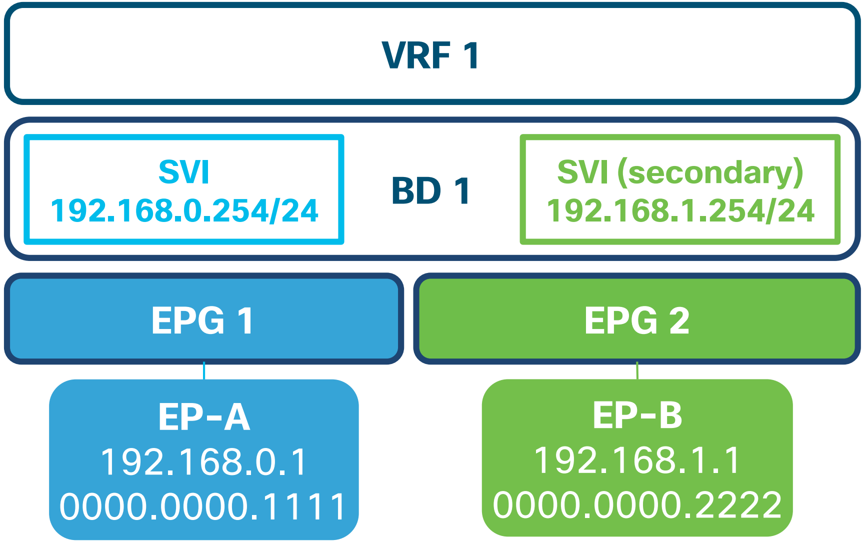
Box spine
N9K-C9364C
N9K-C9332C

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
  - Packet Capture in ACI

- Logical Topology



VRF 1

SVI
192.168.0.254/24

BD 1

SVI (secondary)
192.168.1.254/24

EPG 1

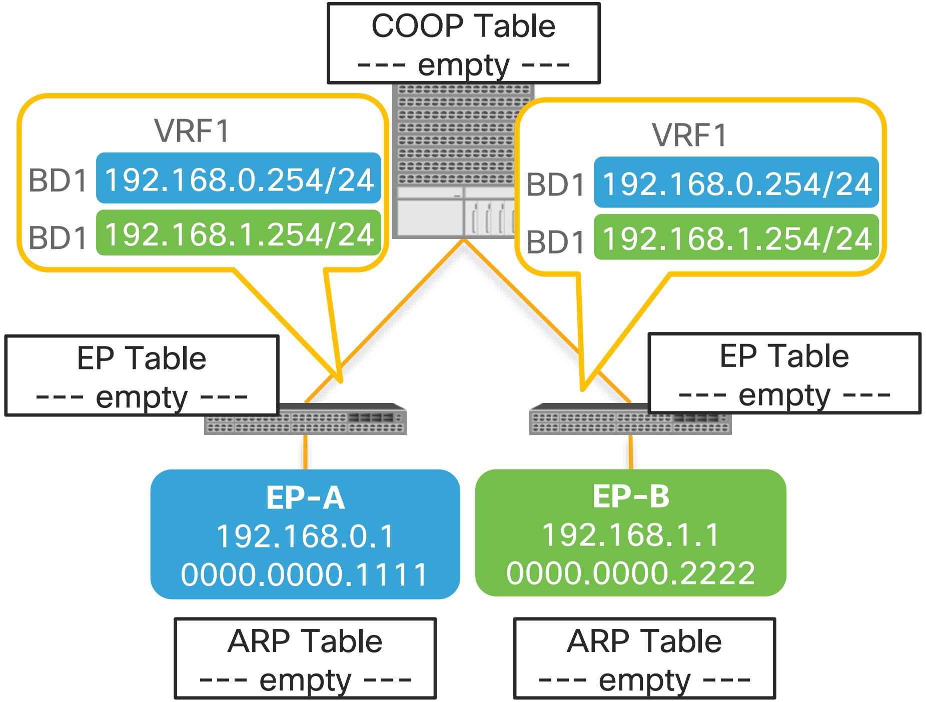
EPG 2

EP-A
192.168.0.1
0000.0000.1111

EP-B
192.168.1.1
0000.0000.2222

ARP Flooding : OFF
Unicast Routing : ON

- Physical Topology



COOP Table
--- empty ---

VRF1
BD1 192.168.0.254/24
BD1 192.168.1.254/24

VRF1
BD1 192.168.0.254/24
BD1 192.168.1.254/24

EP Table
--- empty ---

EP Table
--- empty ---

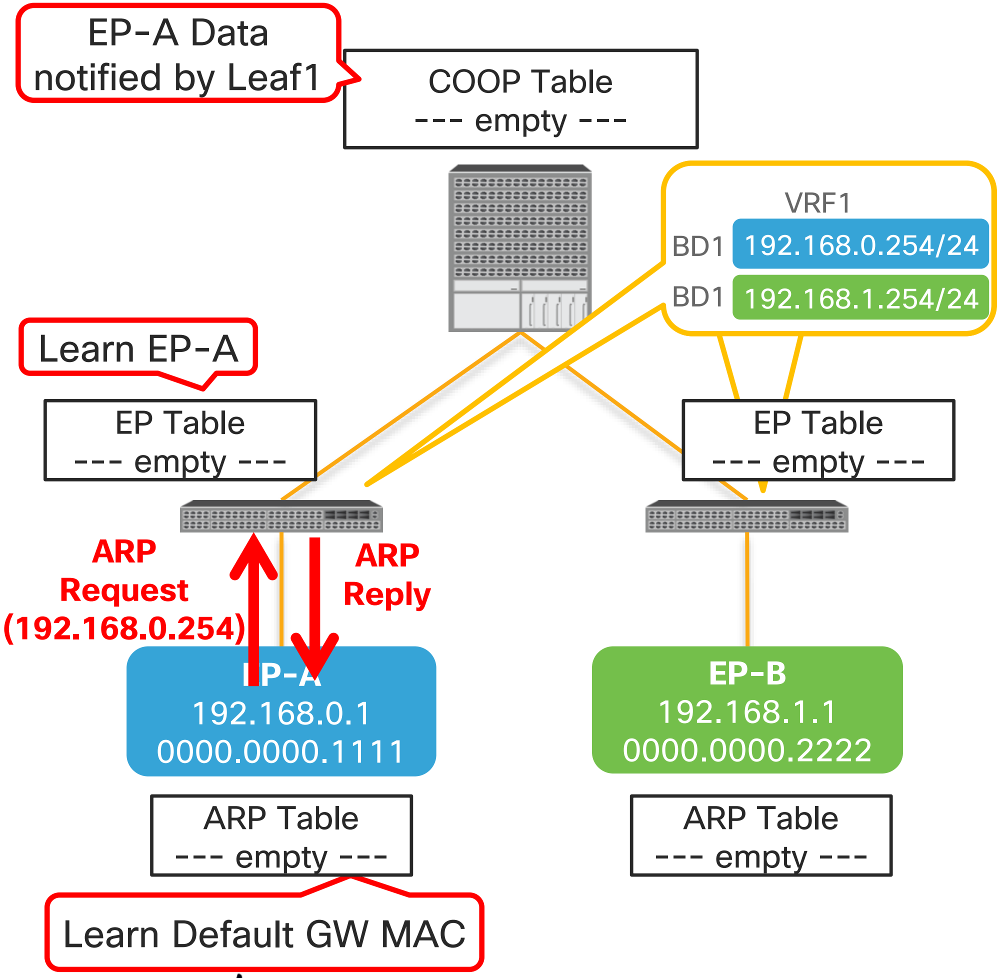
EP-A
192.168.0.1
0000.0000.1111

EP-B
192.168.1.1
0000.0000.2222

ARP Table
--- empty ---

ARP Table
--- empty ---

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      Fabric    H/W id  BD VLAN    Endpoint
                        (Type Value)      Encap                         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
+--------------------+---------------+-----------------+--------------+-------------+-------------------+
      VLAN/                Encap           MAC Address       MAC Info/       Interface     Endpoint Group
      Domain               VLAN            IP Address        IP Info                       Info
+--------------------+---------------+-----------------+--------------+-------------+-------------------+
69                              vlan-753    0000.0000.1111 L                  eth1/11          TK:APP1:EPG1
TK:VRF1                         vlan-753       192.168.0.1 L                  eth1/11
```

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], 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

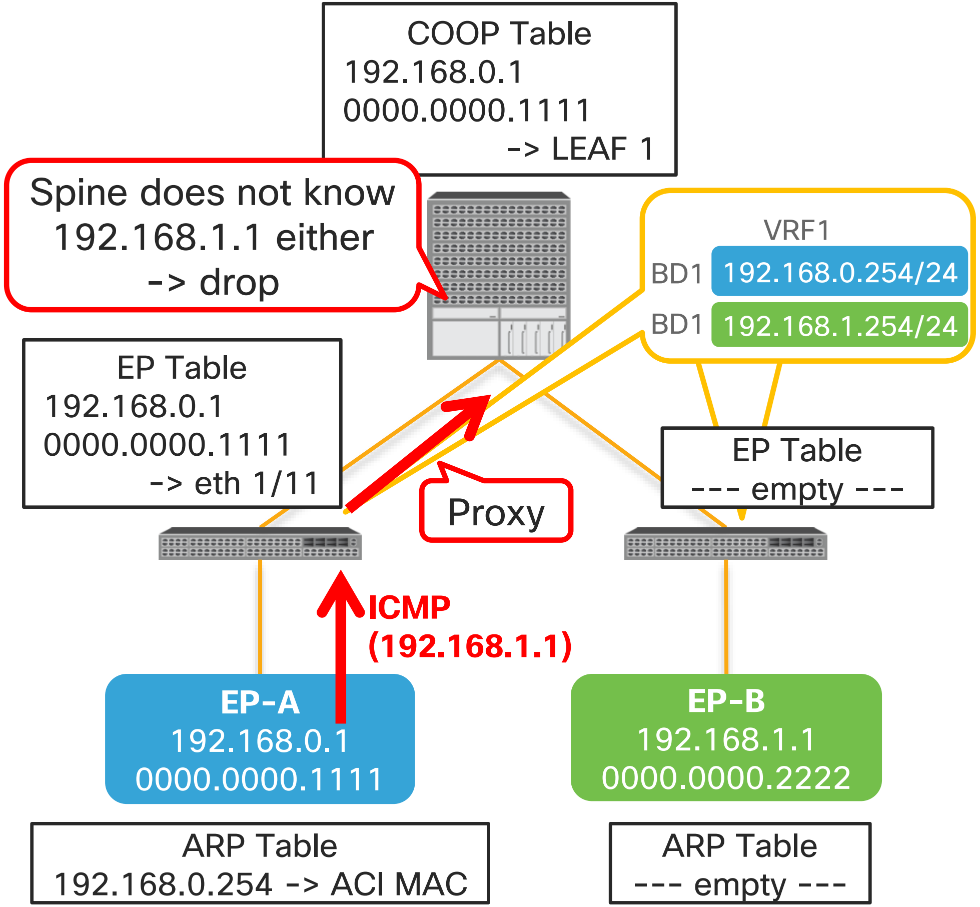
```
fab2-spine1# show coop internal info ip-db key 2228224 192.168.0.1

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 snipped



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/                  Encap          MAC Address        MAC Info/        Interface      Endpoint Group
      Domain                 VLAN           IP Address         IP Info                         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, local
```

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)

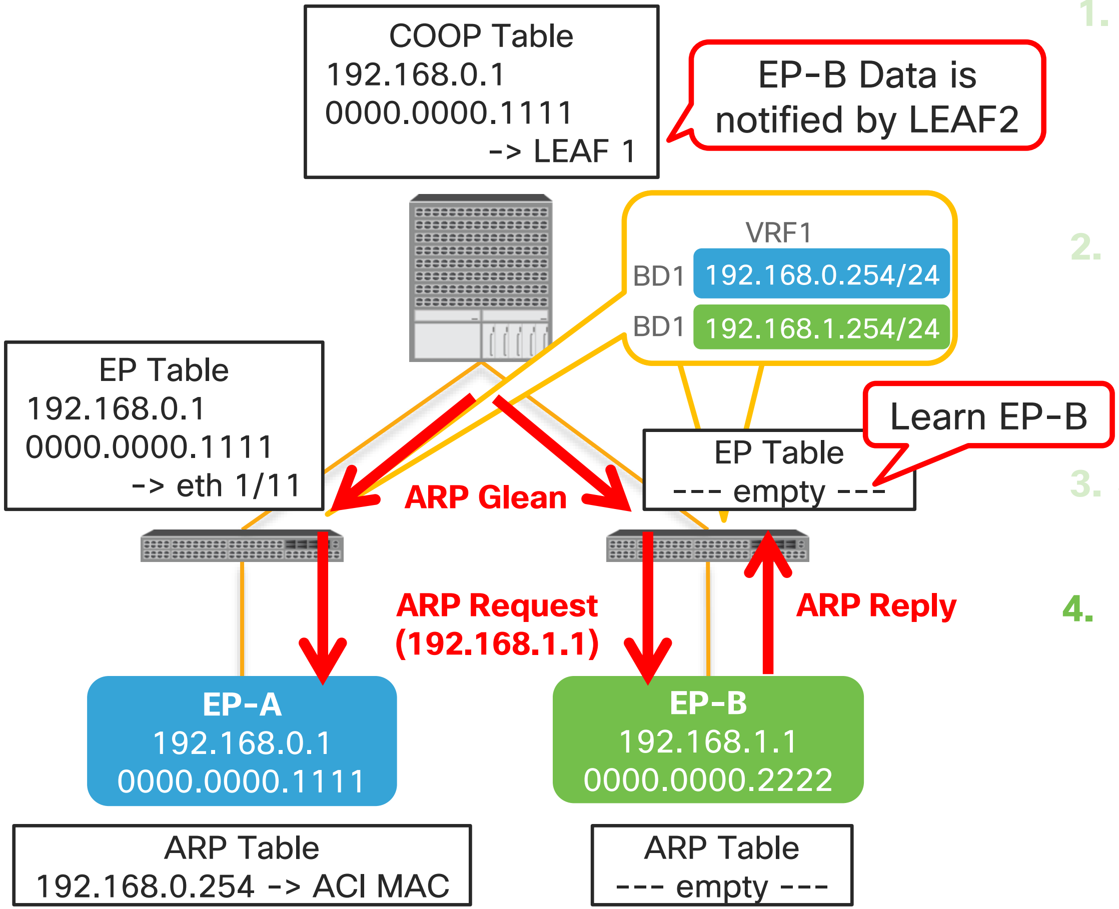
CLI notes (Coop Check)

```
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 (192.168.1.0/254)
      - ➢ 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      Fabric     H/W id  BD VLAN    Endpoint
                           (Type Value)      Encap                          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

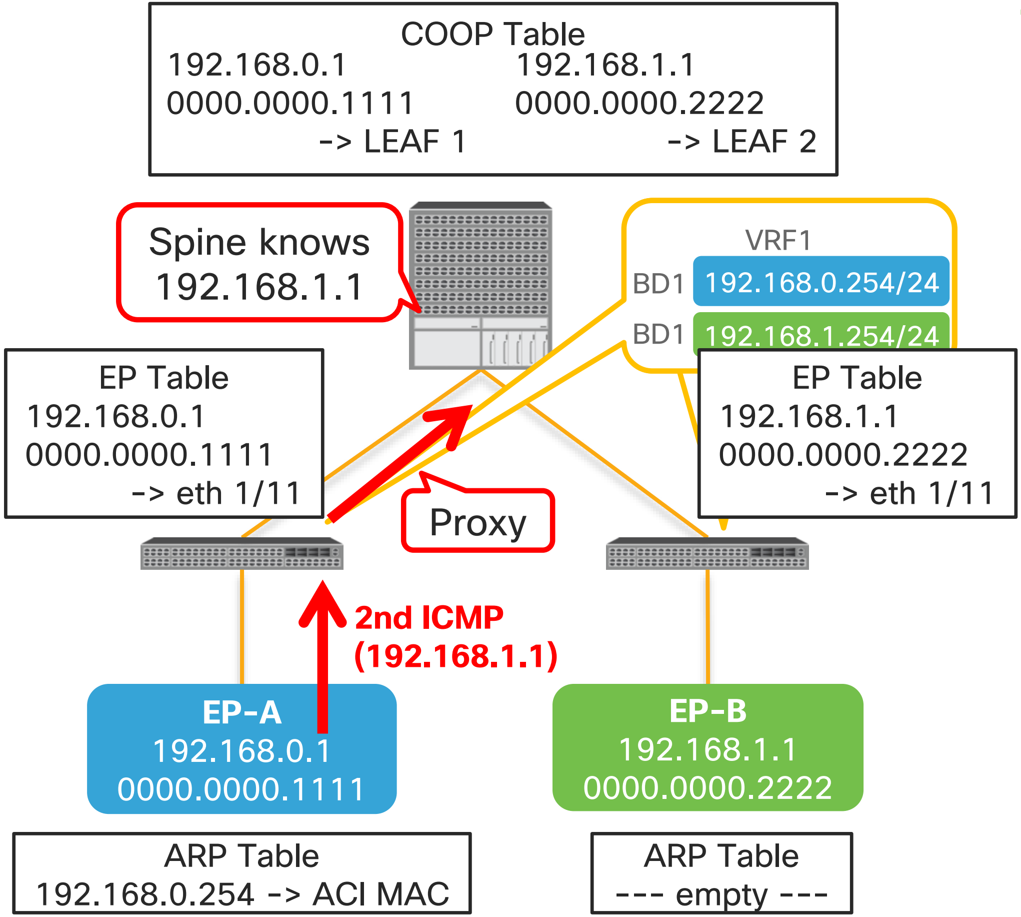
CLI notes (LEAF2 Source learning)

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

EndPoint Table
(= host table)

```
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)



COOP Table
192.168.0.1
0000.0000.1111
-> LEAF 1

192.168.1.1
0000.0000.2222
-> LEAF 2

Spine knows 192.168.1.1

VRF1
BD1 192.168.0.254/24
BD1 192.168.1.254/24

EP Table
192.168.0.1
0000.0000.1111
-> eth 1/11

EP Table
192.168.1.1
0000.0000.2222
-> eth 1/11

Proxy

2nd ICMP (192.168.1.1)

EP-A
192.168.0.1
0000.0000.1111

EP-B
192.168.1.1
0000.0000.2222

ARP Table
192.168.0.254 -> ACI MAC

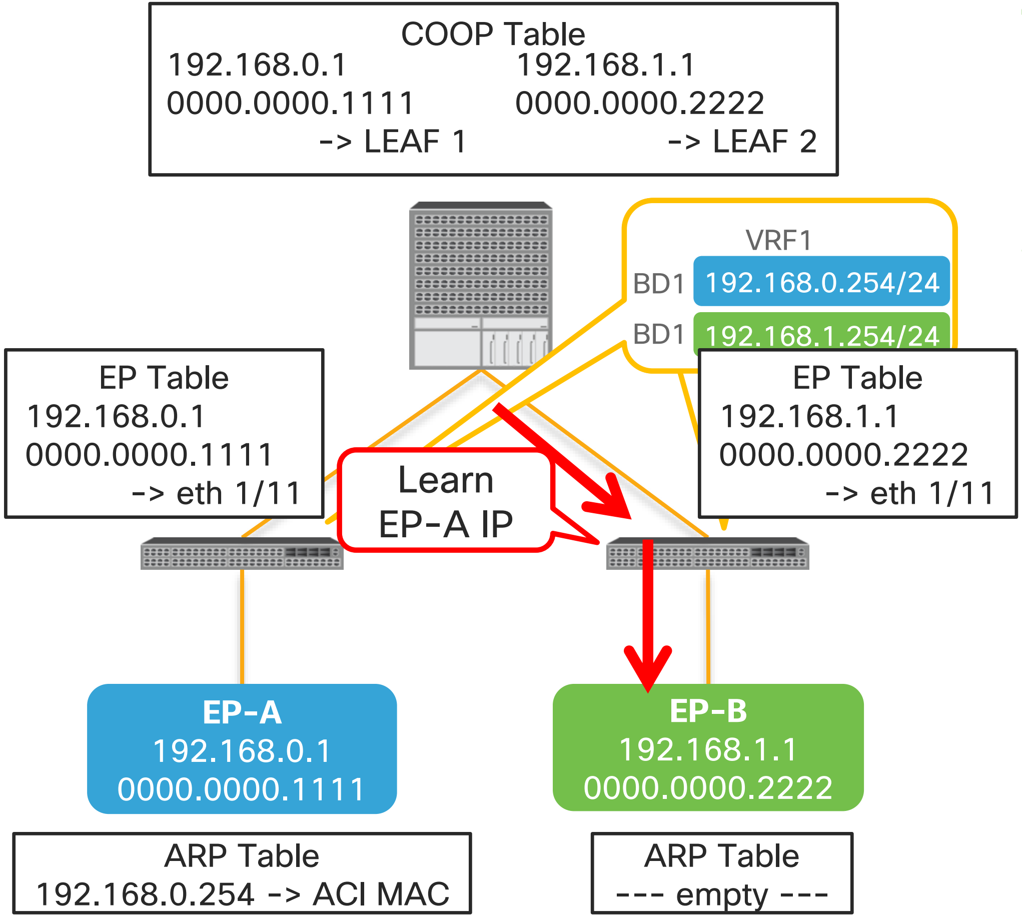
ARP Table
--- empty ---

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



COOP Table
192.168.0.1 0000.0000.1111 -> LEAF 1
192.168.1.1 0000.0000.2222 -> LEAF 2

VRF1
BD1 192.168.0.254/24
BD1 192.168.1.254/24

EP Table
192.168.0.1
0000.0000.1111
-> eth 1/11

Learn
EP-A IP

EP Table
192.168.1.1
0000.0000.2222
-> eth 1/11

EP-A
192.168.0.1
0000.0000.1111

EP-B
192.168.1.1
0000.0000.2222

ARP Table
192.168.0.254 -> ACI MAC

ARP Table
--- empty ---

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**

CLI notes (remote EP learning)

```
LEAF2# show endpoint vrf TK:VRF1 detail
Legend:
    --- snip ---
+--------------------------+---------------+-----------------+--------------+-------------+-------------------+
      VLAN/                     Encap           MAC Address      MAC Info/       Interface     Endpoint Group
      Domain                    VLAN            IP Address       IP Info                       Info
+--------------------------+---------------+-----------------+--------------+-------------+-------------------+
TK:VRF1                                         192.168.0.1                      tunnel11
10                                vlan-754    0000.0000.2222 L                    eth1/11        TK:APP1:EPG2
TK:VRF1                           vlan-754       192.168.1.1 L                    eth1/11
```

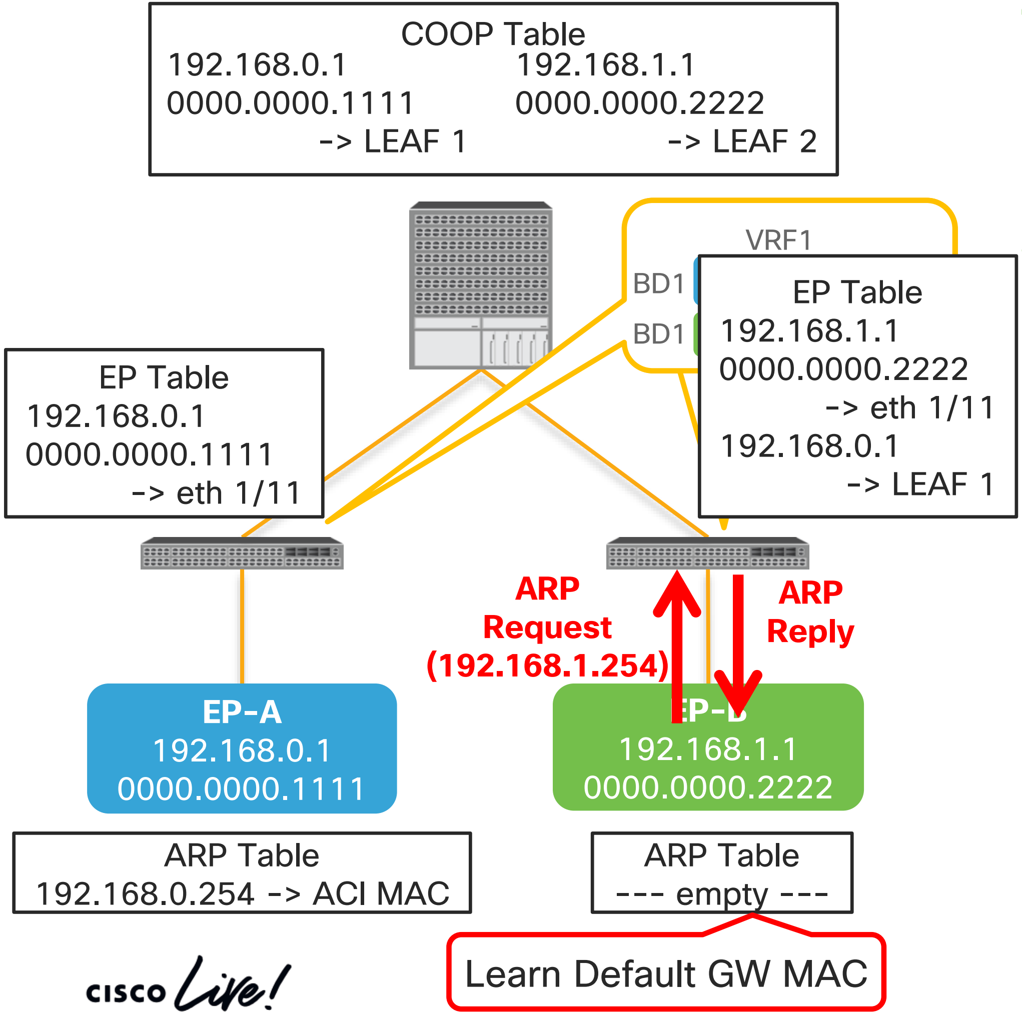
Learn Remote EP

```
LEAF2# show int tunnel 11 | grep dest
    Tunnel destination 10.0.8.95
LEAF2# acidiag fnvread | 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 : Tunnel11
Flags : 0x80004400 ::: sclass : 5479 ::: Ref count : 3
--- snip ---
```

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



COOP Table
192.168.0.1
0000.0000.1111
-> LEAF 1
192.168.1.1
0000.0000.2222
-> LEAF 2

EP Table
192.168.0.1
0000.0000.1111
-> eth 1/11

EP Table
192.168.1.1
0000.0000.2222
-> eth 1/11
192.168.0.1
-> LEAF 1

EP-A
192.168.0.1
0000.0000.1111

EP-B
192.168.1.1
0000.0000.2222

ARP Table
192.168.0.254 -> ACI MAC

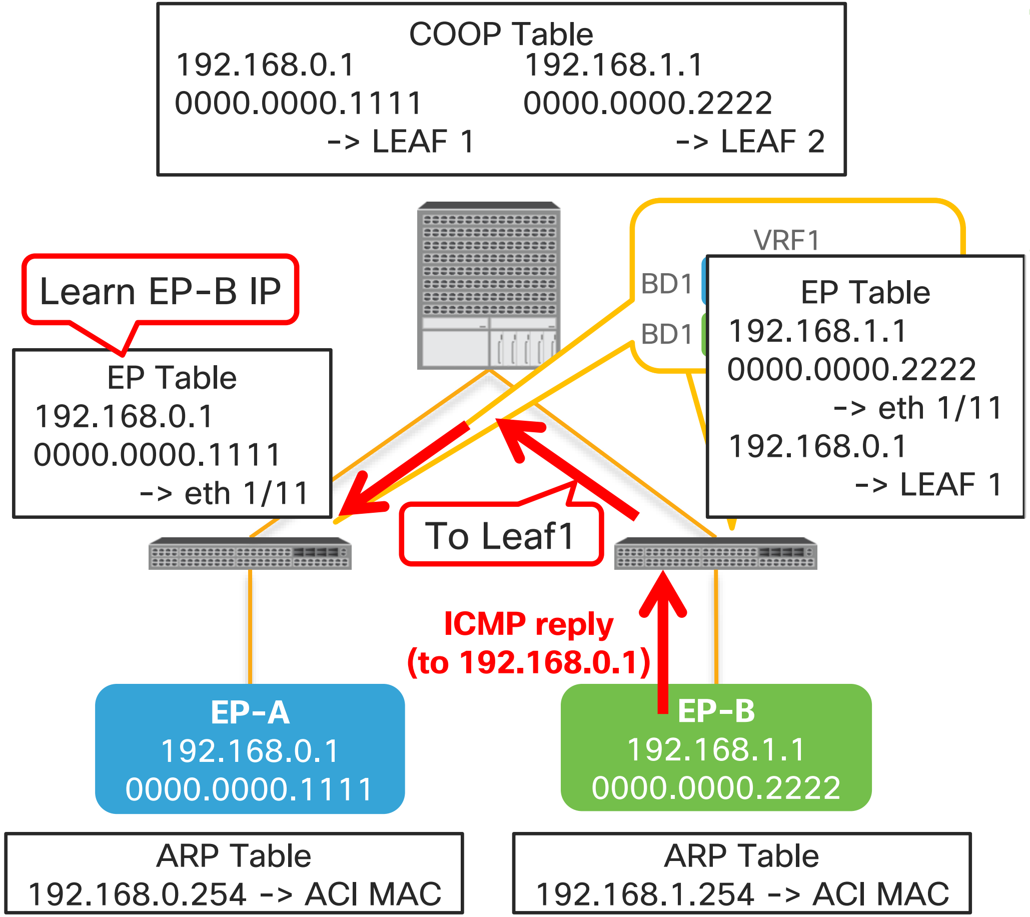
ARP Table
--- empty ---

Learn Default GW MAC

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)**



COOP Table
192.168.0.1
0000.0000.1111
-> LEAF 1
192.168.1.1
0000.0000.2222
-> LEAF 2

Learn EP-B IP

EP Table
192.168.0.1
0000.0000.1111
-> eth 1/11

EP Table
192.168.1.1
0000.0000.2222
-> eth 1/11
192.168.0.1
-> LEAF 1

EP-A
192.168.0.1
0000.0000.1111

EP-B
192.168.1.1
0000.0000.2222

ARP Table
192.168.0.254 -> ACI MAC

ARP Table
192.168.1.254 -> ACI MAC

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/                           Encap           MAC Address       MAC Info/      Interface
      Domain                          VLAN            IP Address        IP Info
+-----------------------------------+---------------+-----------------+--------------+-------------+
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
```

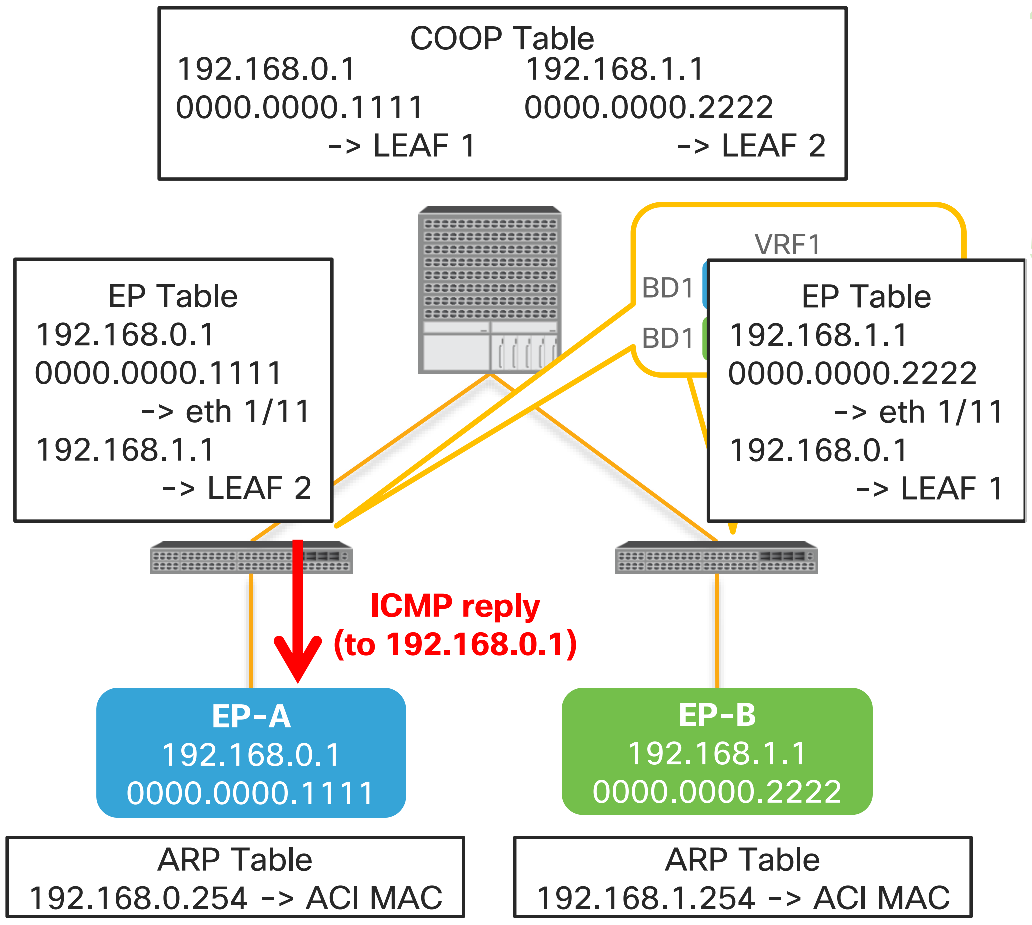
Learn Remote EP

```
LEAF1# show int tunnel 6 | grep dest
    Tunnel destination 10.0.8.90
LEAF1# acidiag fnvread | 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



COOP Table
192.168.0.1
0000.0000.1111
-> LEAF 1

192.168.1.1
0000.0000.2222
-> LEAF 2

VRF1
BD1
BD1

EP Table
192.168.0.1
0000.0000.1111
-> eth 1/11
192.168.1.1
-> LEAF 2

EP Table
192.168.1.1
0000.0000.2222
-> eth 1/11
192.168.0.1
-> LEAF 1

ICMP reply
(to 192.168.0.1)

EP-A
192.168.0.1
0000.0000.1111

EP-B
192.168.1.1
0000.0000.2222

ARP Table
192.168.0.254 -> ACI MAC

ARP Table
192.168.1.254 -> ACI MAC

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

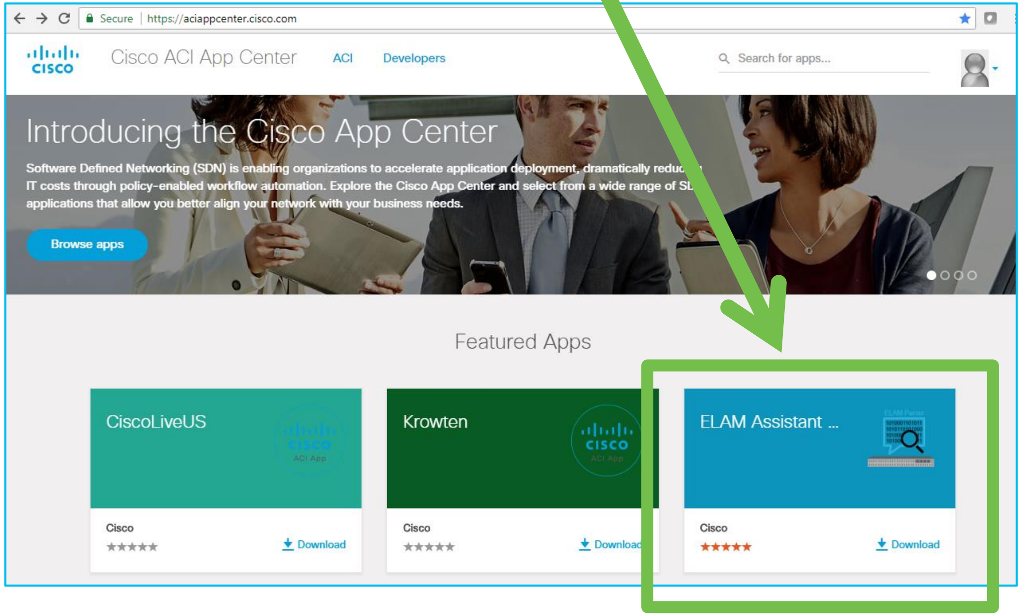
Packet Capture in ACI

ELAM Assistant in ACI AppCenter

Interested in more detail packet forwarding verification ?

➢ **ELAM Assistant!!**

https://dcappcenter.cisco.com



Introducing the Cisco App Center

Software Defined Networking (SDN) is enabling organizations to accelerate application deployment, dramatically reduc IT costs through policy-enabled workflow automation. Explore the Cisco App Center and select from a wide range of SL applications that allow you better align your network with your business needs.

ELAM (Embedded Logic Analyzer Module)

- Perform an ASIC level packet capture

ELAM Assistant

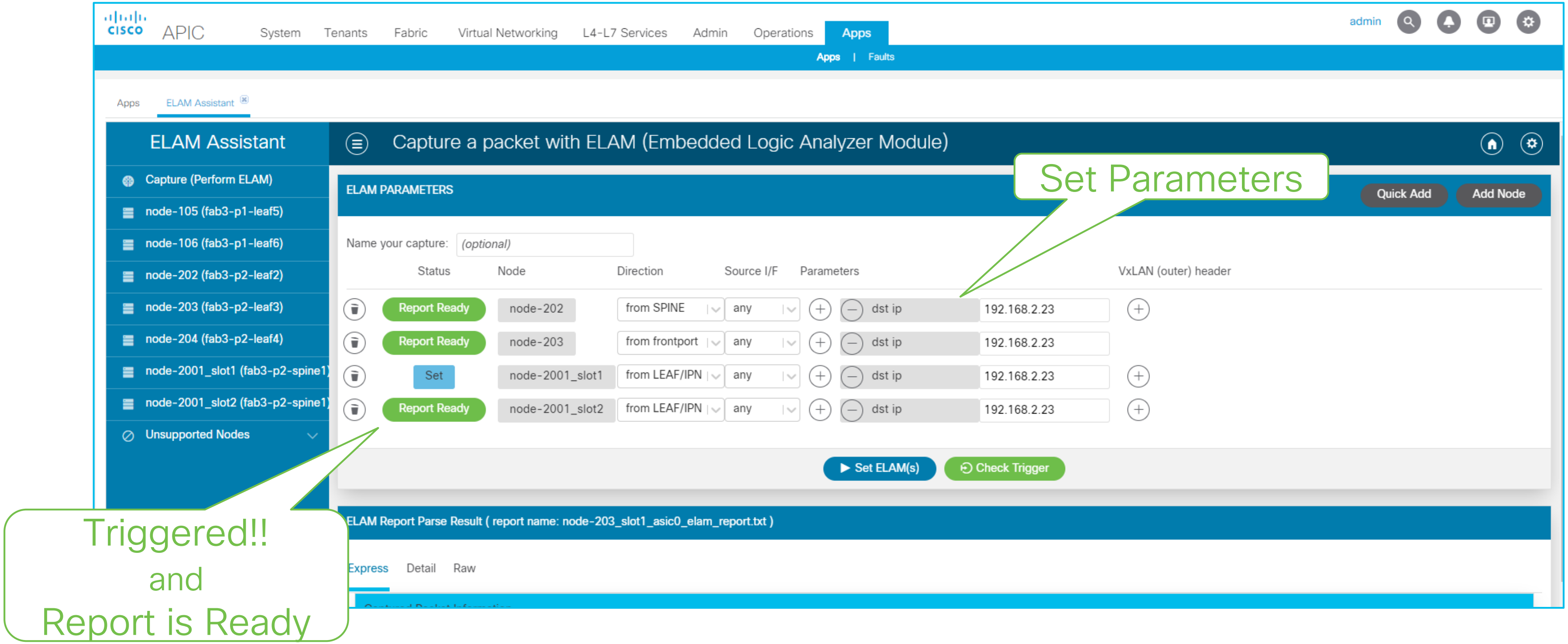
- You can perform ELAM like a TAC engineer!
- With a nicely formatted result report

Detail Explanations:

- https://dcappcenter.cisco.com/elam-assistant.html
  - ➢ How to use video, pictures
  - ➢ A download link for ELAM Assistant
- https://learningnetwork.cisco.com/docs/DOC-34985
  - ➢ ACI webinar for ELAM Assistant

ELAM Assistant in ACI AppCenter (example)

1. Perform ELAM



ELAM Assistant

- Capture (Perform ELAM)
- node-105 (fab3-p1-leaf5)
- node-106 (fab3-p1-leaf6)
- node-202 (fab3-p2-leaf2)
- node-203 (fab3-p2-leaf3)
- node-204 (fab3-p2-leaf4)
- node-2001_slot1 (fab3-p2-spine1)
- node-2001_slot2 (fab3-p2-spine1)
- Unsupported Nodes

Capture a packet with ELAM (Embedded Logic Analyzer Module)

ELAM PARAMETERS

Name your capture: *(optional)*

Status	Node	Direction	Source I/F	Parameters		VxLAN (outer) header
Report Ready	node-202	from SPINE	any	dst ip	192.168.2.23	+
Report Ready	node-203	from frontport	any	dst ip	192.168.2.23	
Set	node-2001_slot1	from LEAF/IPN	any	dst ip	192.168.2.23	+
Report Ready	node-2001_slot2	from LEAF/IPN	any	dst ip	192.168.2.23	+

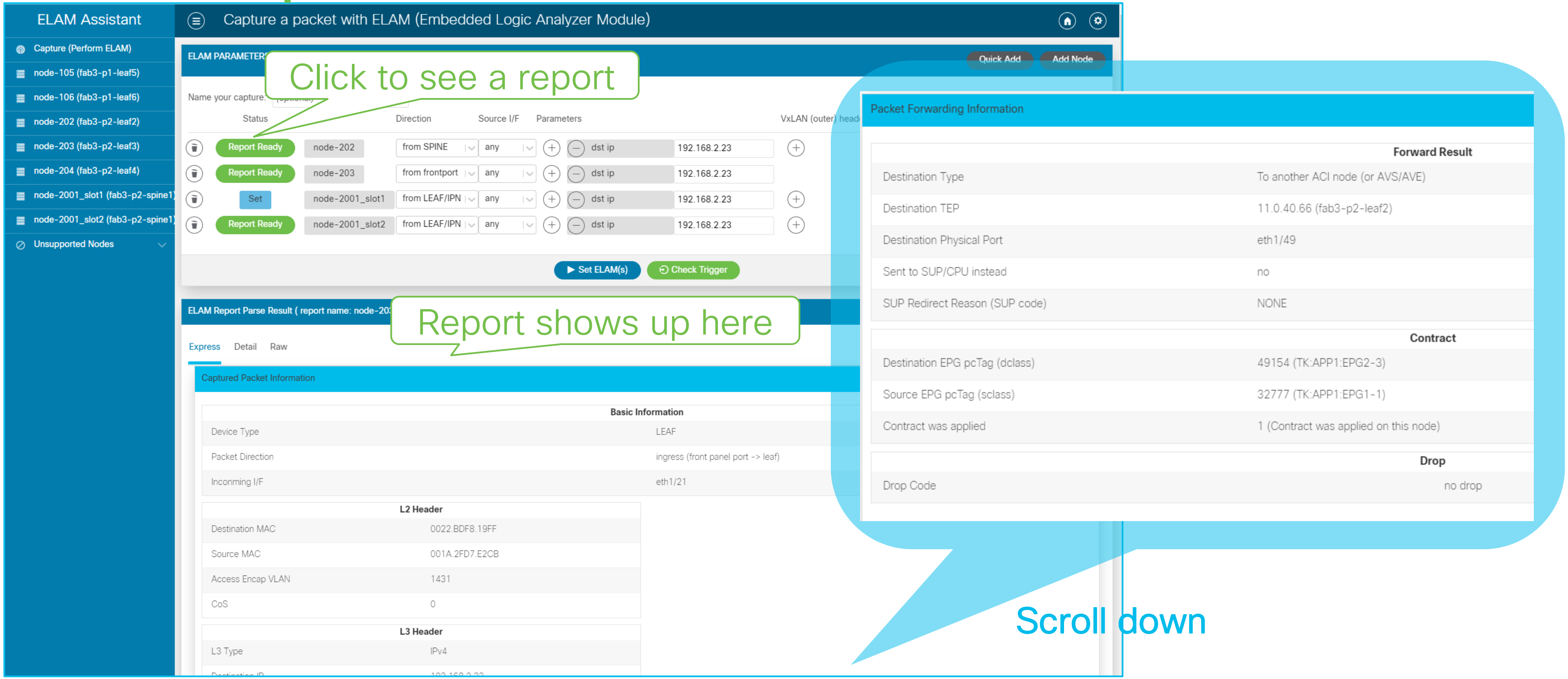
ELAM Report Parse Result (report name: node-203_slot1_asic0_elam_report.txt)

Set Parameters

Triggered!!
and
Report is Ready

ELAM Assistant in ACI AppCenter (example)

2. Read a report



Capture a packet with ELAM (Embedded Logic Analyzer Module)

ELAM PARAMETERS

Name your capture: (optional)

Status		Direction	Source I/F	Parameters		VxLAN (outer) header
Report Ready	node-202	from SPINE	any	dst ip	192.168.2.23	
Report Ready	node-203	from frontport	any	dst ip	192.168.2.23	
Set	node-2001_slot1	from LEAF/IPN	any	dst ip	192.168.2.23	
Report Ready	node-2001_slot2	from LEAF/IPN	any	dst ip	192.168.2.23	

Click to see a report

ELAM Report Parse Result (report name: node-203

Report shows up here

Captured Packet Information

Basic Information	
Device Type	LEAF
Packet Direction	ingress (front panel port -> leaf)
Inconming I/F	eth1/21

L2 Header	
Destination MAC	0022.BDF8.19FF
Source MAC	001A.2FD7.E2CB
Access Encap VLAN	1431
CoS	0

L3 Header	
L3 Type	IPv4
Destination IP	192.168.2.23

Packet Forwarding Information

Forward Result	
Destination Type	To another ACI node (or AVS/AVE)
Destination TEP	11.0.40.66 (fab3-p2-leaf2)
Destination Physical Port	eth1/49
Sent to SUP/CPU instead	no
SUP Redirect Reason (SUP code)	NONE

Contract	
Destination EPG pcTag (dclass)	49154 (TK:APP1:EPG2-3)
Source EPG pcTag (sclass)	32777 (TK:APP1:EPG1-1)
Contract was applied	1 (Contract was applied on this node)

Drop	
Drop Code	no drop

Scroll down

Complete your online 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 (starting on 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 Content Catalog on ciscolive.com/emea.

Cisco Live sessions will be available for viewing on demand after the event at ciscolive.com.

Continue your education

Demos in the Cisco campus

Walk-in labs

Meet the engineer 1:1 meetings

Related sessions

Thank you

You make **possible**