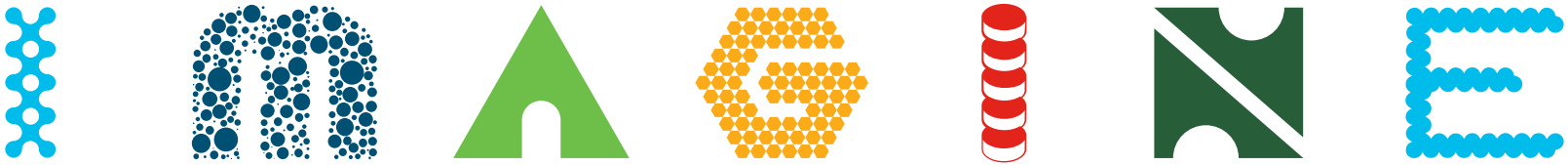


INTUITIVE

Cisco live!

5-8 March 2019 • Melbourne, Australia

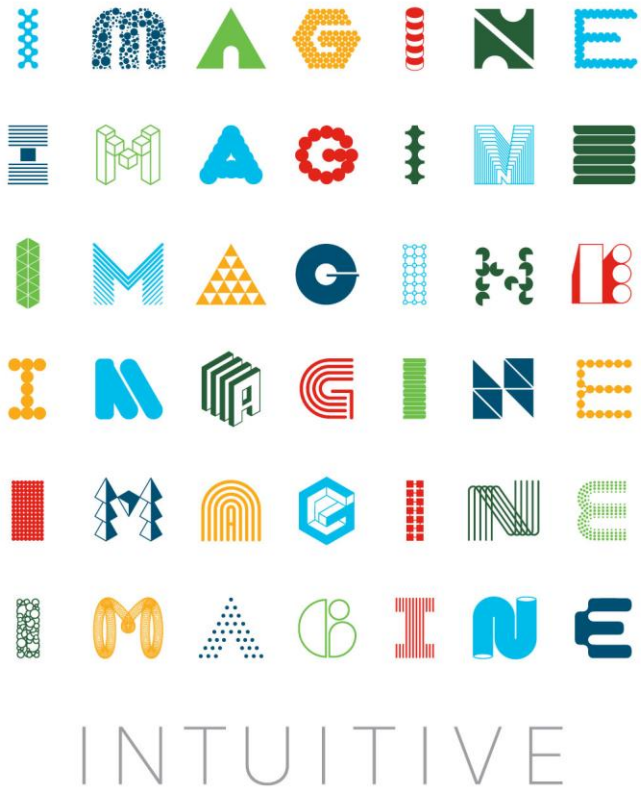
#CLMEL

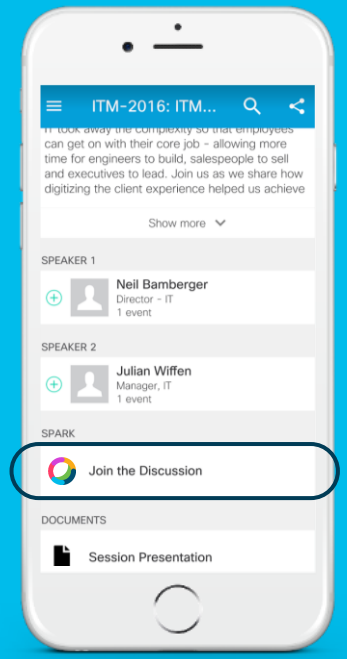
BRKACI-3545

Mastering ACI Forwarding Behavior

– A day in the life of a packet –

Takuya Kishida, Technical Leader, Service





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. Open the Cisco Events Mobile App
2. Find your desired session in the “Session Scheduler”
3. Click “Join the Discussion”
4. Install Webex Teams or go directly to the team space
5. 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

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

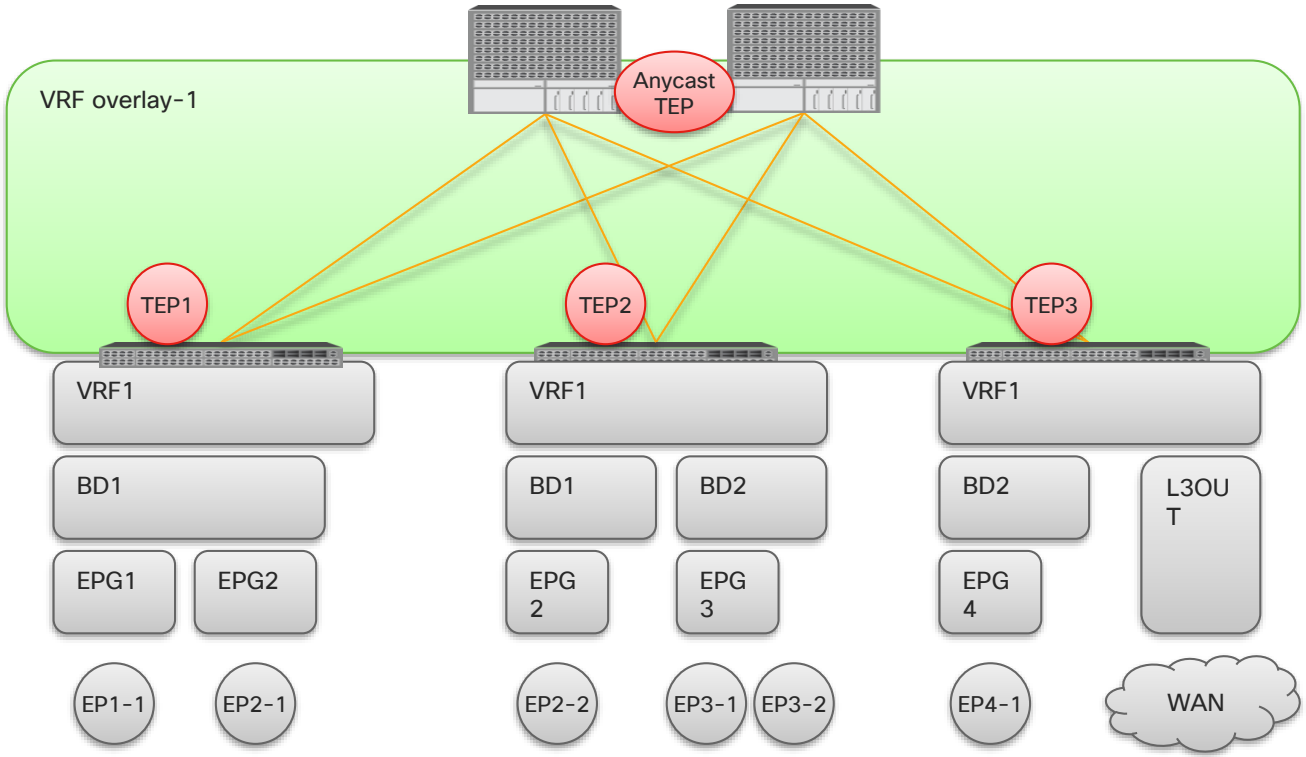
Acronyms	Definitions
d*XXX*o	Outer Destination XXX (dIPo = Outer Destination IP)
s*XXX*o	Outer Source XXX (sIPo = Outer Source IP)
d*XXX*i	Inner Destination XXX (dIPi = Inner Destination IP)
s*XXX*i	Inner Source XXX (sIPi = Inner Source IP)
GIPo	Outer Multicast Group IP
VNID	Virtual Network Identifier

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

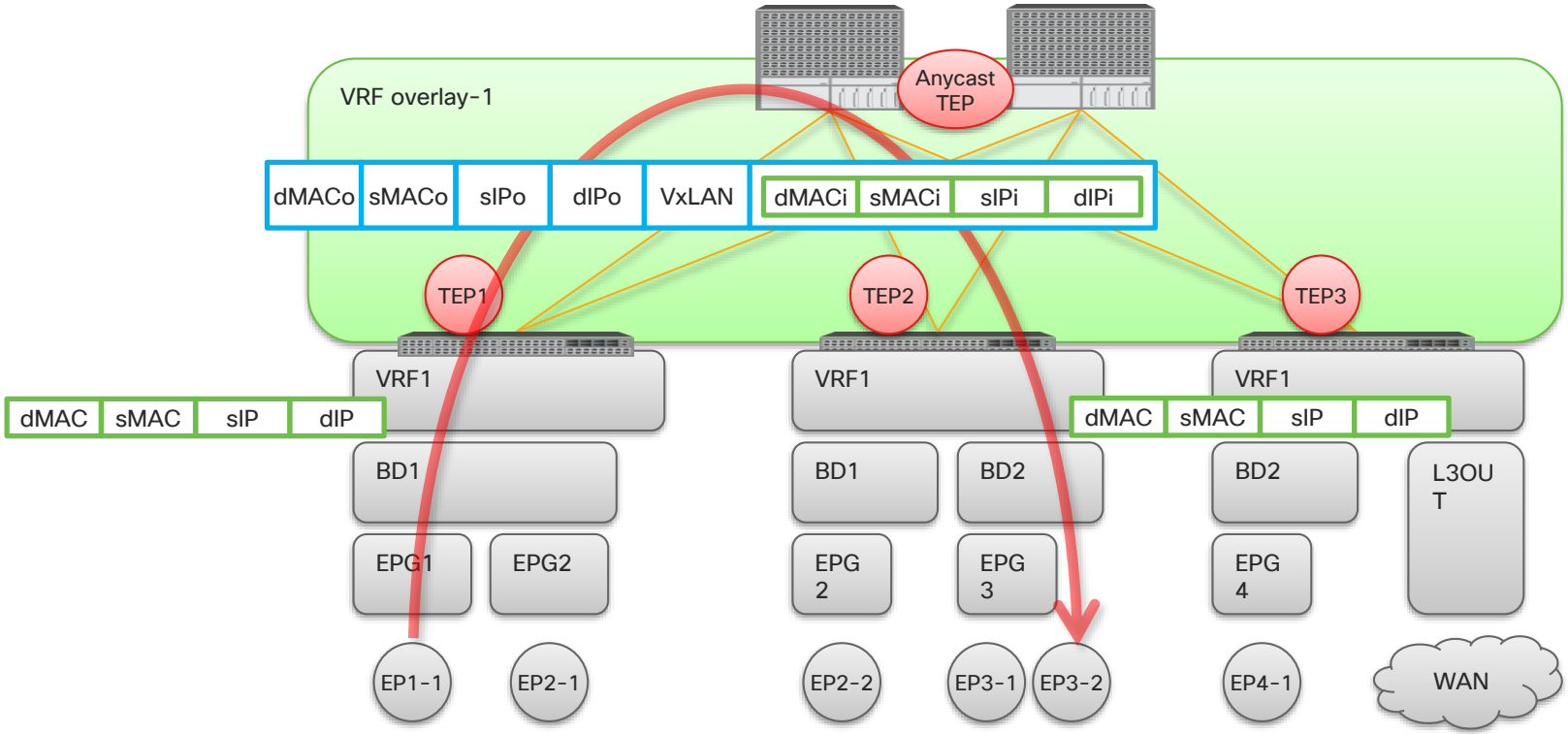
ACI Overlay VxLAN and TEP

※ TEP : Tunnel EndPoint



ACI Overlay VxLAN and TEP

※ TEP : Tunnel EndPoint



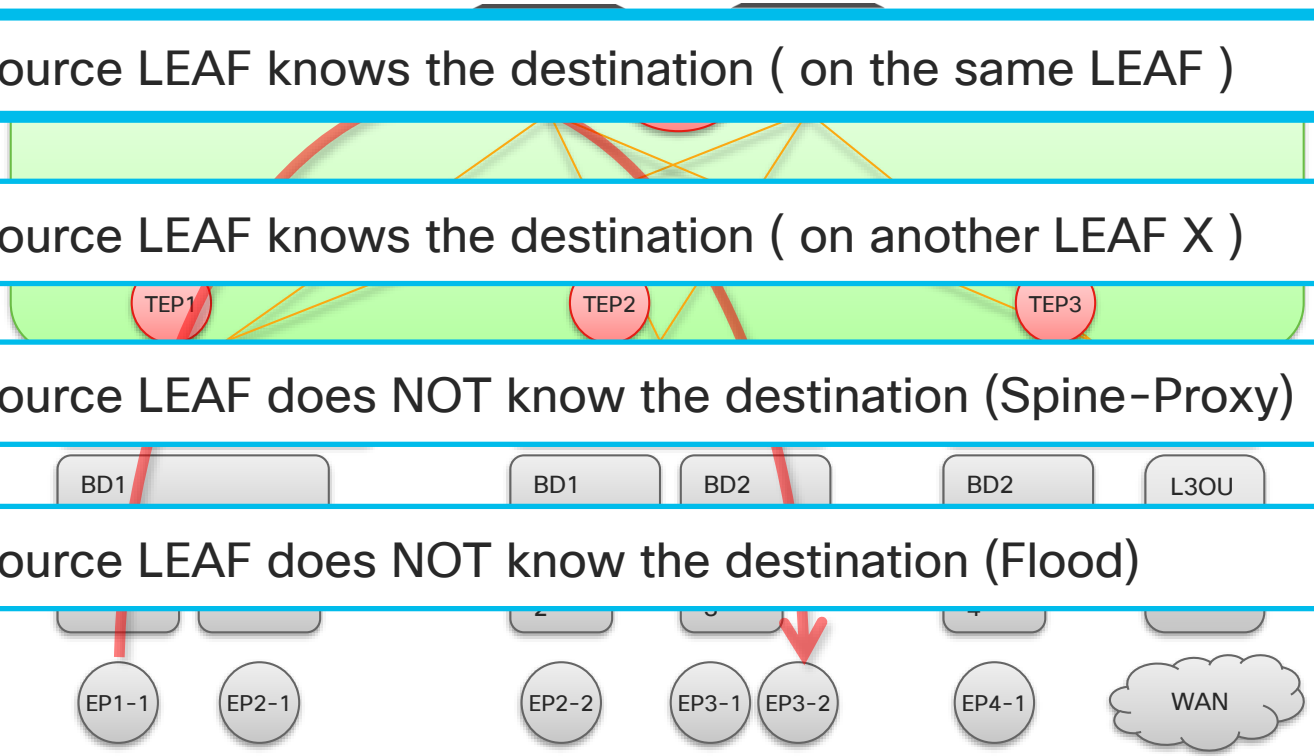
ACI Overlay VxLAN and TEP

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

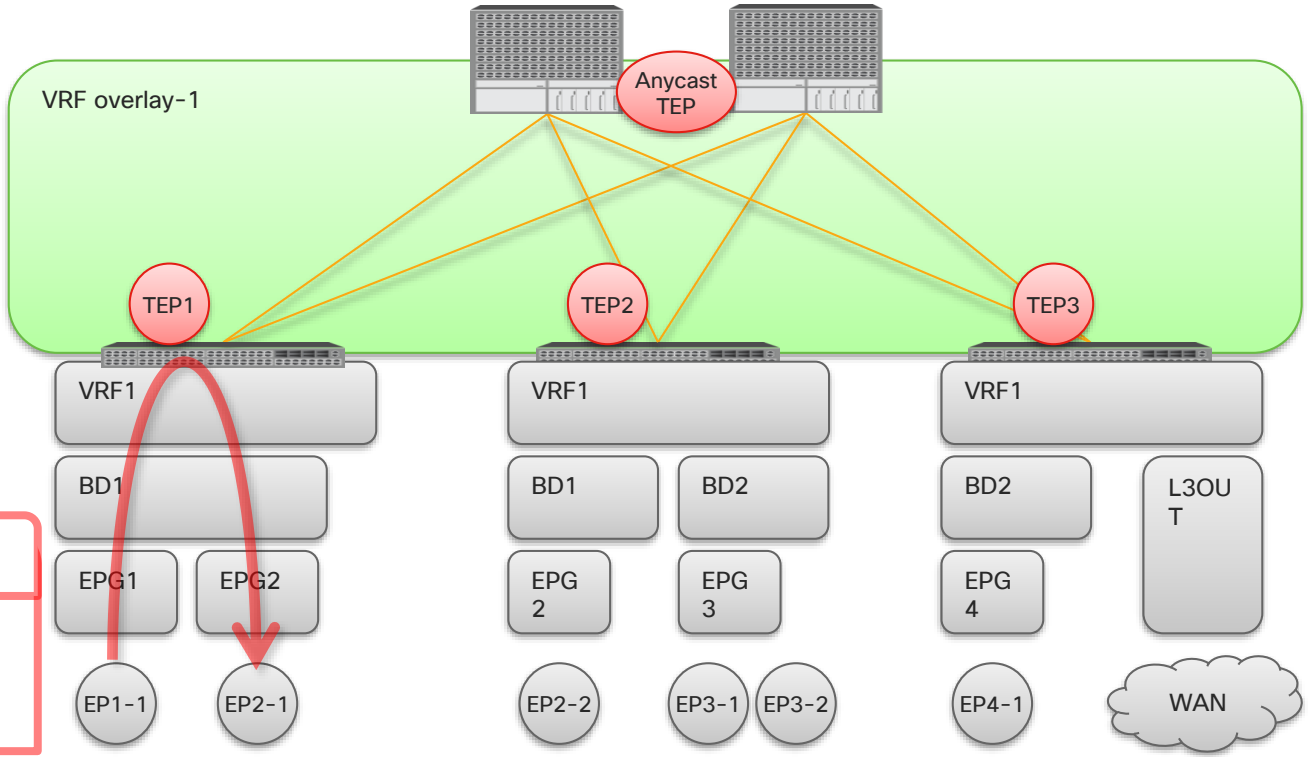
Scenario 2 : source LEAF knows the destination (on another LEAF X)

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

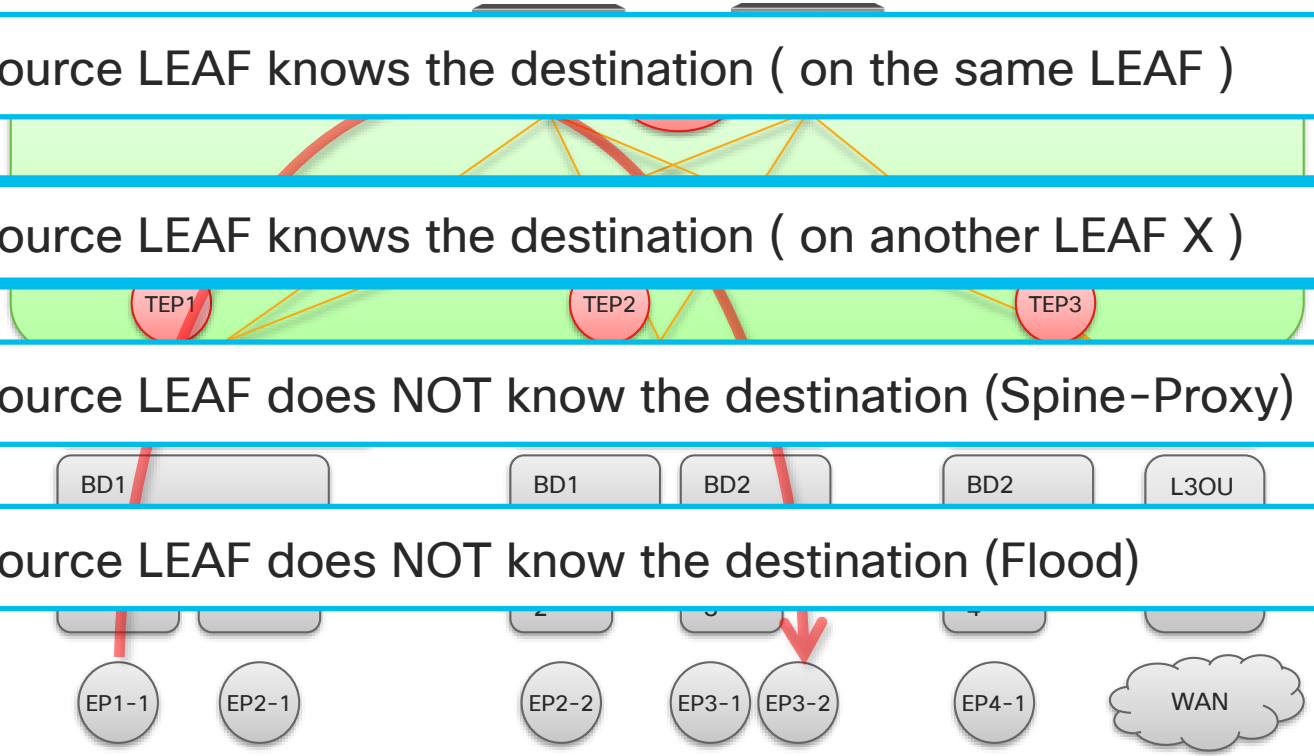
ACI Overlay VxLAN and TEP

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

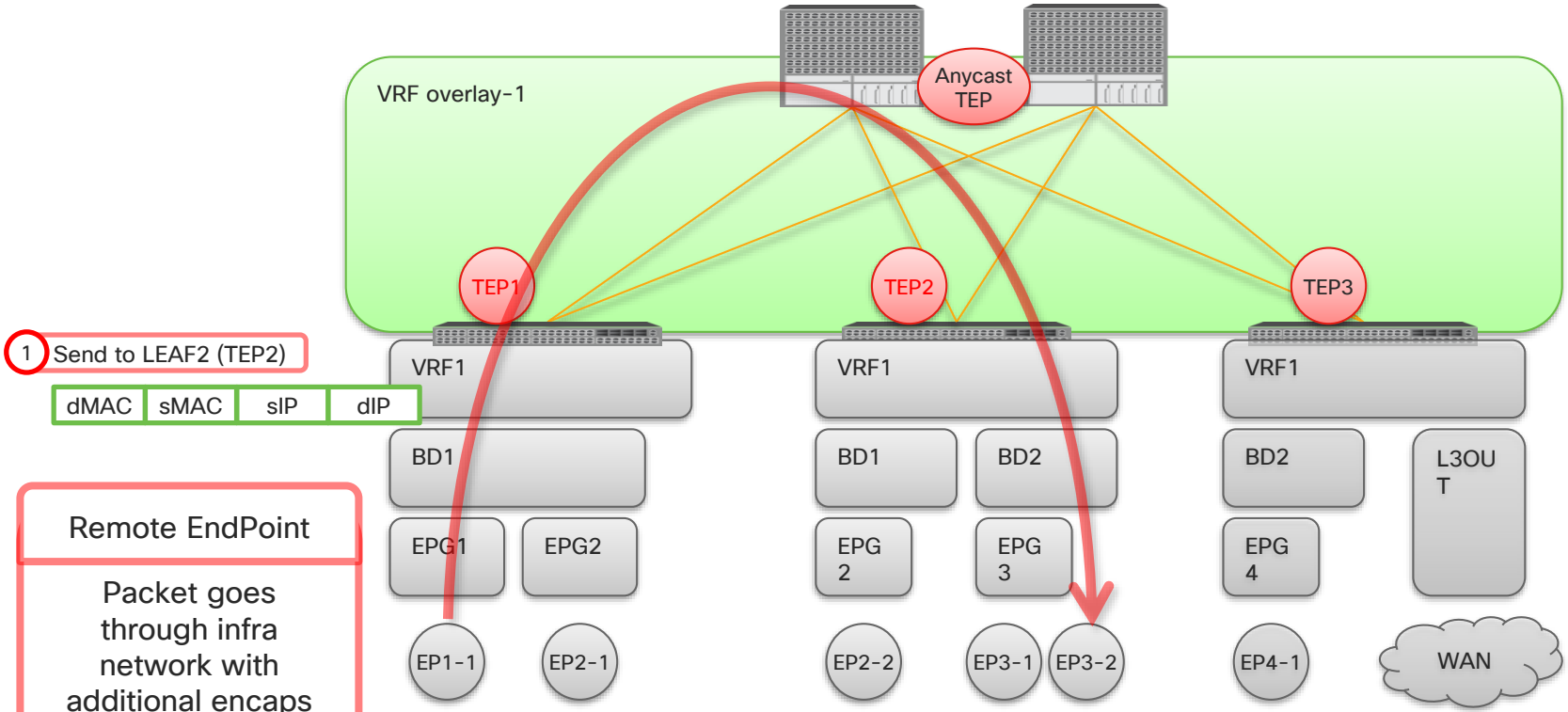
Scenario 2 : source LEAF knows the destination (on another LEAF X)

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)

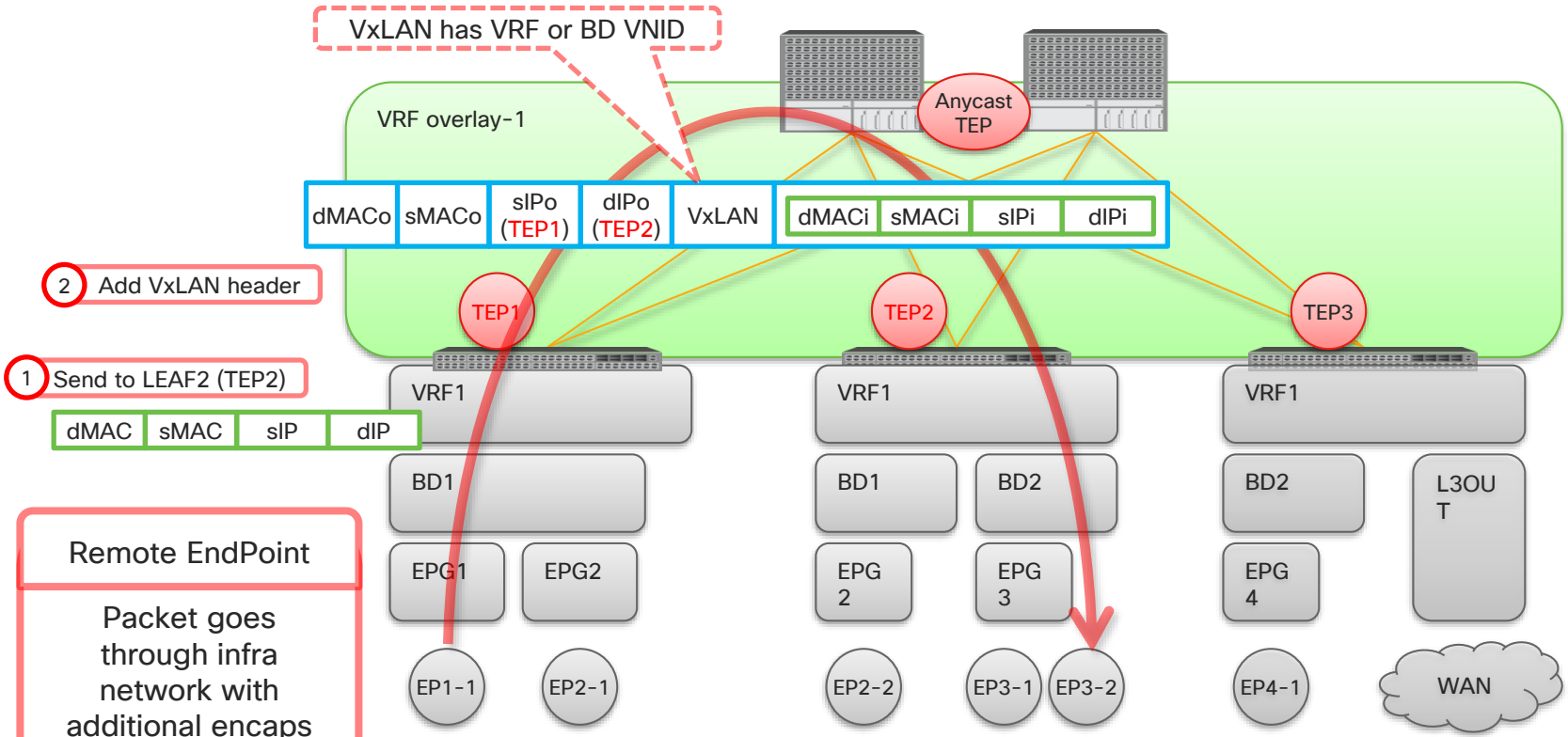


Remote EndPoint

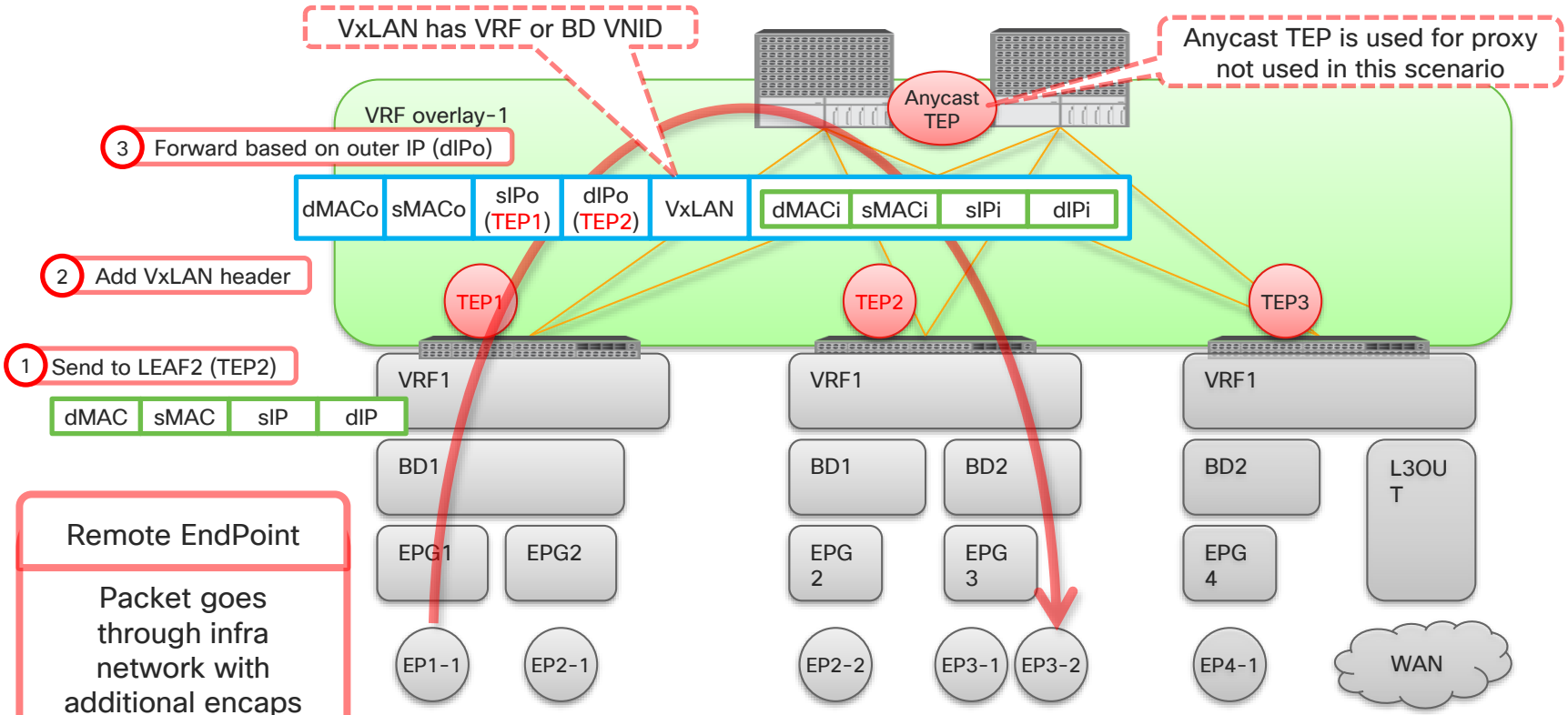
Packet goes through infra network with additional encaps (iVxLAN)

※ 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)

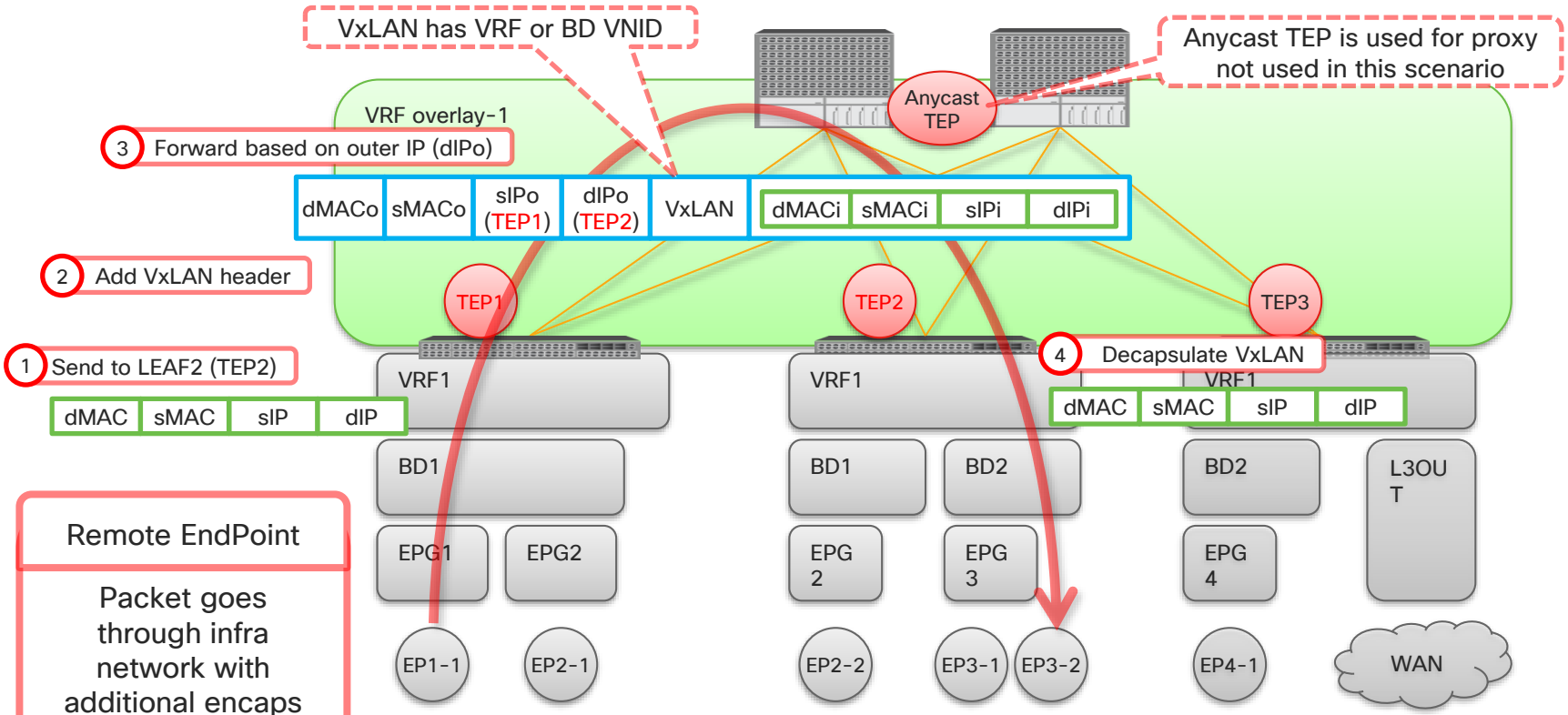


Remote EndPoint

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)

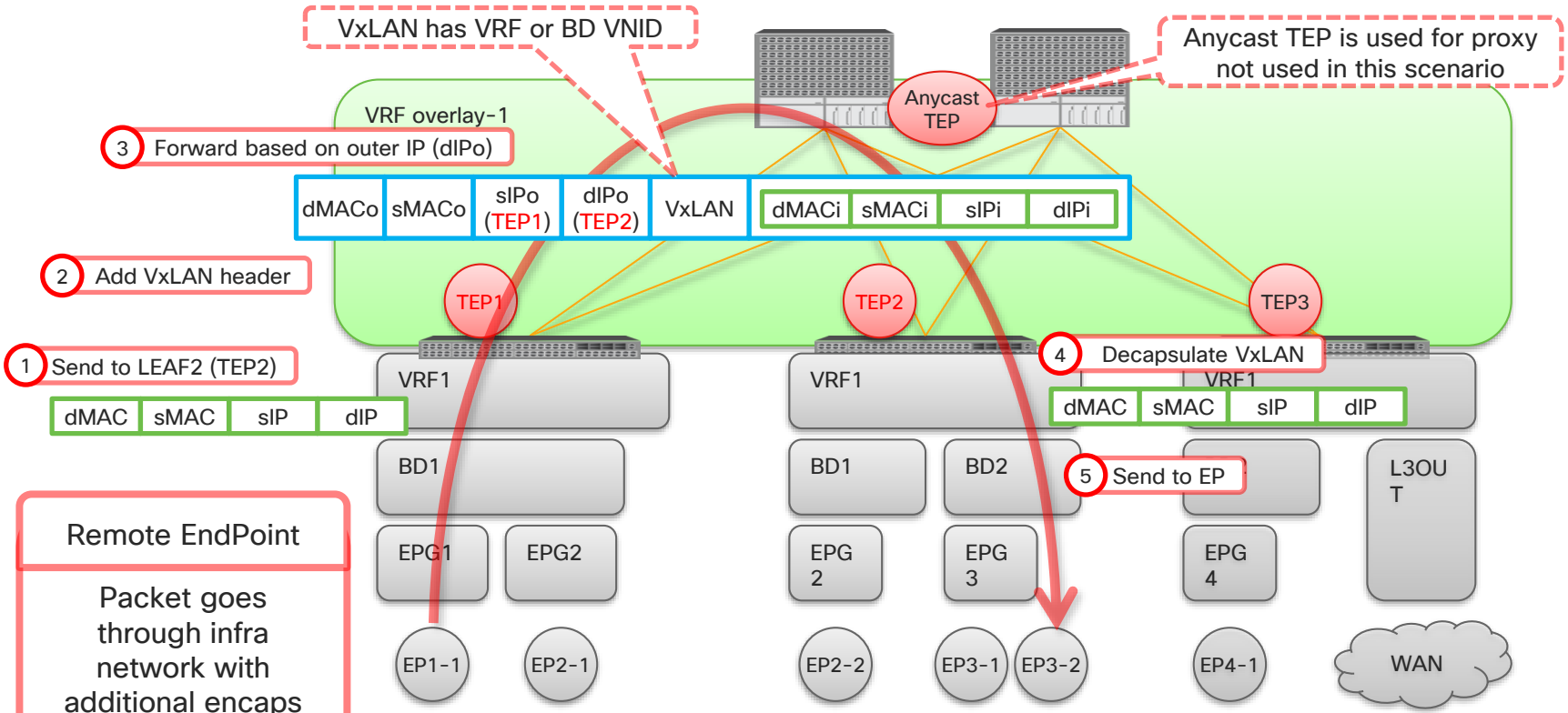


Remote EndPoint

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

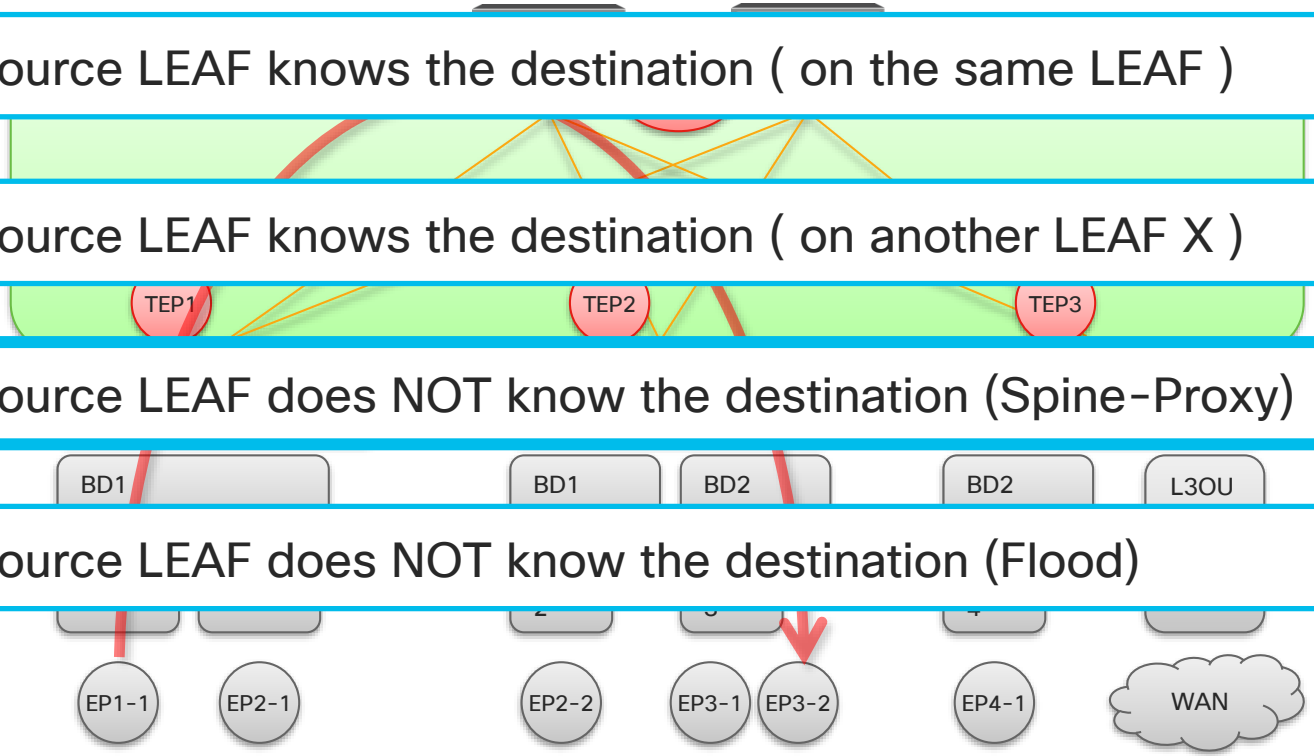
ACI Overlay VxLAN and TEP

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

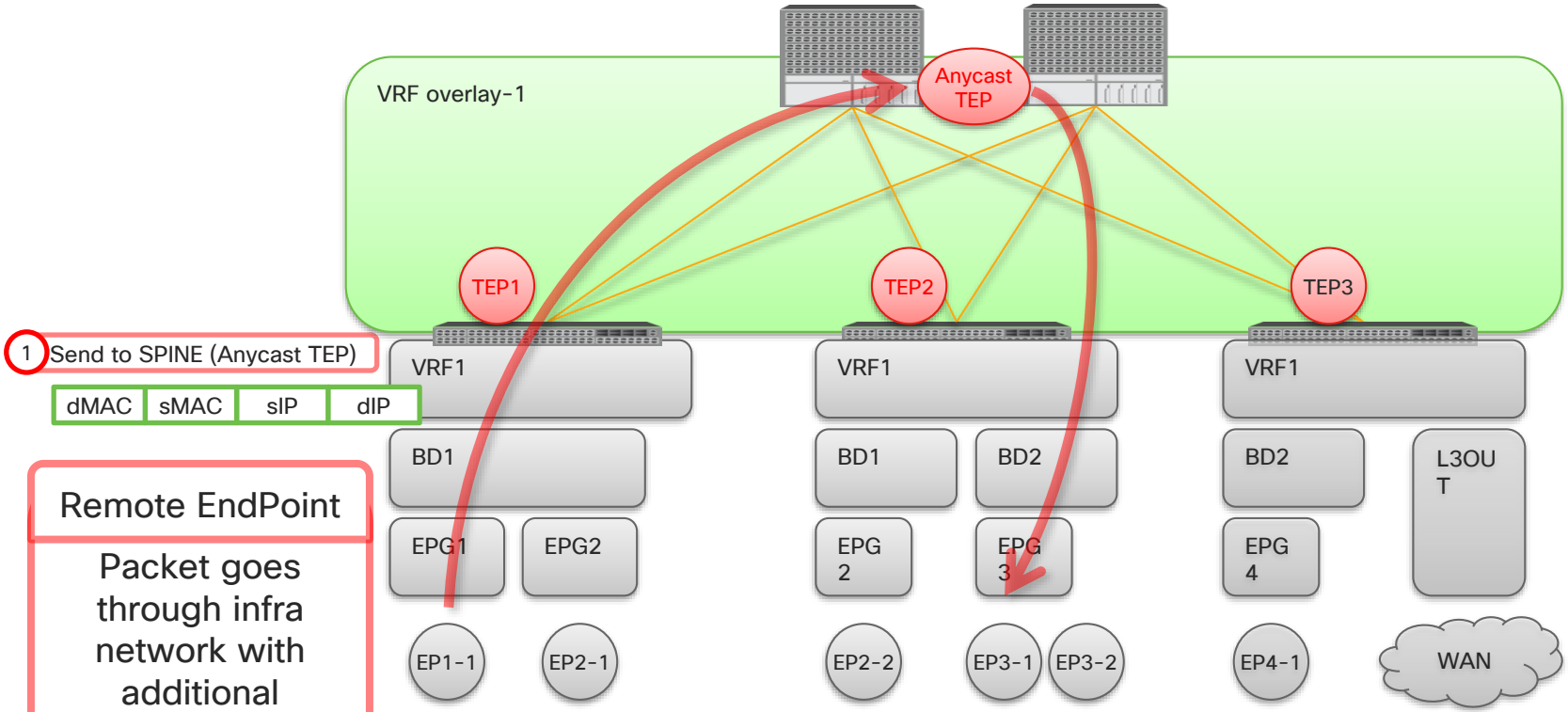
Scenario 2 : source LEAF knows the destination (on another LEAF X)

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)

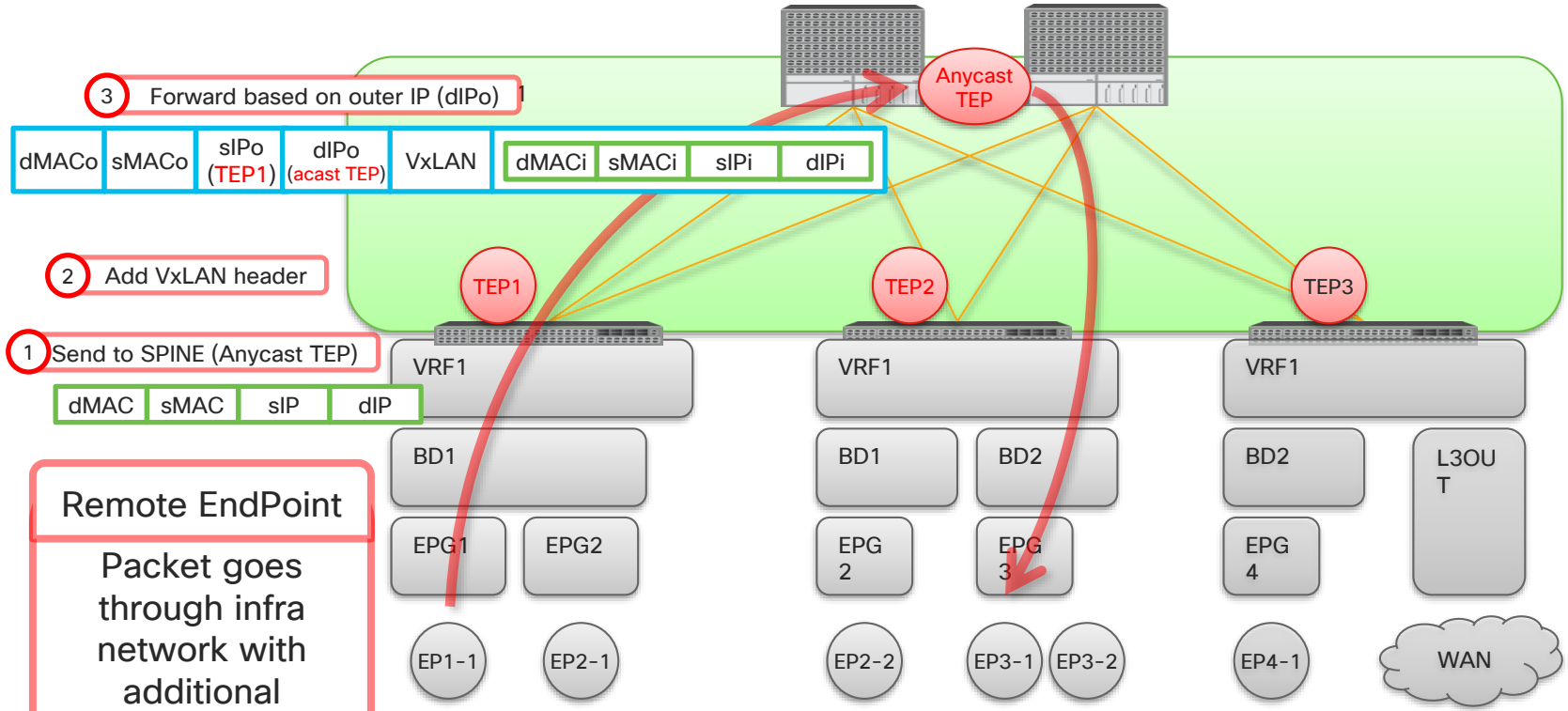


Remote EndPoint

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)

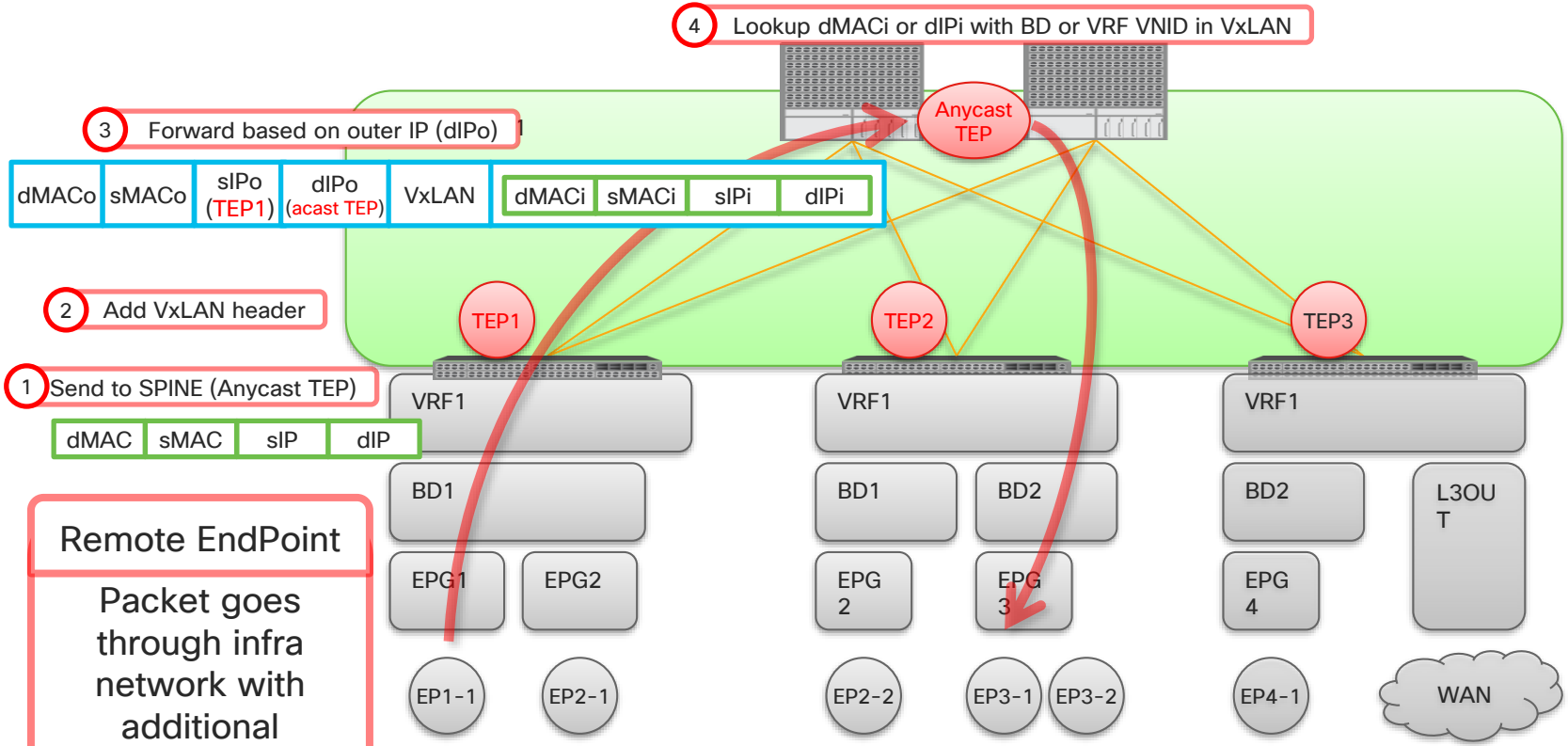


Remote EndPoint

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)

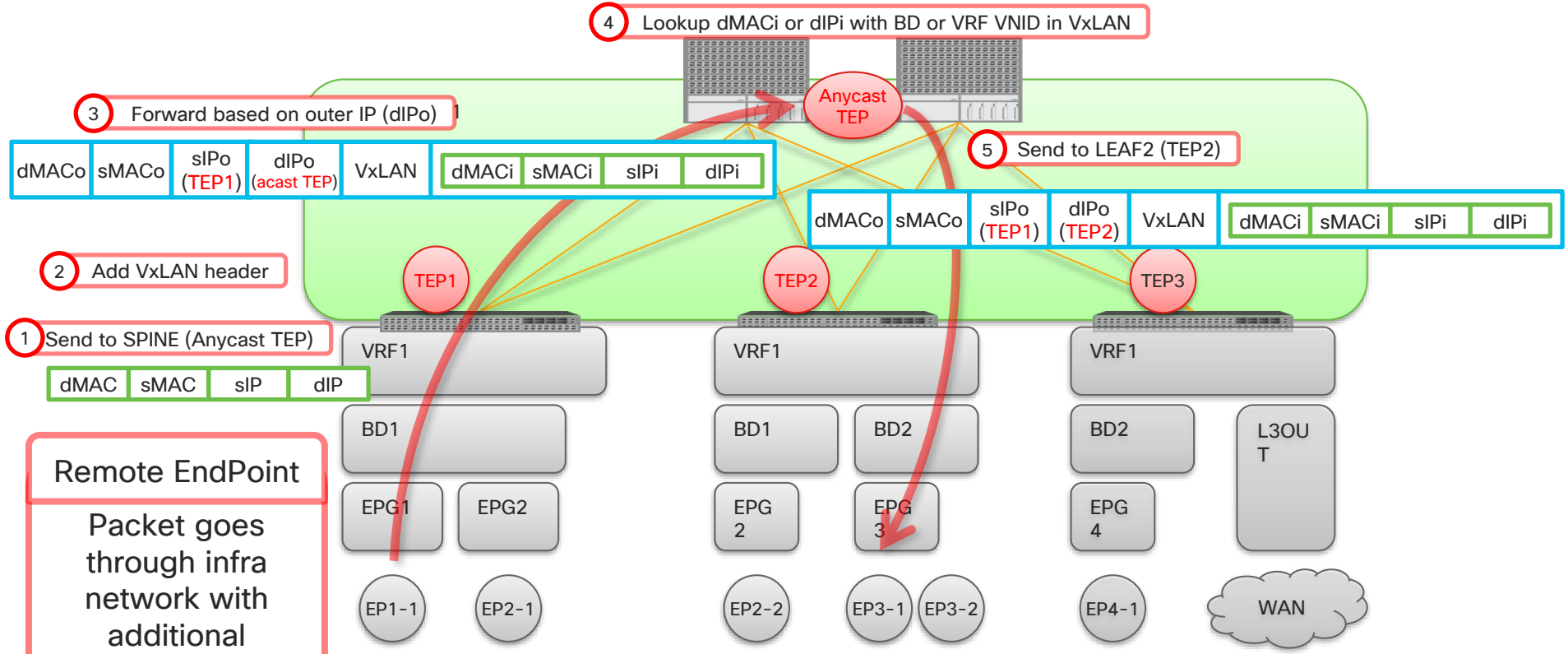


Remote EndPoint

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)

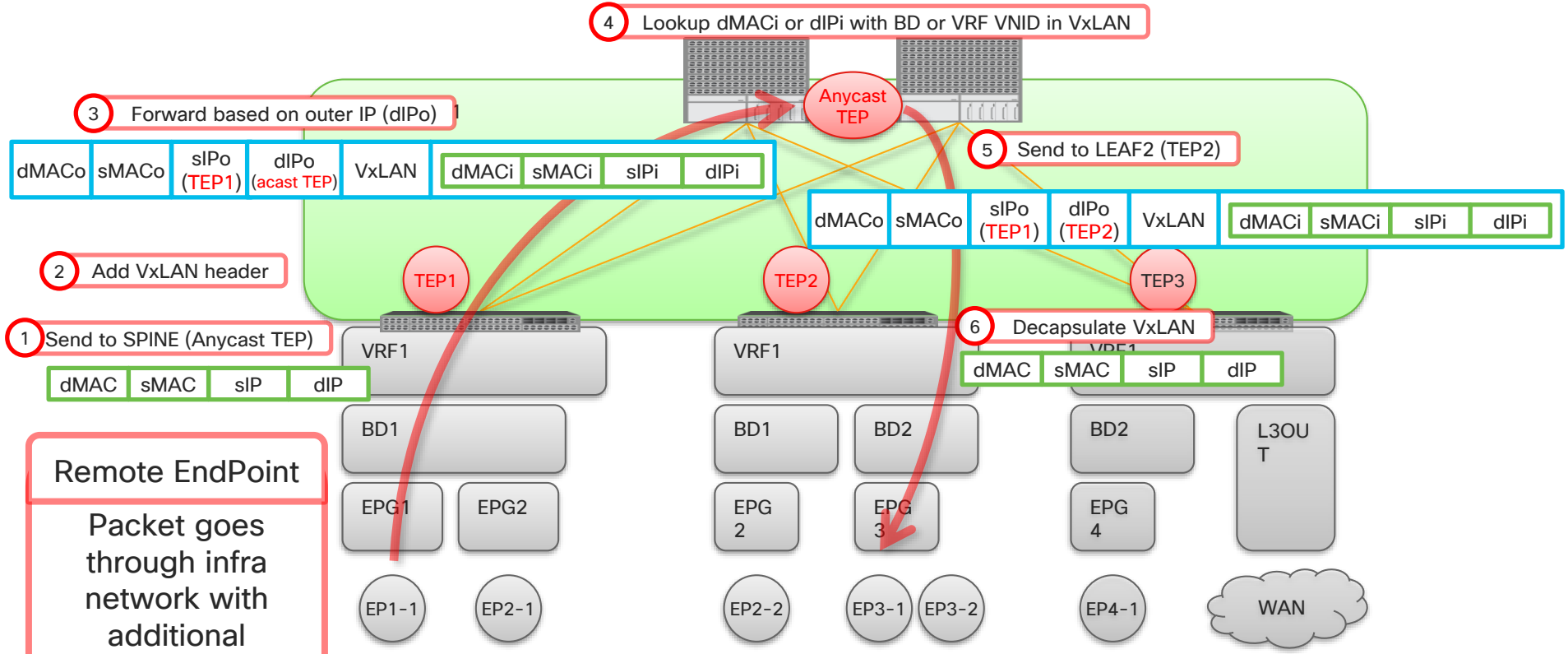


Remote EndPoint

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)

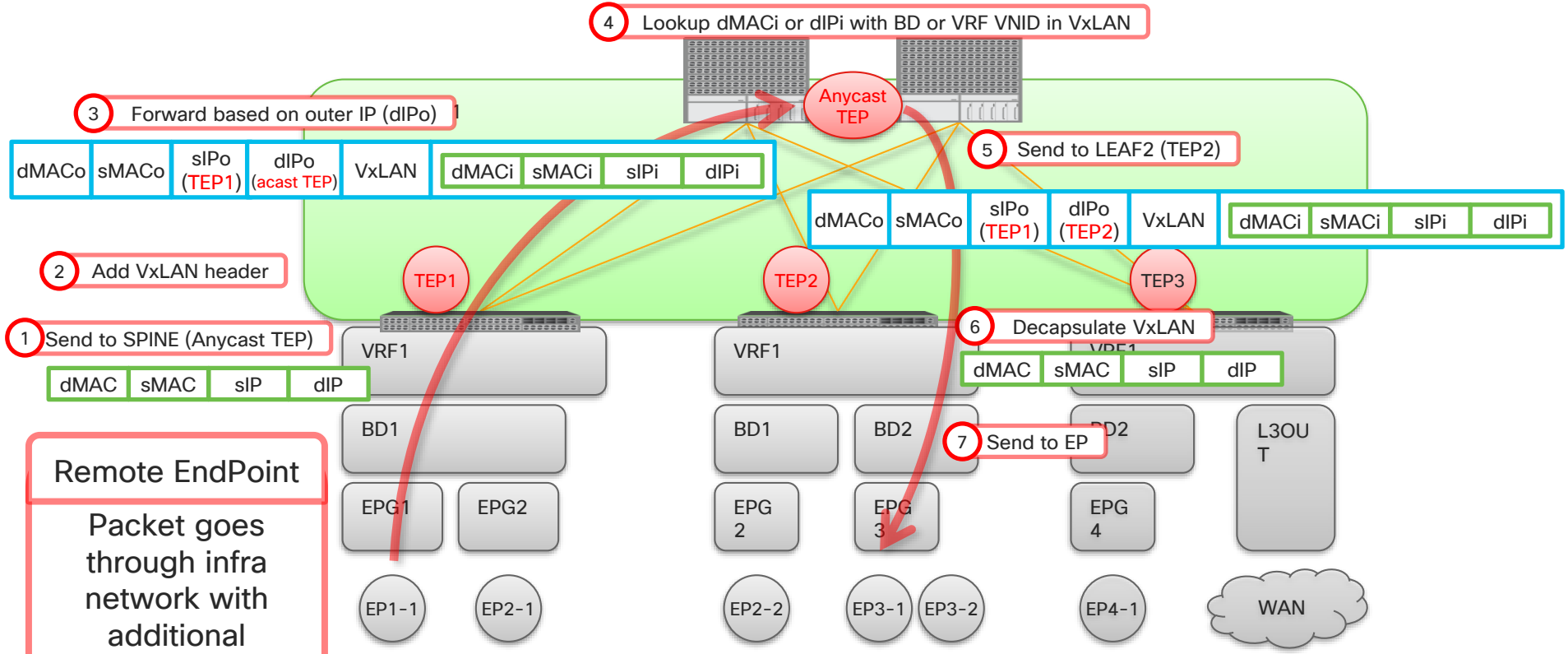


Remote EndPoint

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

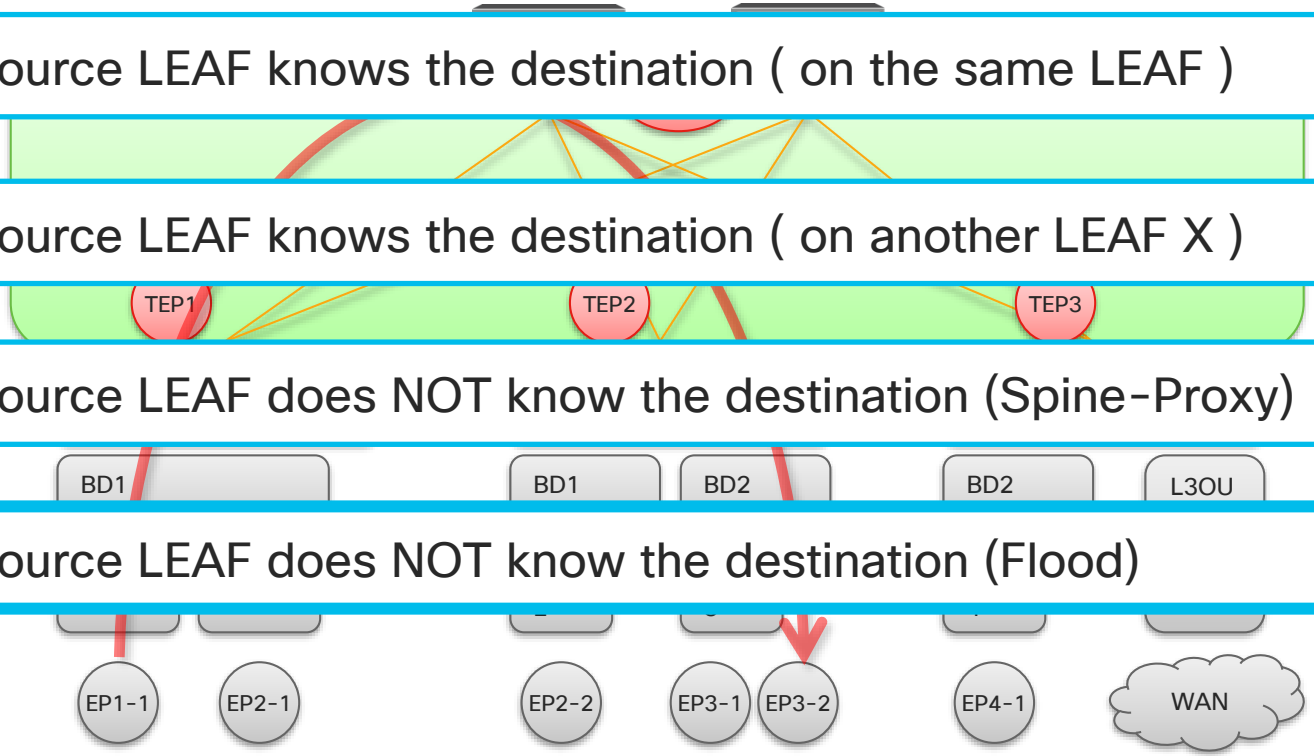
ACI Overlay VxLAN and TEP

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

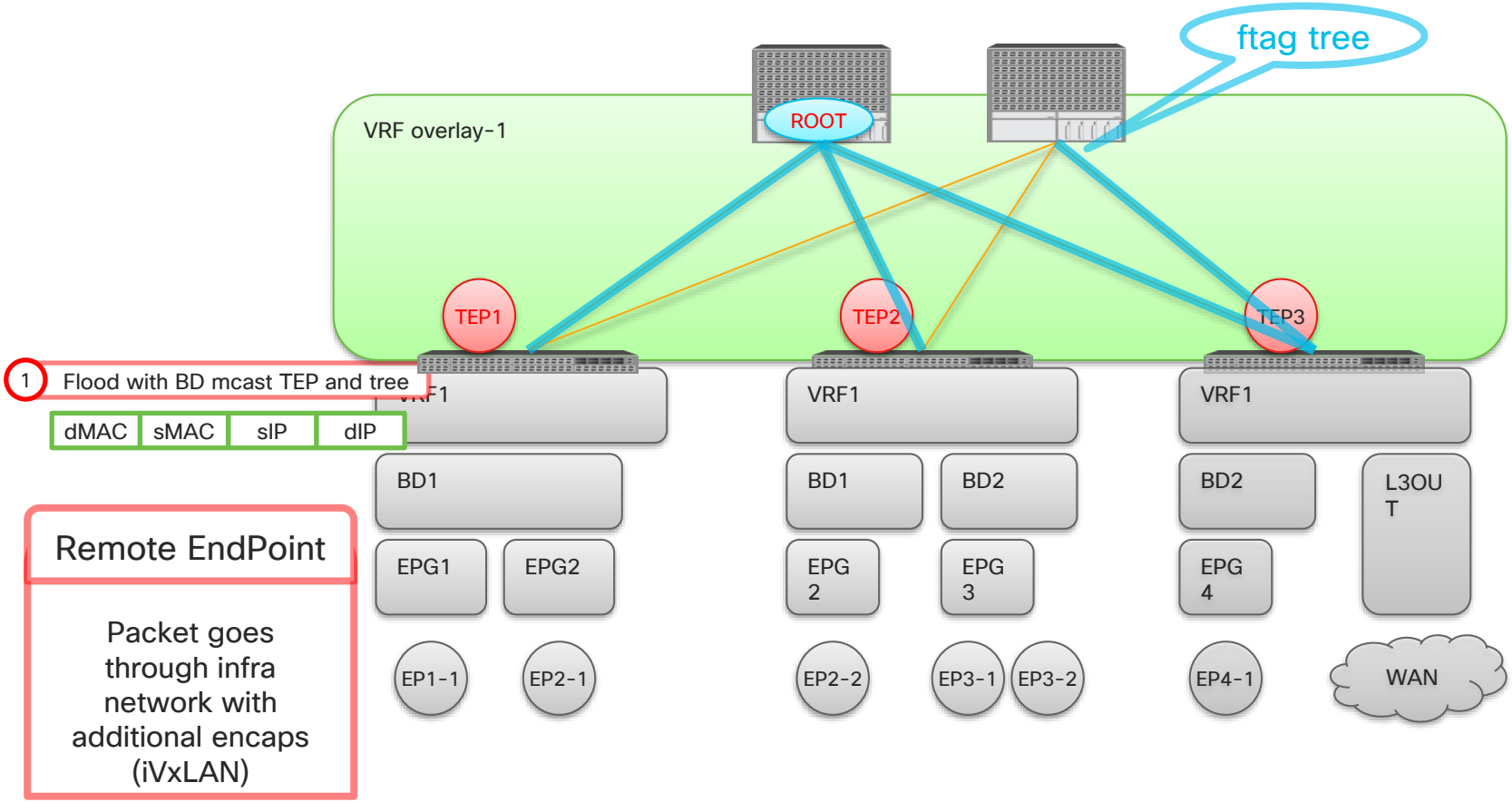
Scenario 2 : source LEAF knows the destination (on another LEAF X)

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

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



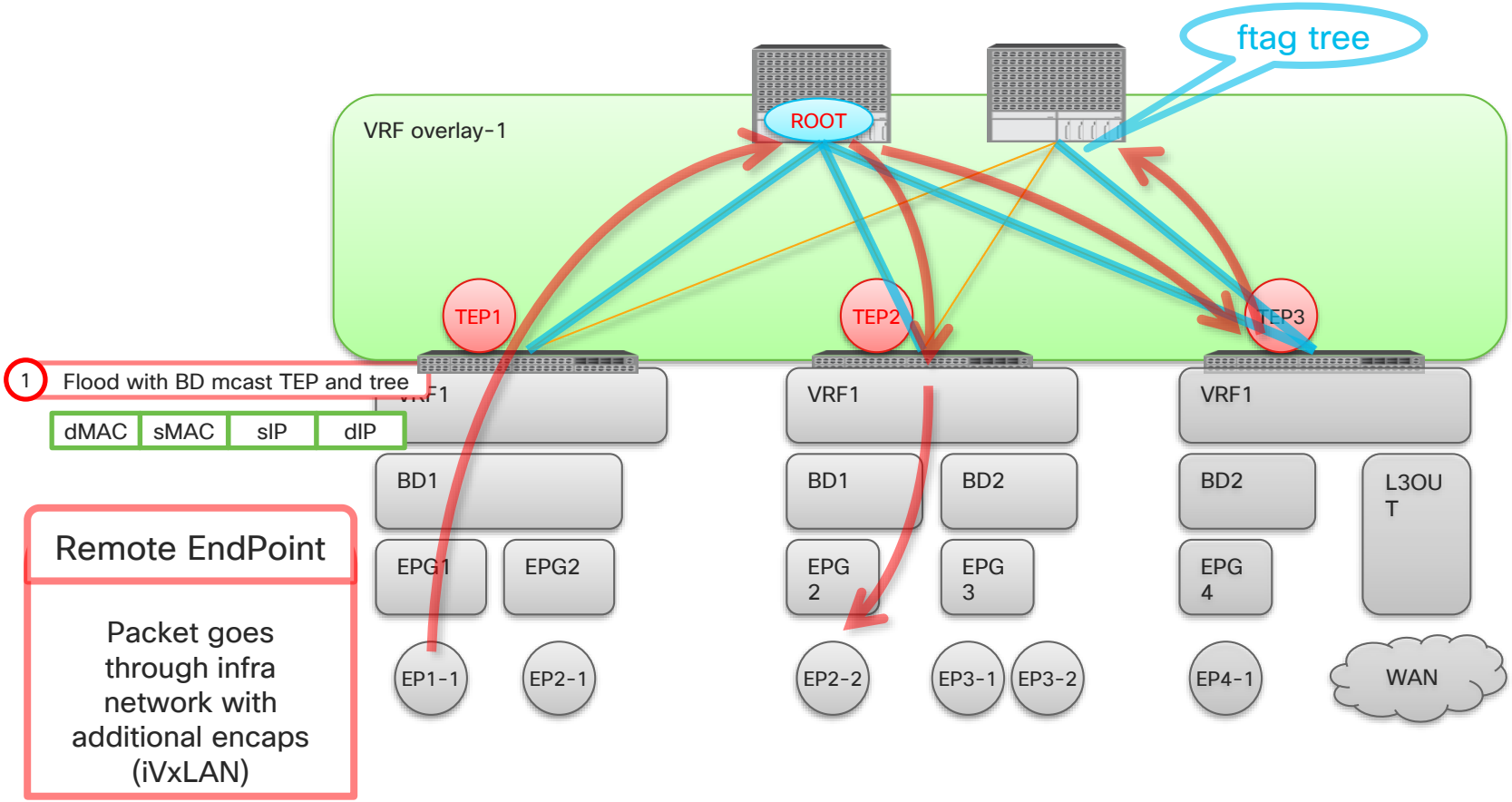
Source LEAF does NOT know the destination (Flood)



Remote EndPoint

Packet goes through infra network with additional encaps (iVxLAN)

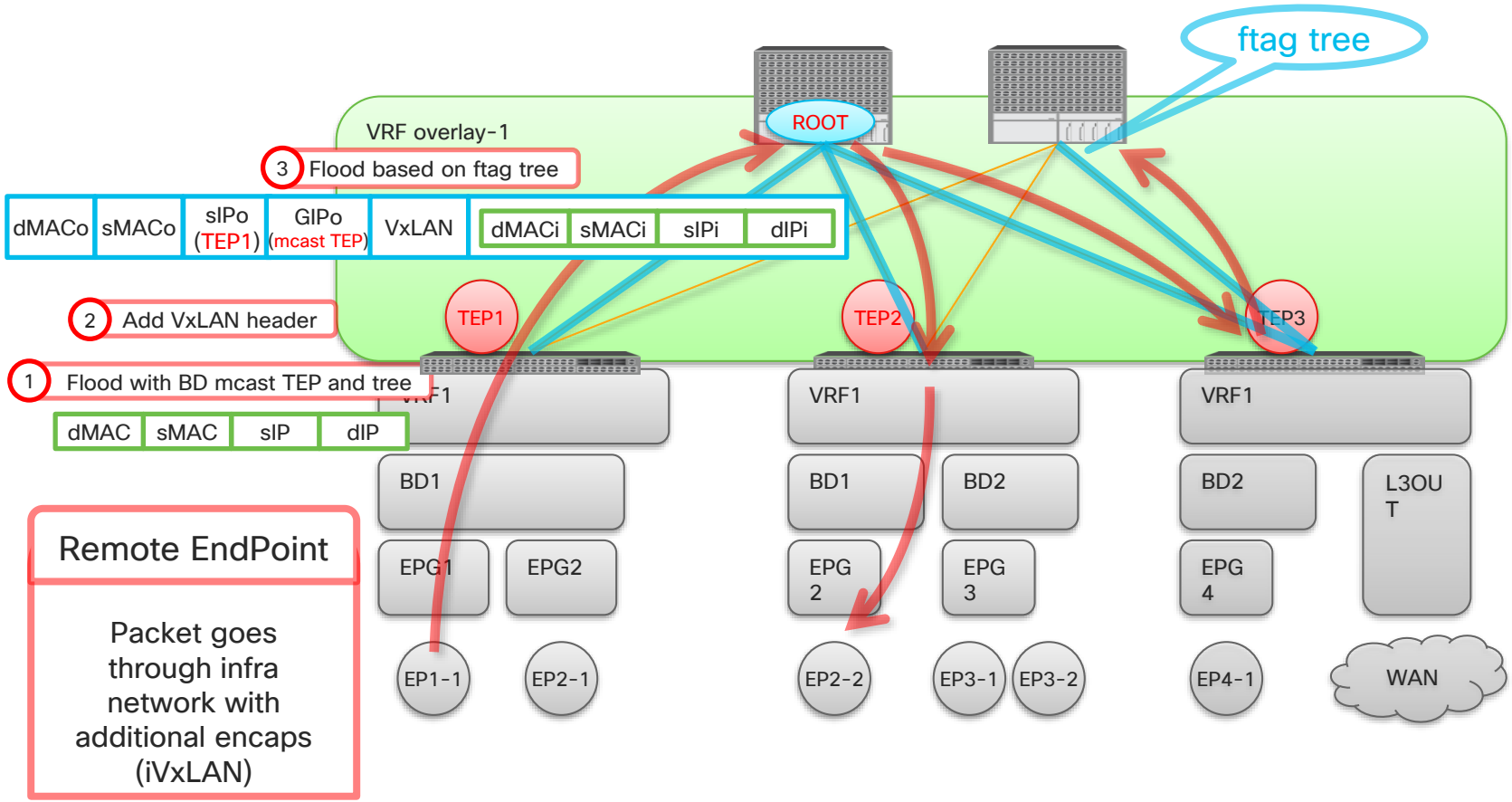
Source LEAF does NOT know the destination (Flood)



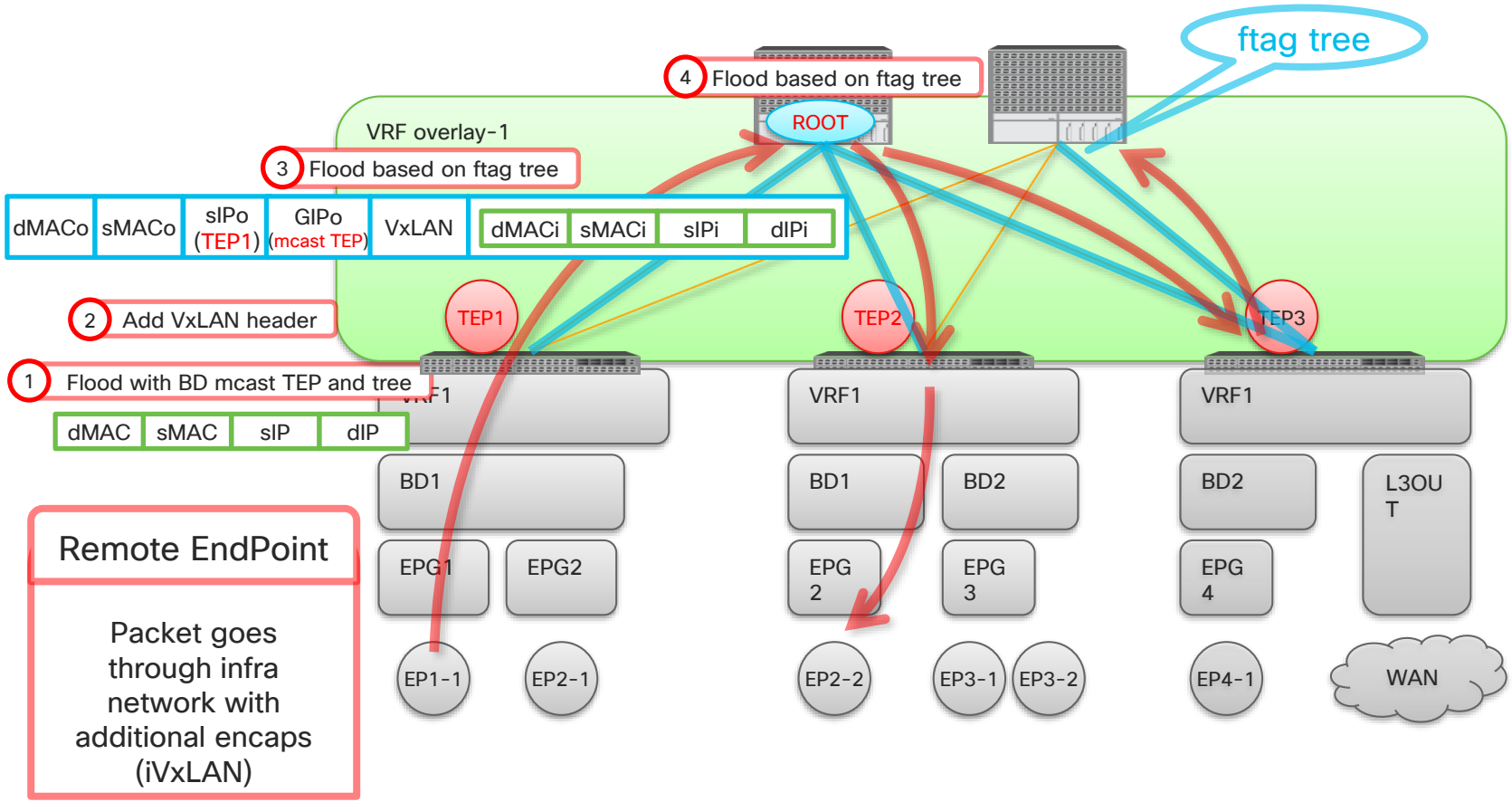
Remote EndPoint

Packet goes through infra network with additional encaps (iVxLAN)

Source LEAF does NOT know the destination (Flood)



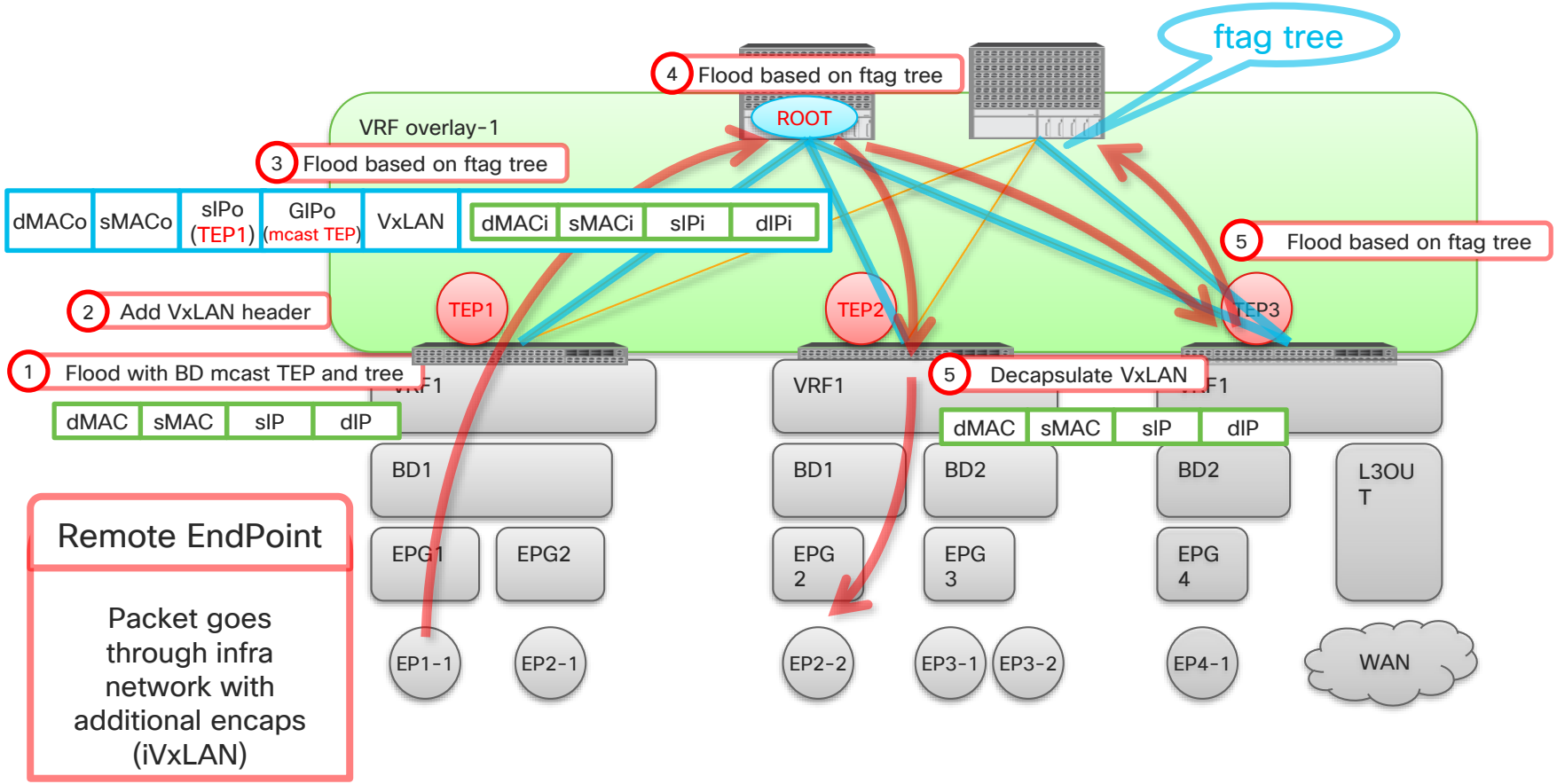
Source LEAF does NOT know the destination (Flood)



Remote EndPoint

Packet goes through infra network with additional encaps (iVxLAN)

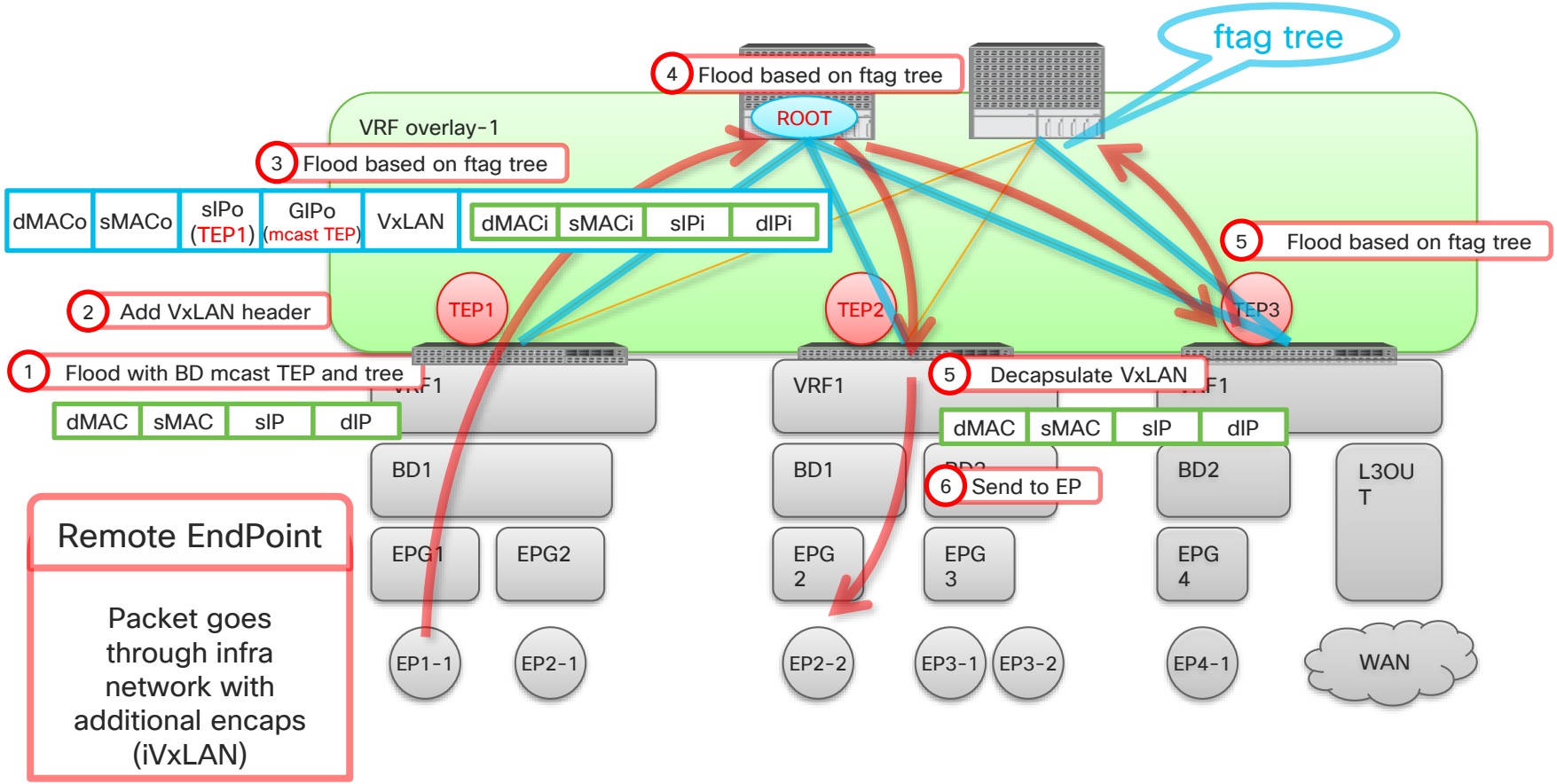
Source LEAF does NOT know the destination (Flood)



Remote EndPoint

Packet goes through infra network with additional encaps (iVxLAN)

Source LEAF does NOT know the destination (Flood)



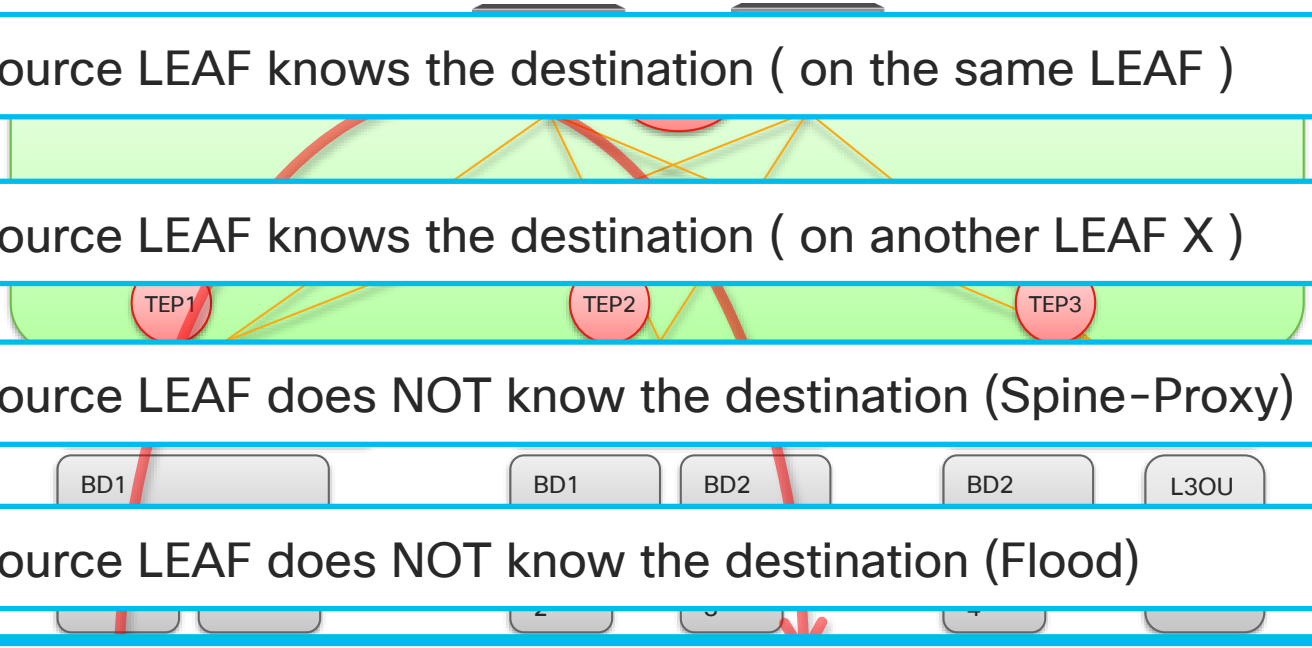
Remote EndPoint

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

ACI Forwarding Component 1

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

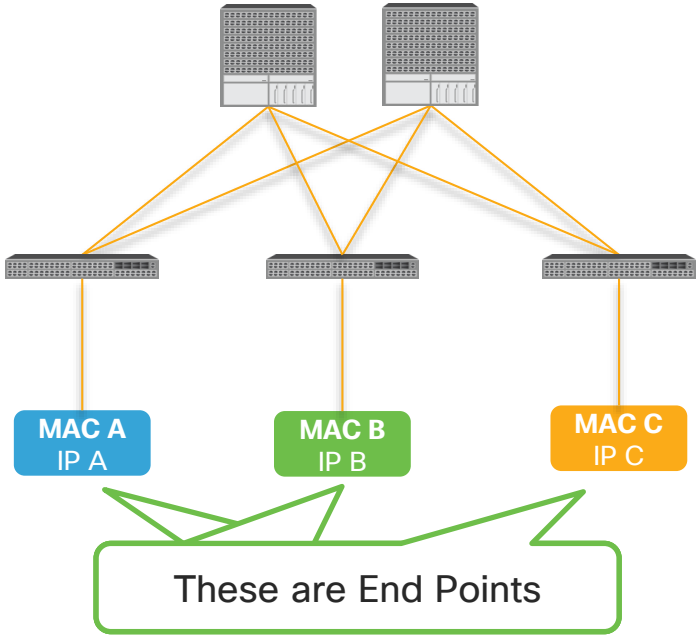
End Point (EP)

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)



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?

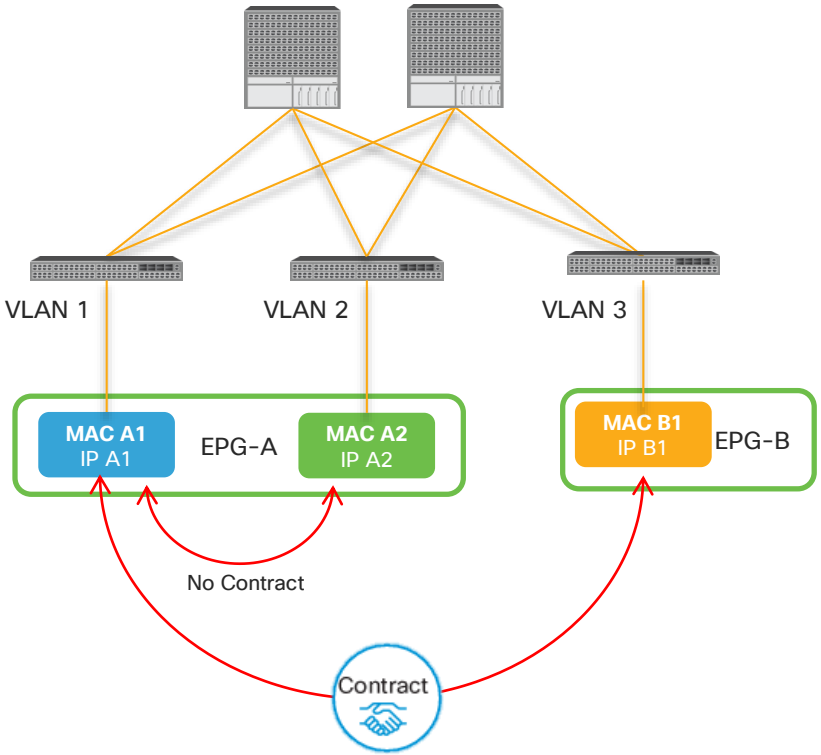
- **To implement traffic filtering**
  - Traffic within the same EPG doesn't get blocked
  - Traffic across EPGs always blocked without a contract
- A contract is applied between EPGs to allow traffic

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

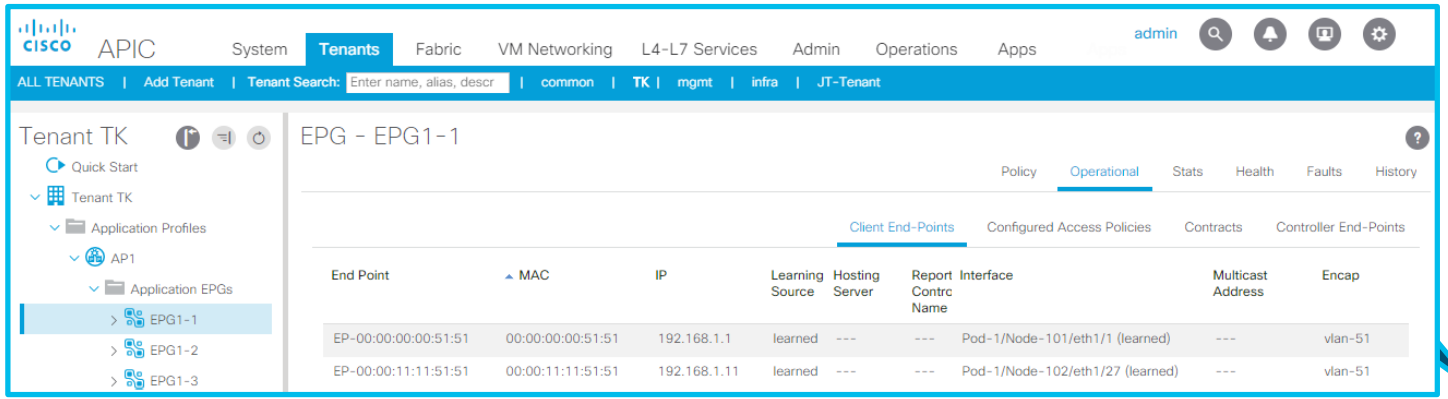
What happens with forwarding?

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



How to check End Points

From APIC GUI (Fabric perspective)

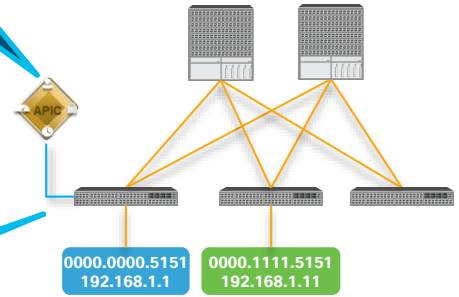


Fabric Wide Visibility

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
```

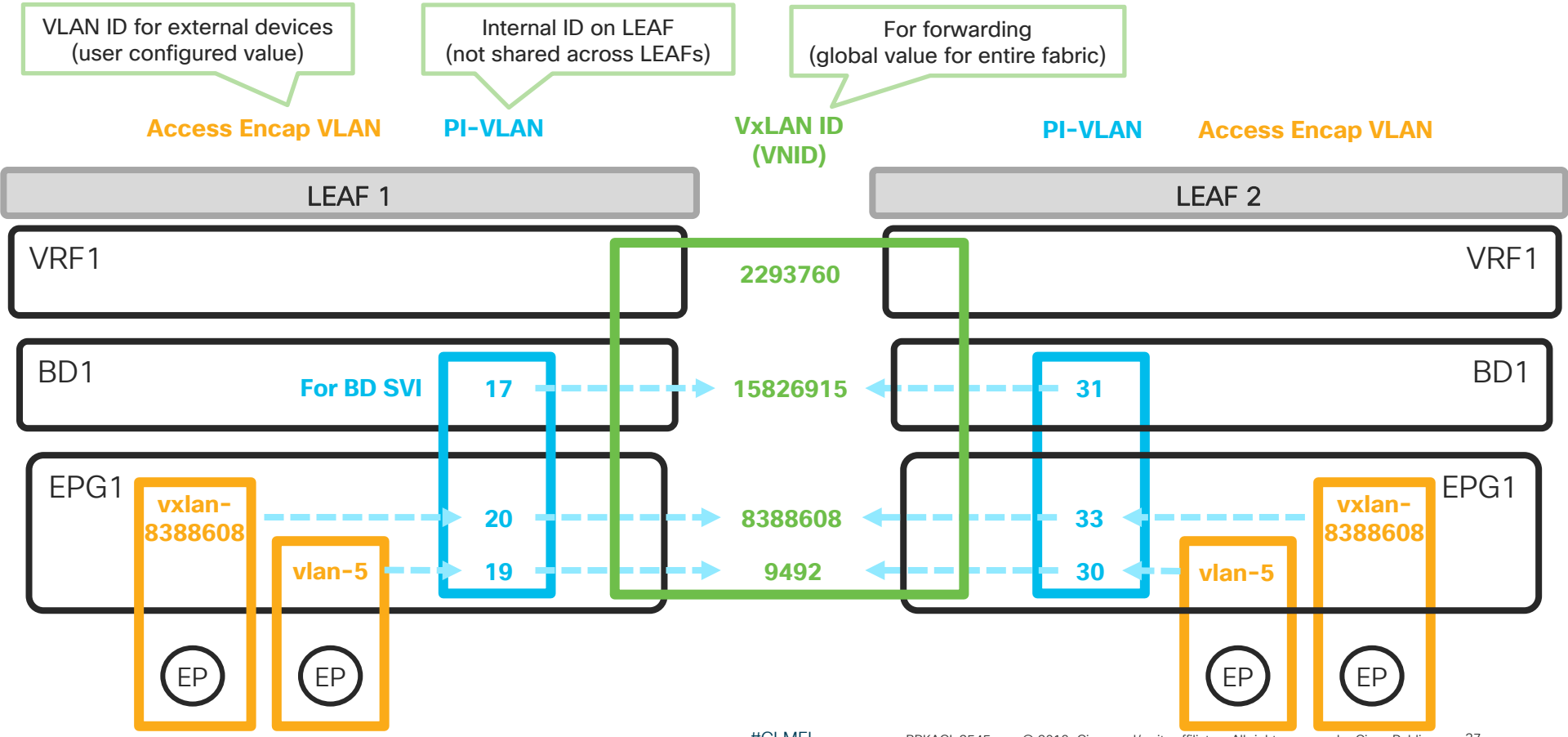


Good for forwarding verification

shows how EPs look from each LEAF

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

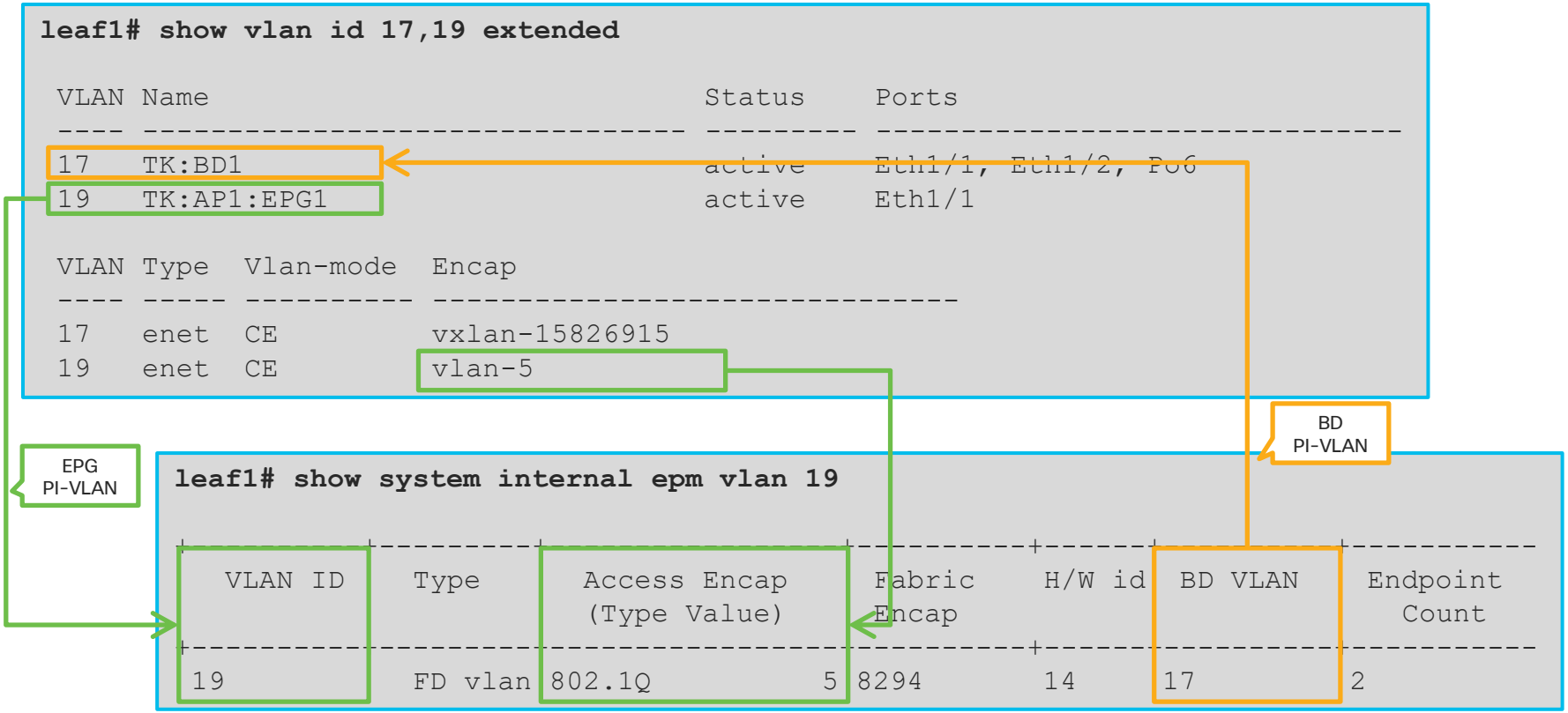
```
leaf1# show vlan id 17,19 extended
 VLAN  Name                                     Status    Ports
 ----  -------------------------------- --------- ----------------------
 17    TK:BD1                                   active    Eth1/1, Eth1/2, Po6
 19    TK:AP1:EPG1                            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



End Point Types

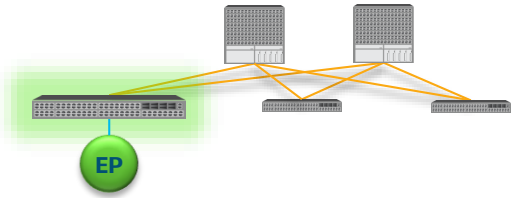
all commands on 

```
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
```

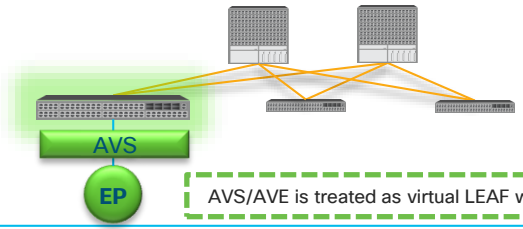


Virtual Local Endpoint (VL)

- 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

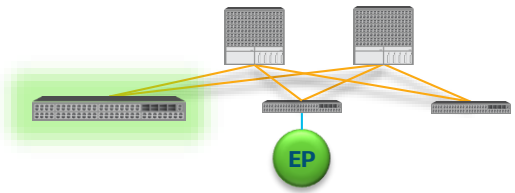
Remote Endpoint (Xr)

- An endpoint on a remote LEAF

BD VNID (not Access Encap VLAN)

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

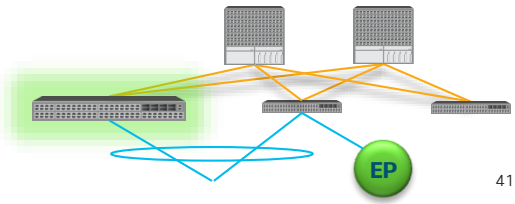
```
fab1-leaf1# show endpoint ip 192.168.0.52
TK:VRF1                                       192.168.0.52              tunnel8
```



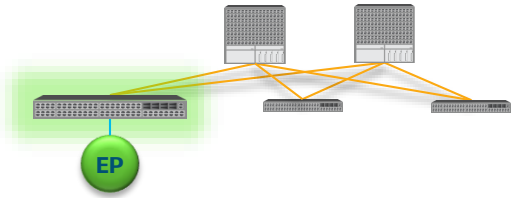
On-Peer Endpoint

- 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)

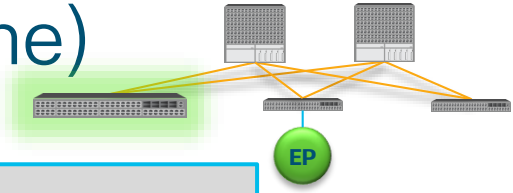
  or

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

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)

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

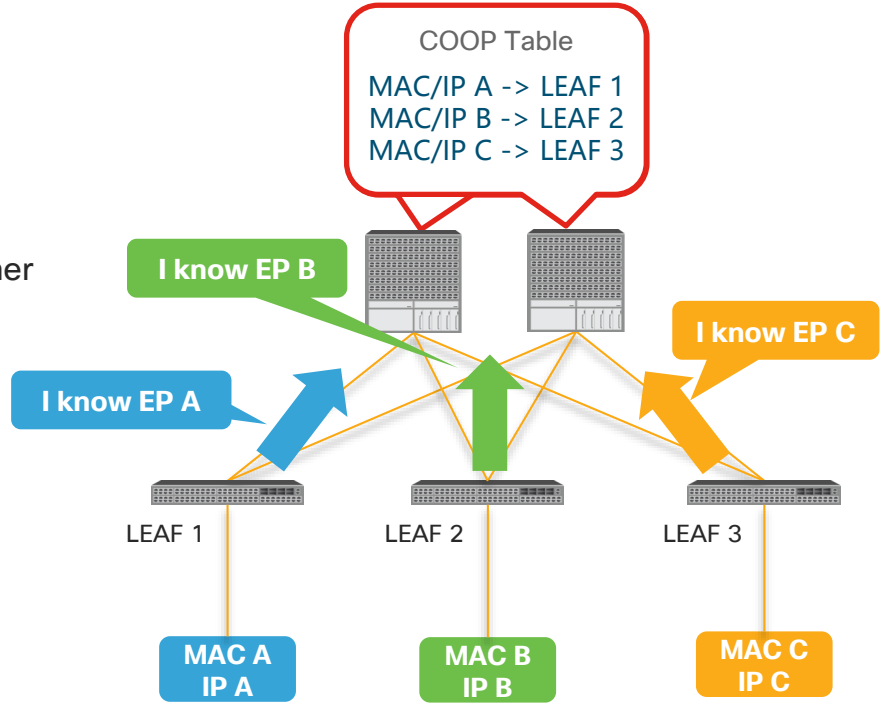
What is the purpose of COOP?

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.



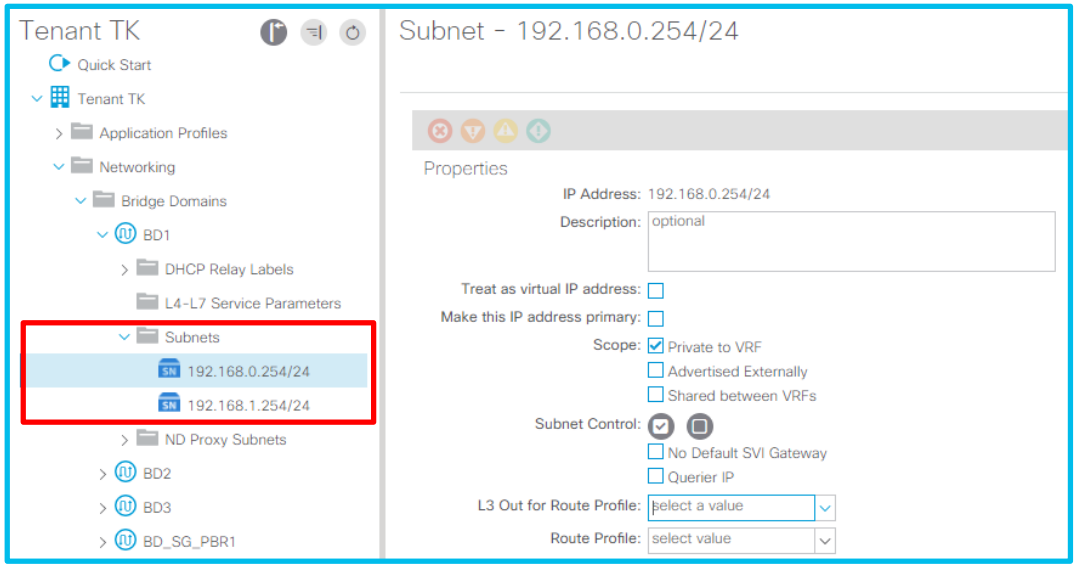
Agenda

- 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)




```
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

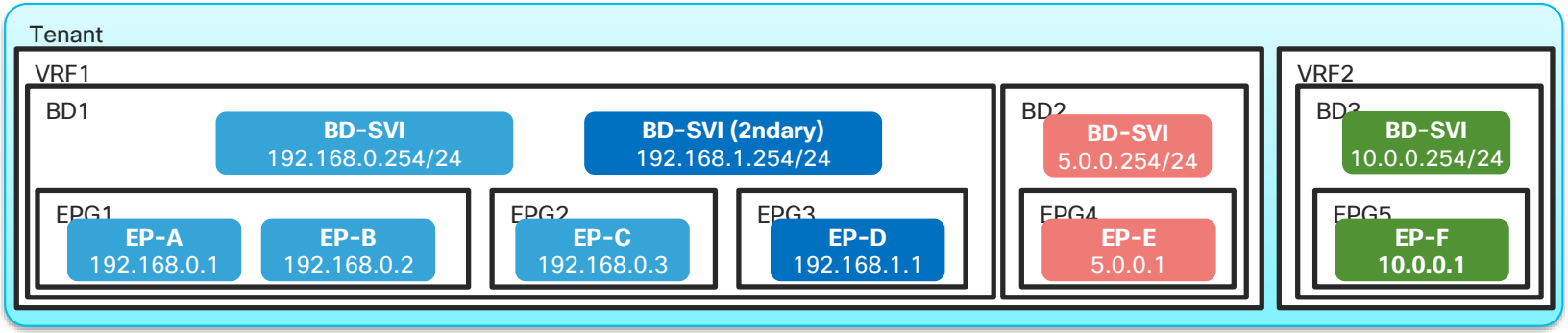
What is pervasive GW for?

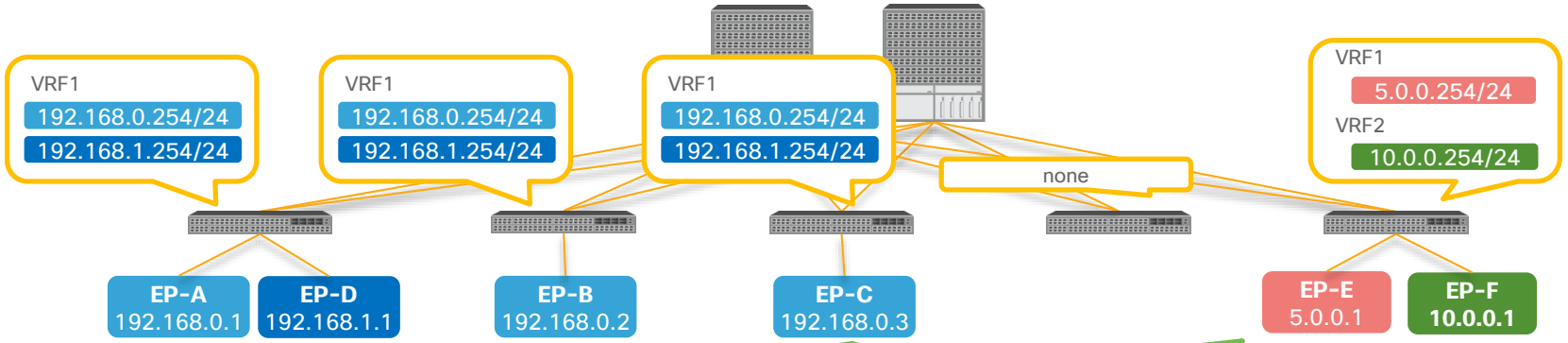
- To be a default GW for EPs in the Fabric
  - All EPs can have consistent gateway IP address one hop away
- To represent subnets(IP ranges) for a BD
  - ACI knows which BD may have potential hidden/silent EPs

How is pervasive GW deployed?

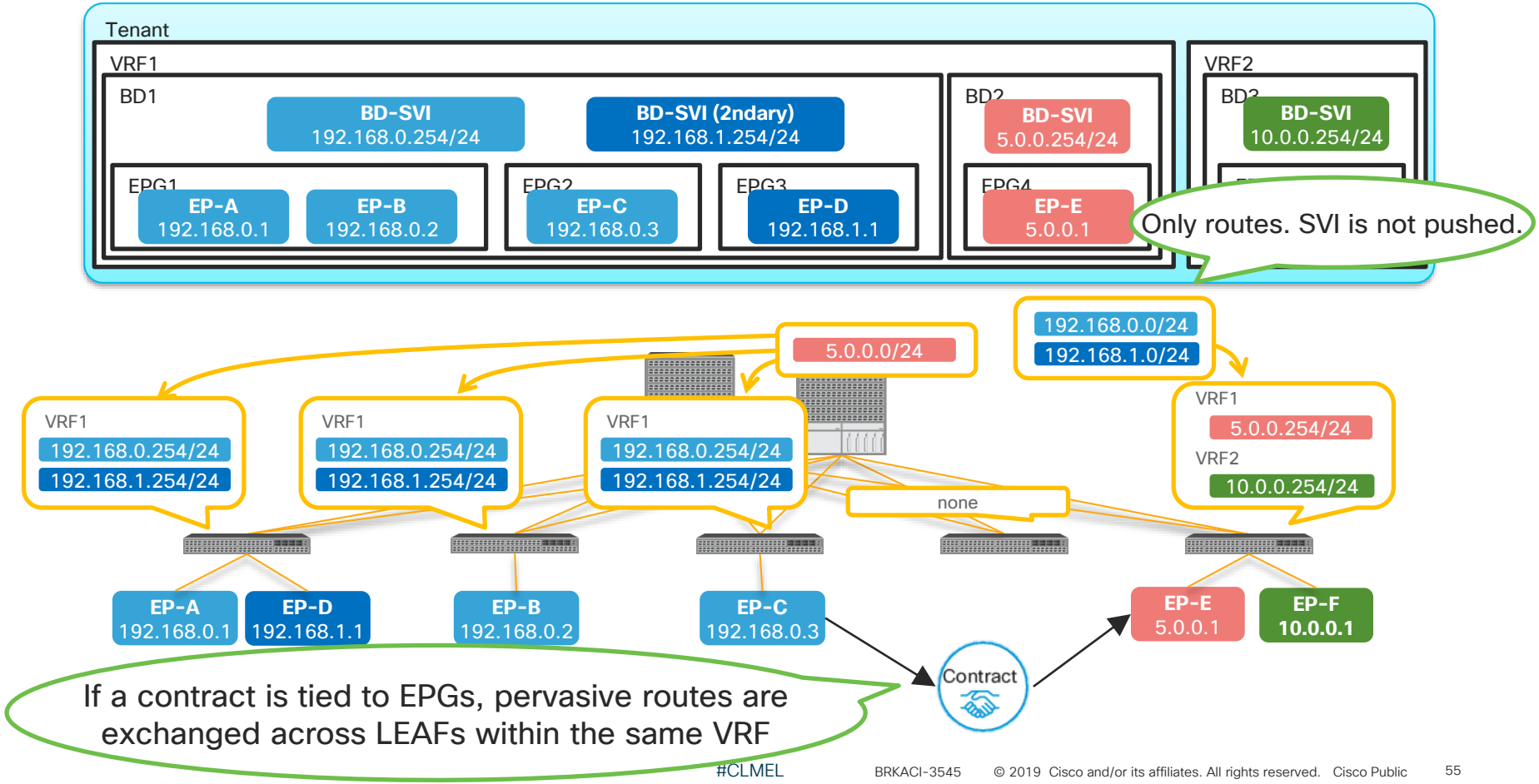
- Installed as an SVI on LEAFs
  - PI-VLAN for BD is used to represent a pervasive GW SVI
  - A pervasive SVI has secondary IP when multiple pervasive GWs are configured on the same BD
    - User can choose a primary address

Pervasive Gateway(BD SVI) example





Pervasive Gateway(BD SVI) example

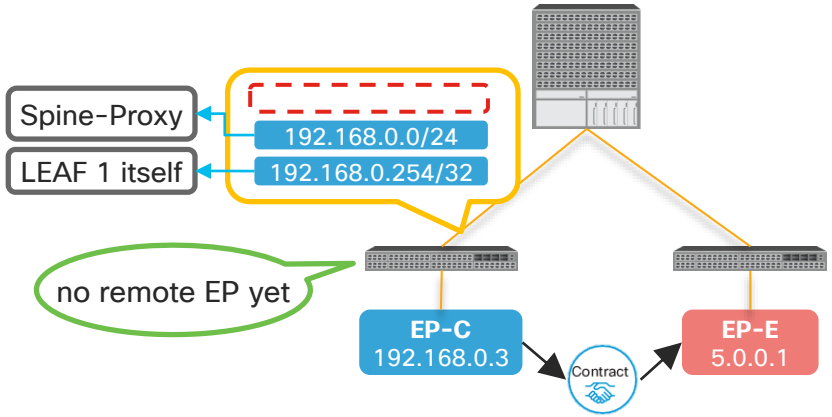


Pervasive Gateway(BD SVI) cont.

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

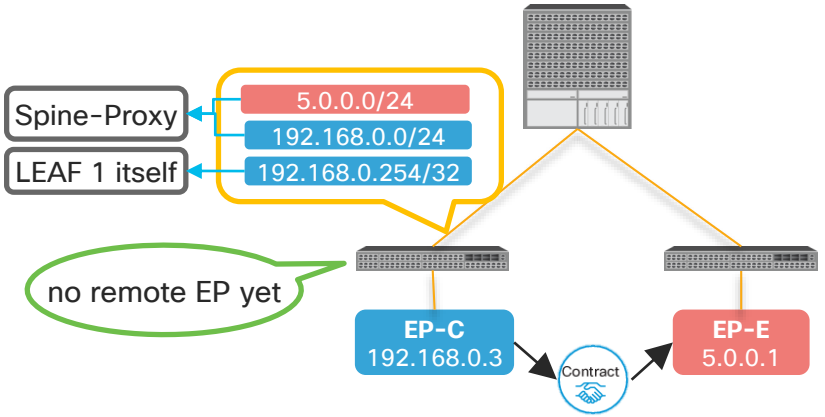
- **Pervasive routes are required for Spine-Proxy**

what if no pervasive route, no remote EP?



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?

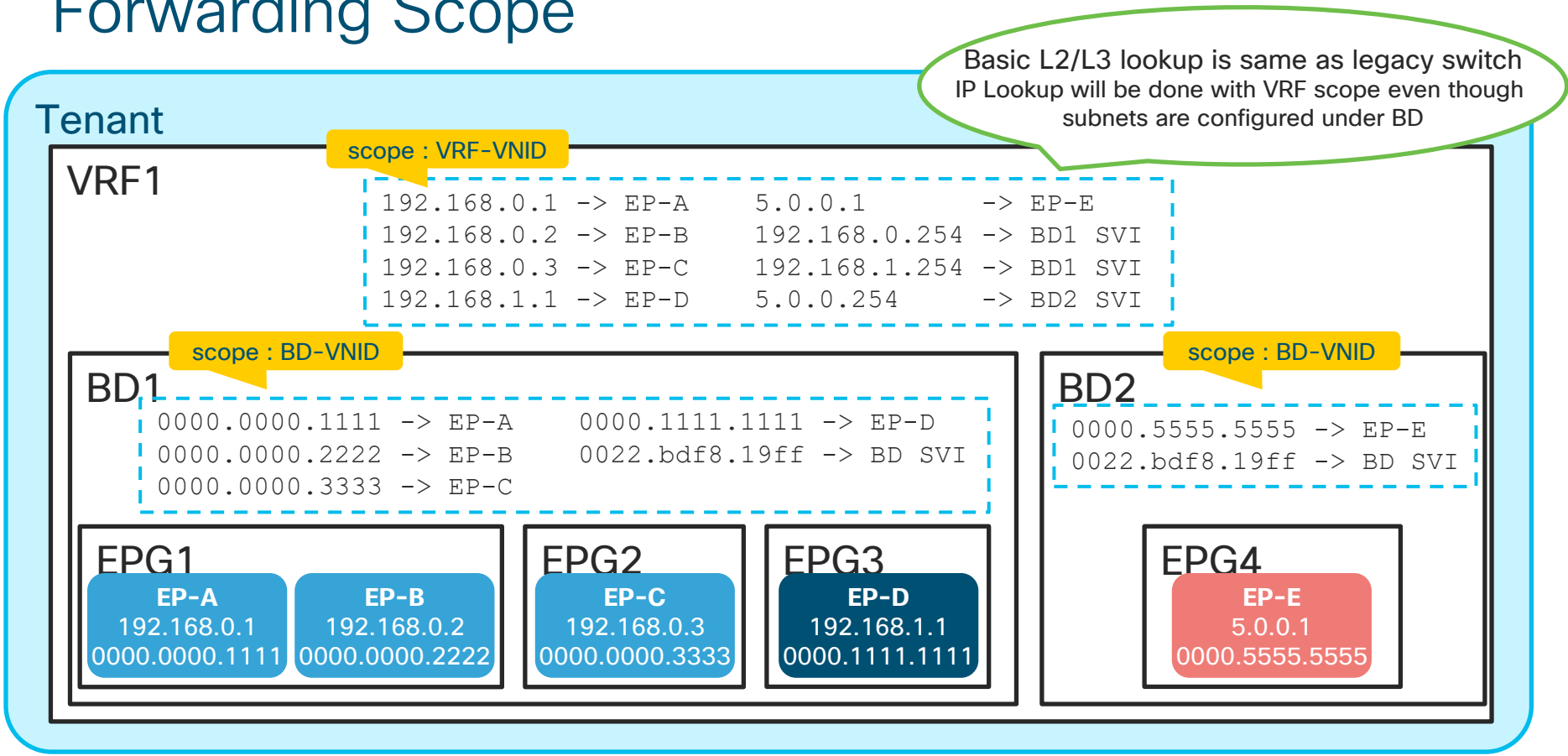


Spine-Proxy for 5.0.0.1
With the contract, ACI knows the LEAF needs to reach out to 5.0.0.0/24

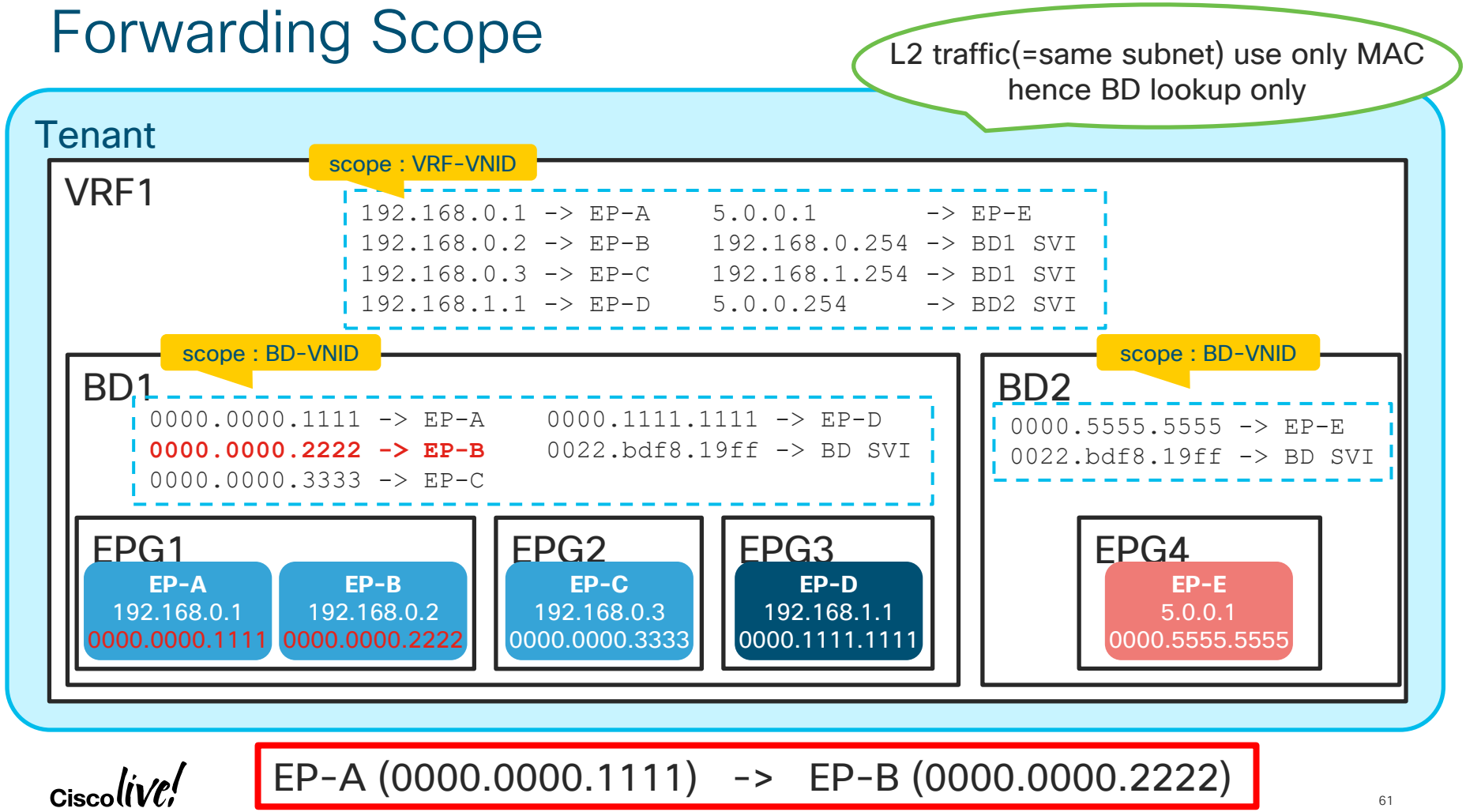
ACI Forwarding Component 2

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

Forwarding Scope

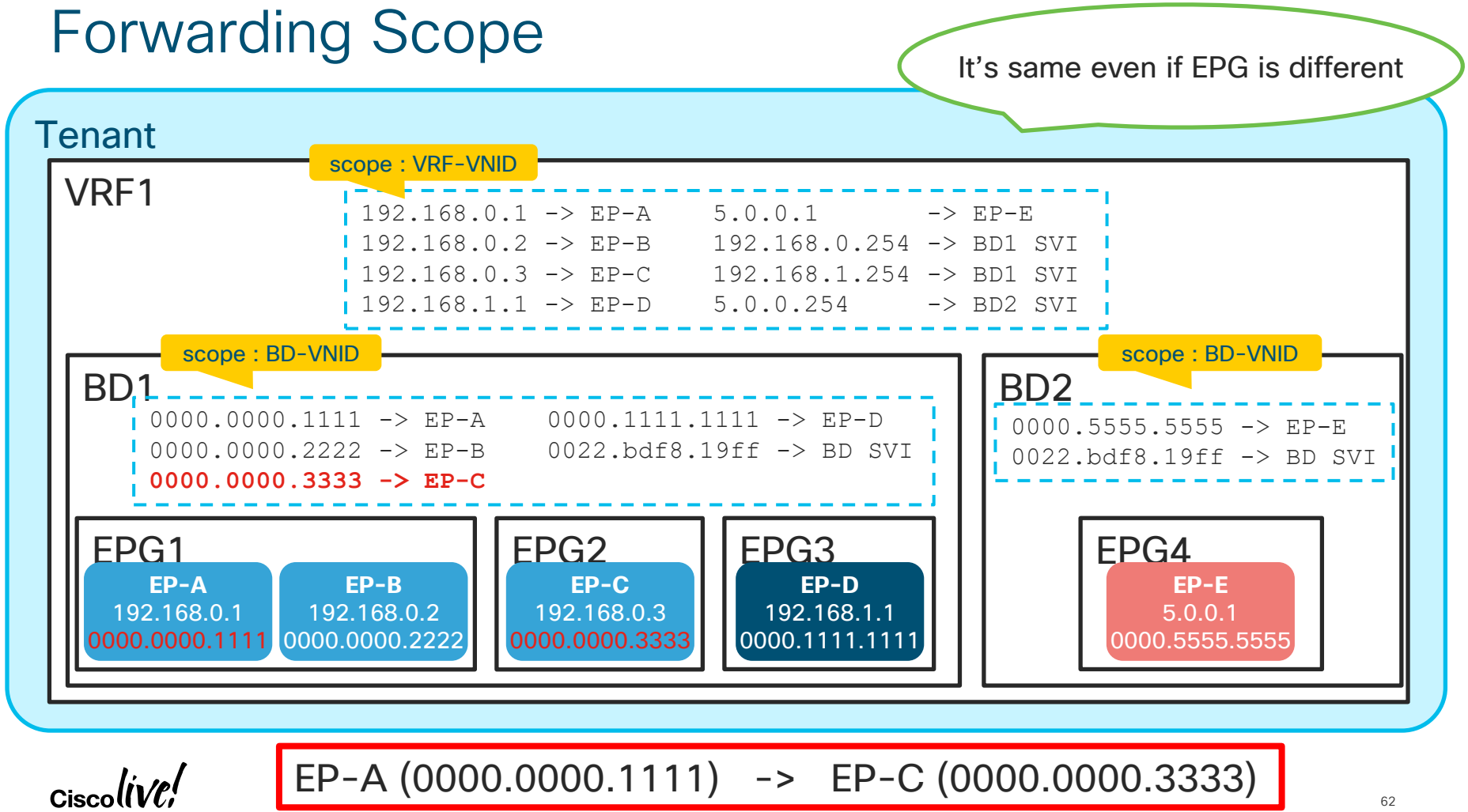


Forwarding Scope



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

Forwarding Scope



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
```

EPG1

EP-A	EP-B
192.168.0.1	192.168.0.2
0000.0000.1111	0000.0000.2222

EPG2

EP-C
192.168.0.3
0000.0000.3333

EPG3

EP-D
192.168.1.1
0000.1111.1111

BD2

scope : BD-VNID

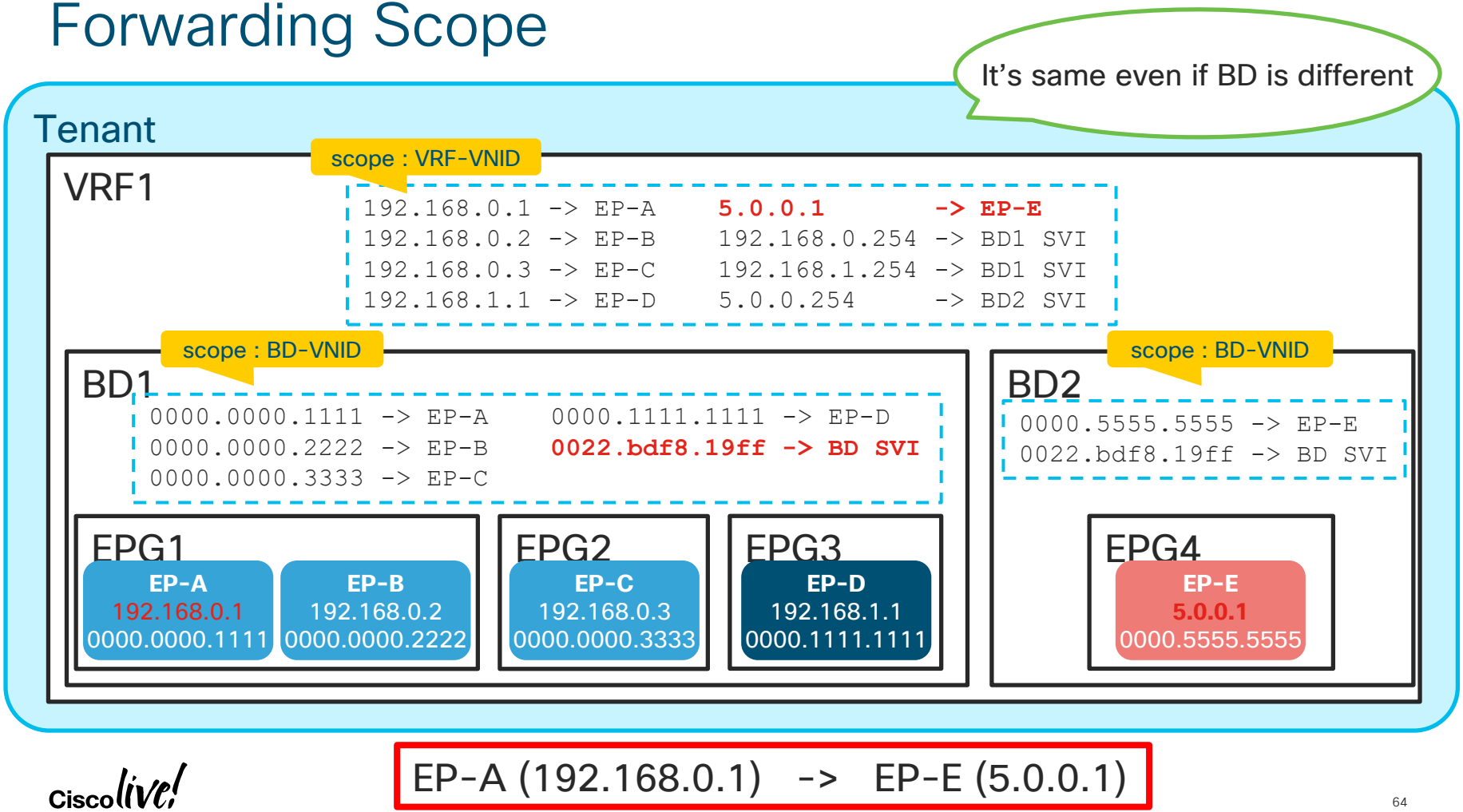
```
0000.5555.5555 -> EP-E
0022.bdf8.19ff -> BD SVI
```

EPG4

EP-E
5.0.0.1
0000.5555.5555

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

Forwarding Scope

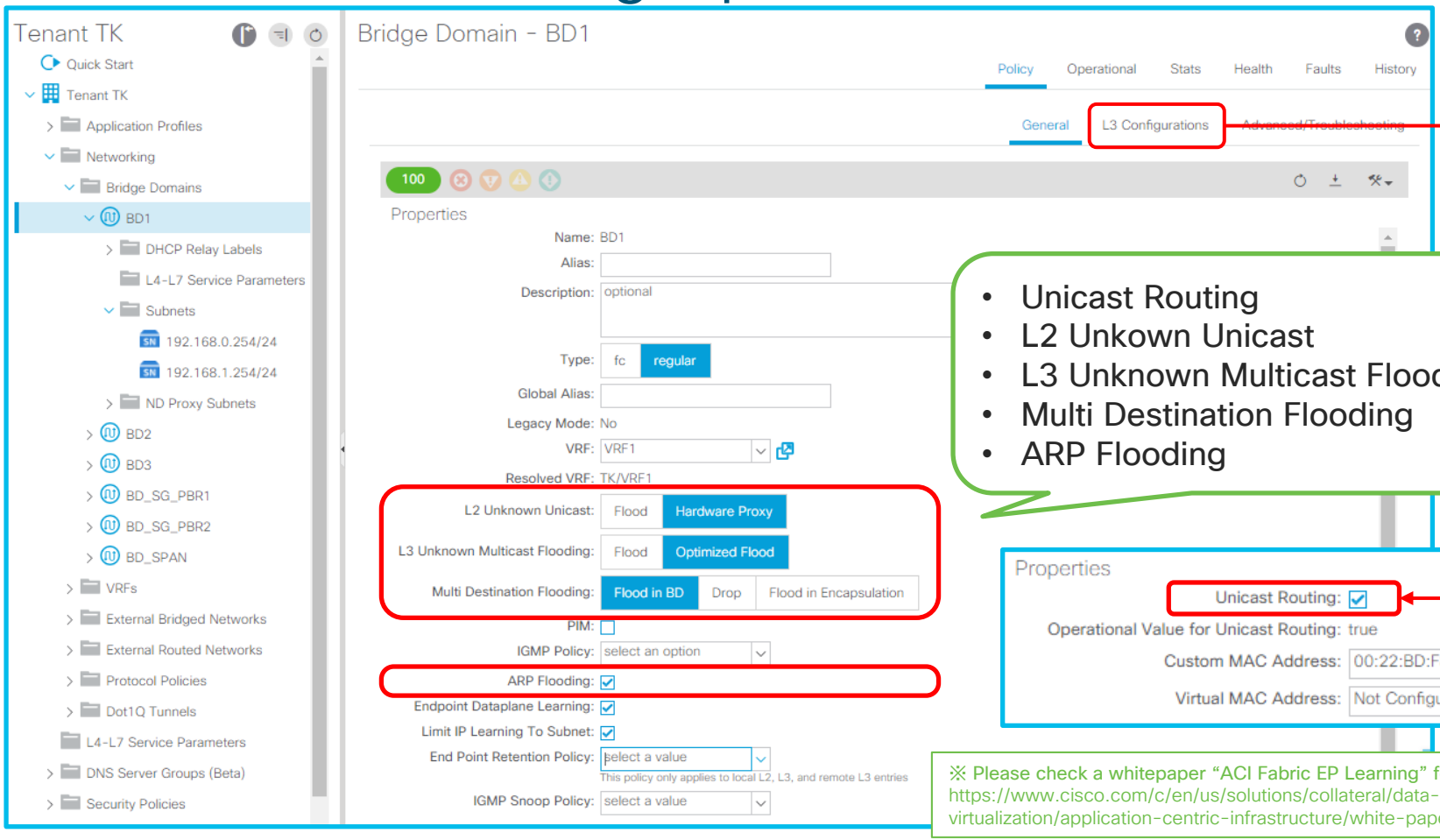


EP-A (192.168.0.1) -> EP-E (5.0.0.1)

ACI Forwarding Component 2

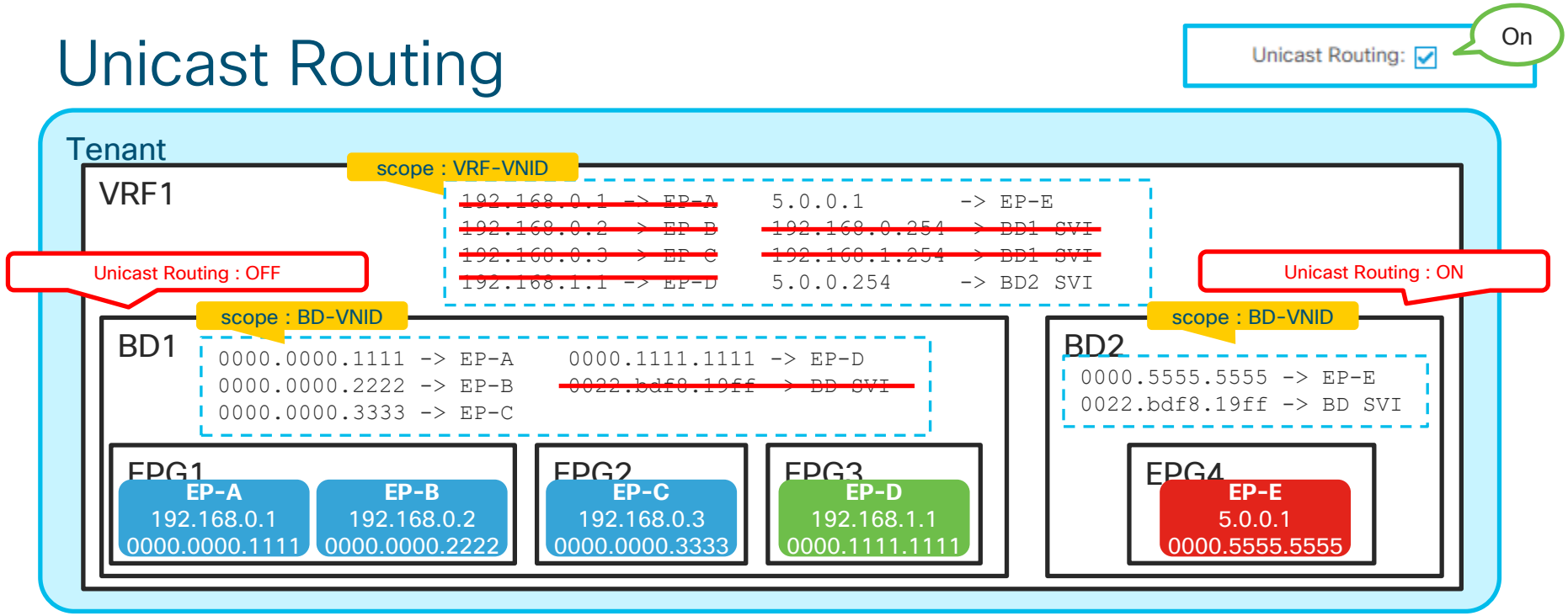
- Pervasive Gateway (BD SVI)
- Forwarding Scope (VRF or BD)
- **Forwarding mode in BD**

ACI BD Forwarding Option



※ 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



If Unicast Routing is disabled, (BD1 in above)

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

  => Only L2 Forwarding is available

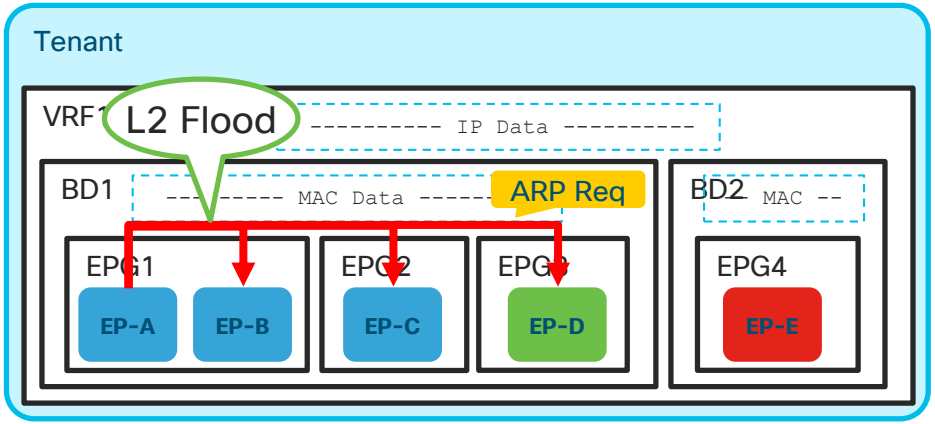
In above example :

- 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: ☑ On

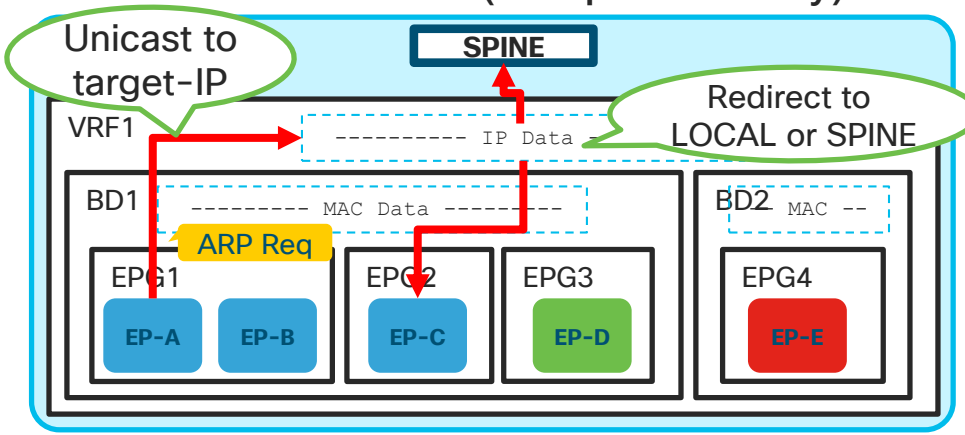
- ARP Flood On



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 will forward it to Leaf on which DstIP resides

Spine-Proxy

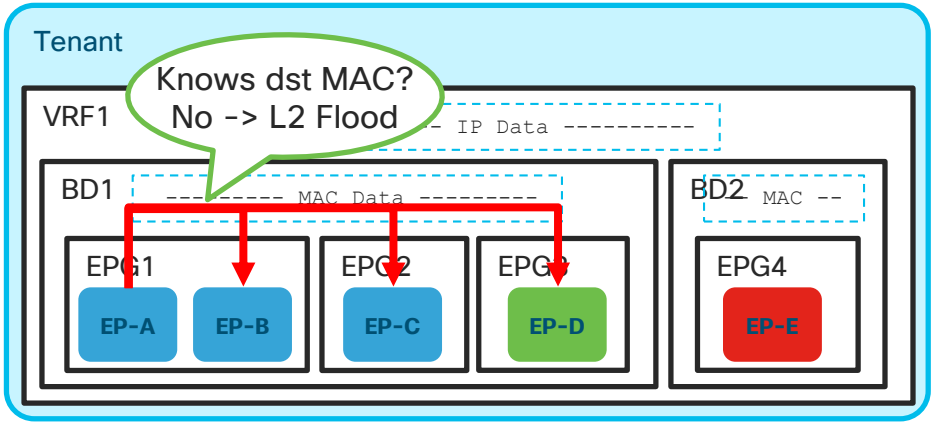
- If IP is not learned even on Spine,
  - Drop and ARP Glean (only within BD)

※ 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**

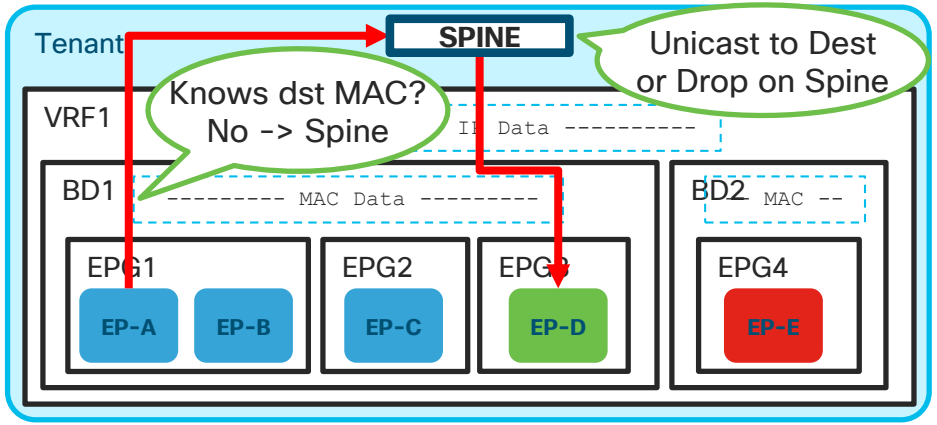
- Flood



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

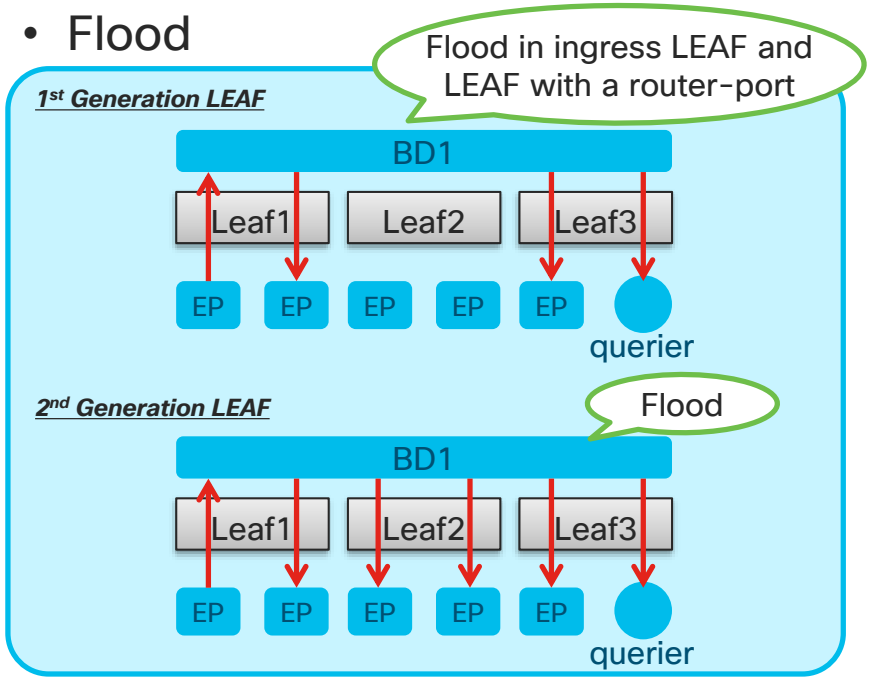
- If DST-MAC is learned on Spine,
  - ➢ Spine forwards it directly to dest Leaf
- If DST-MAC is not learned even on Spine
  - ➢ Drop

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

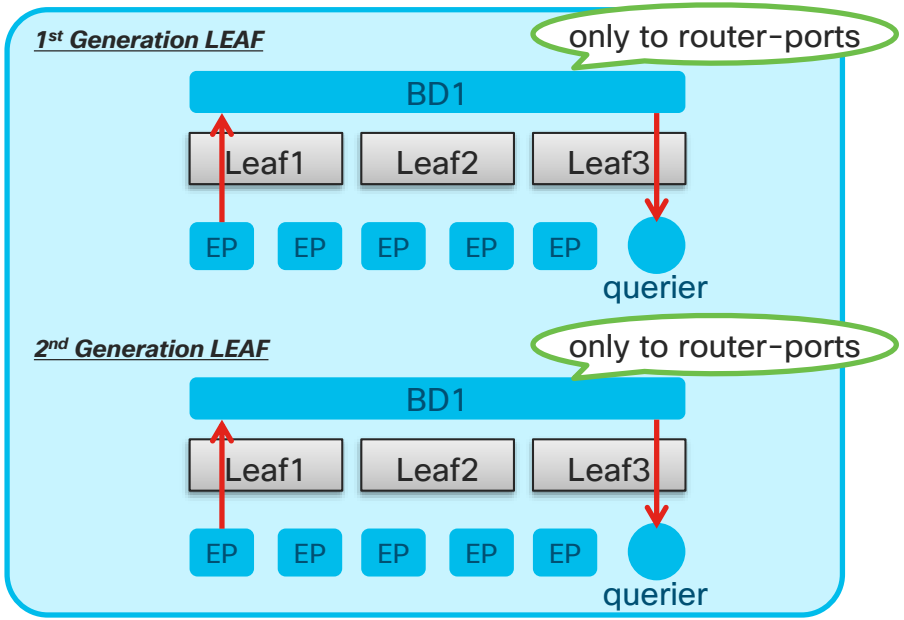
L3 Unknown Multicast Flooding

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

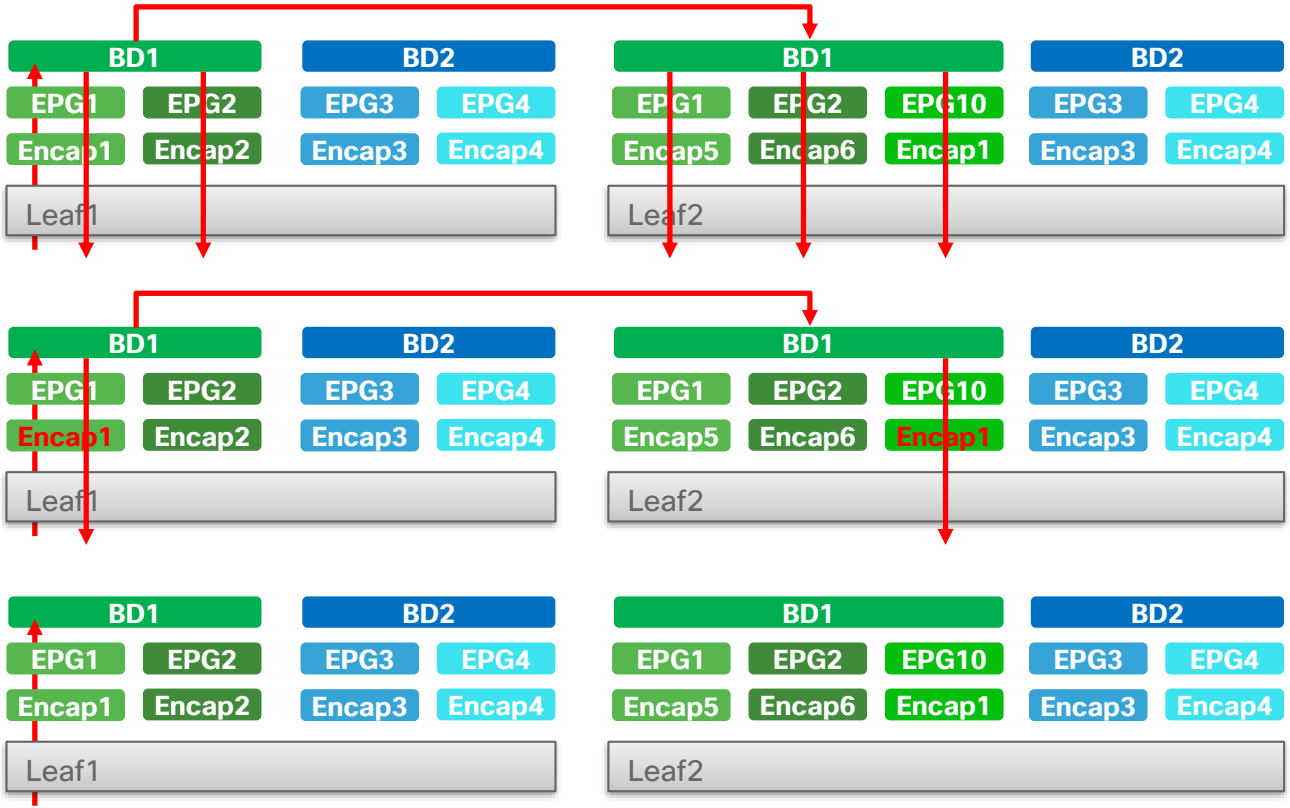
This mode does not apply to OSPF/OSPFv6, BGP, EIGRP, CDP, LACP, LLDP, ISIS, IGMP, PIM, ST-BPDU, ARP/GARP, RARP, ND

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

Behavior change from 3.1

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

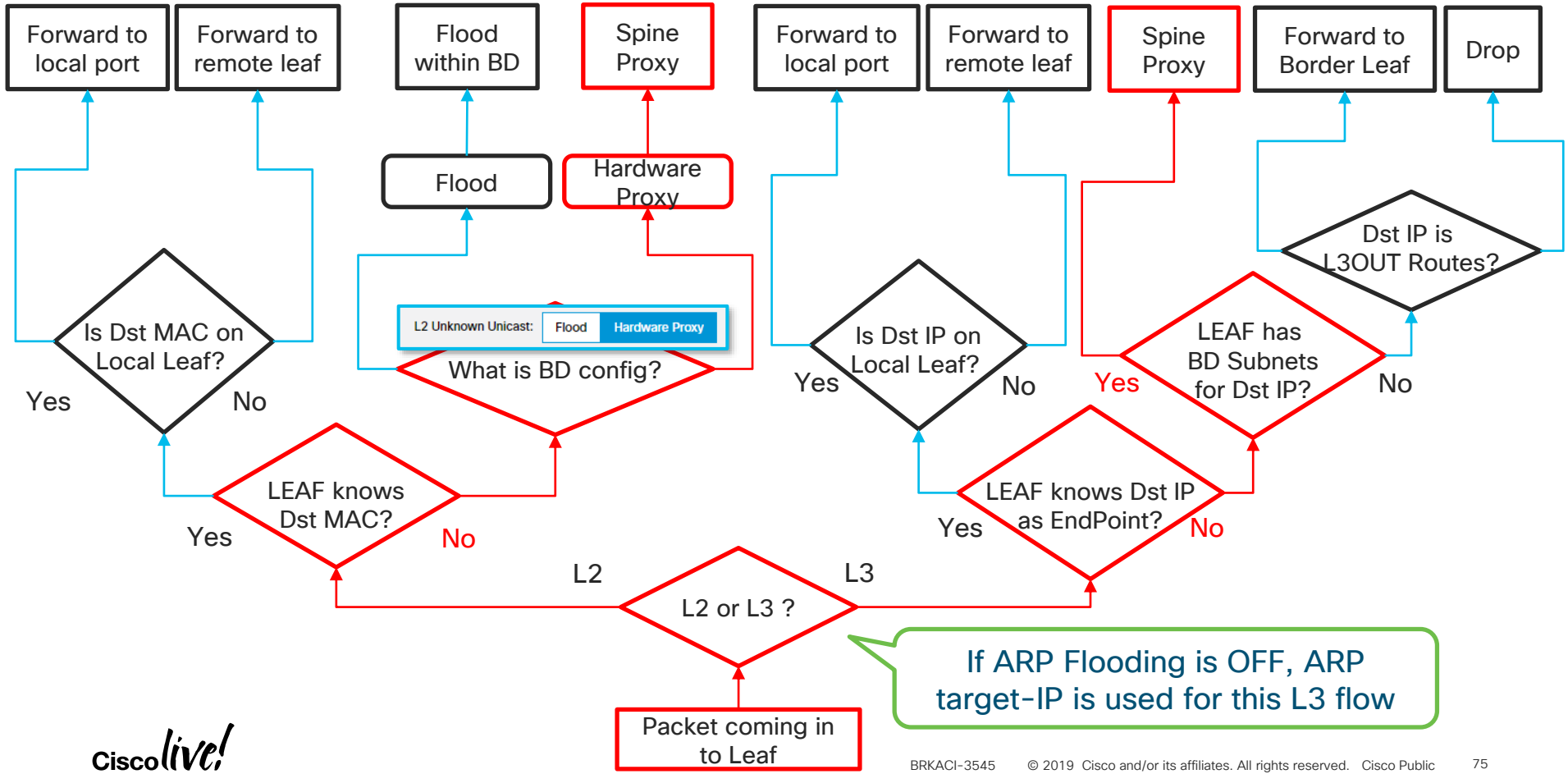
- **Drop**
  No Flood. Just drop.



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

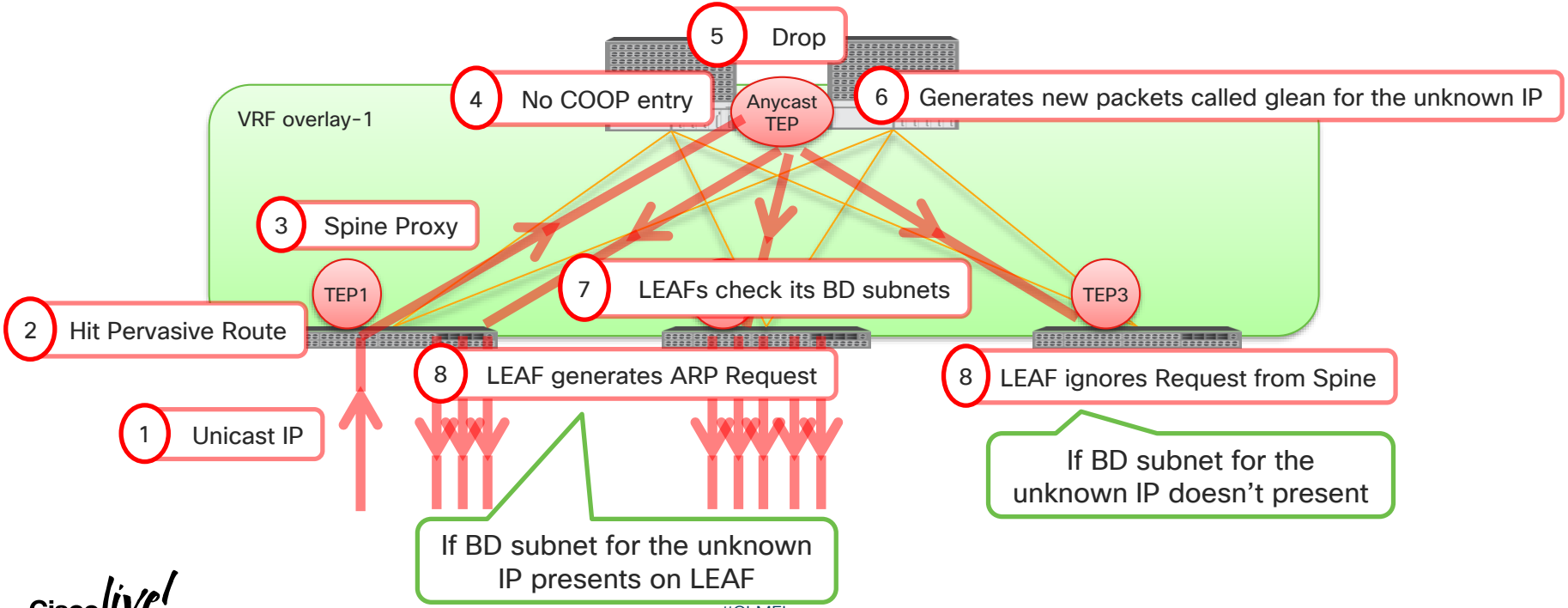
Spine Proxy Summary



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.

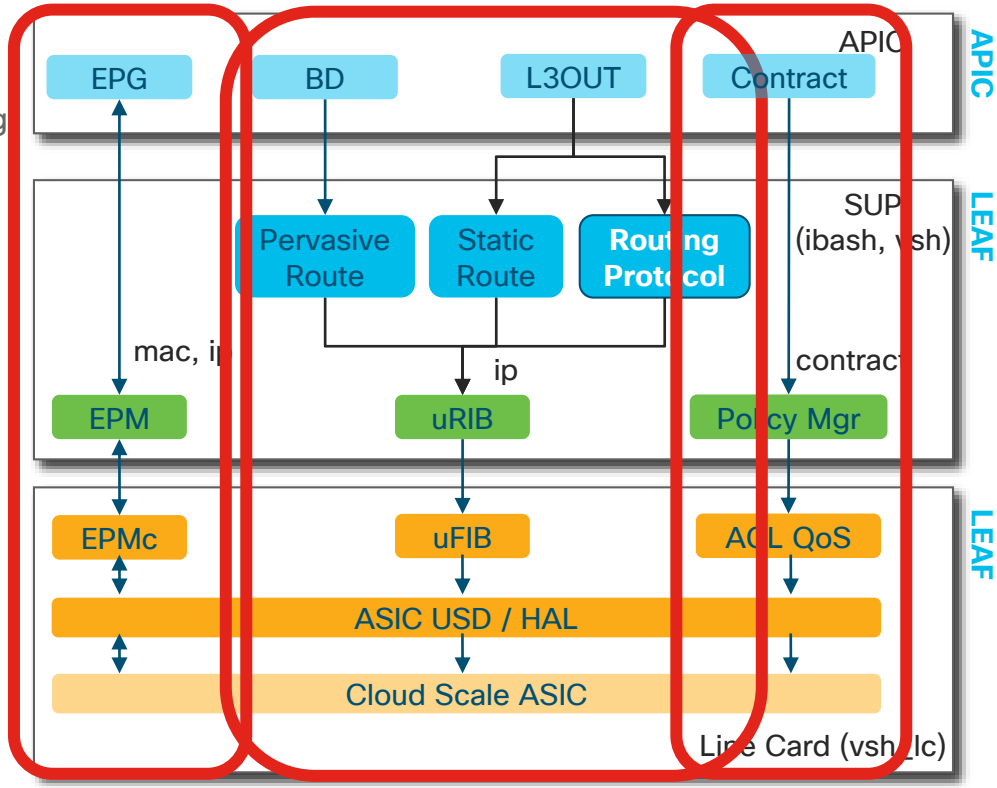
on the **Linecards**:

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 ....

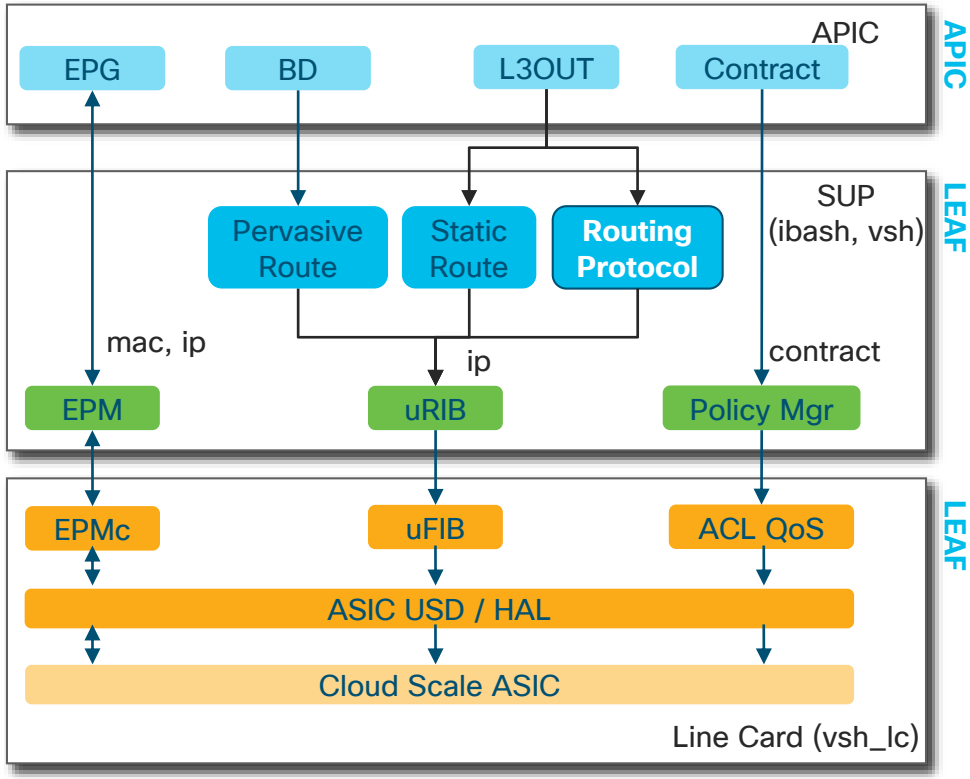
on the **Linecards**: **vsh_lc**

EPMc show system internal epmc ...

uFIB show forwarding ...

ACL QoS show system internal aclqos ...

HAL show platform internal hal ...

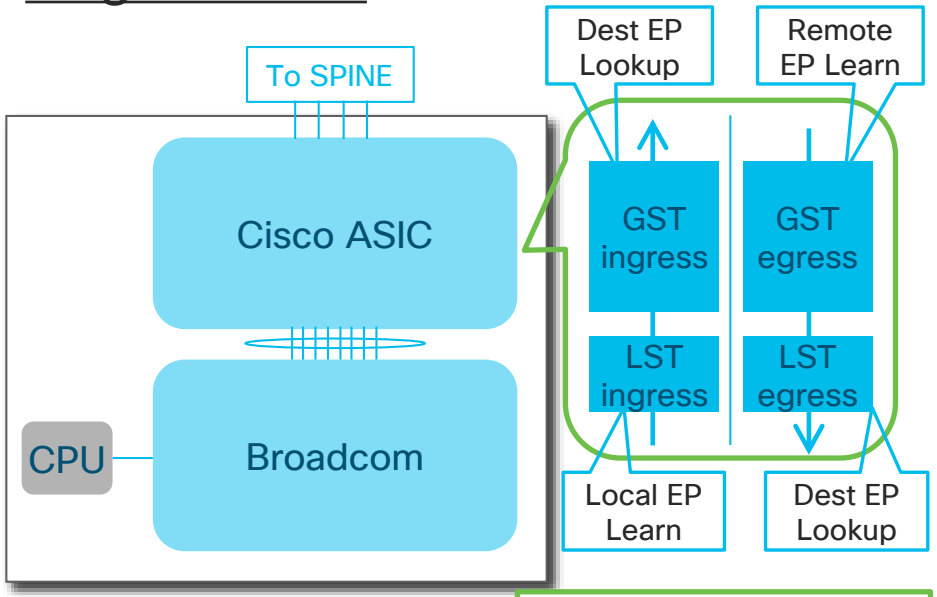


※ ASIC USD (User Space Driver) is only for 1st generation ASIC

LEAF ASIC Generations

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

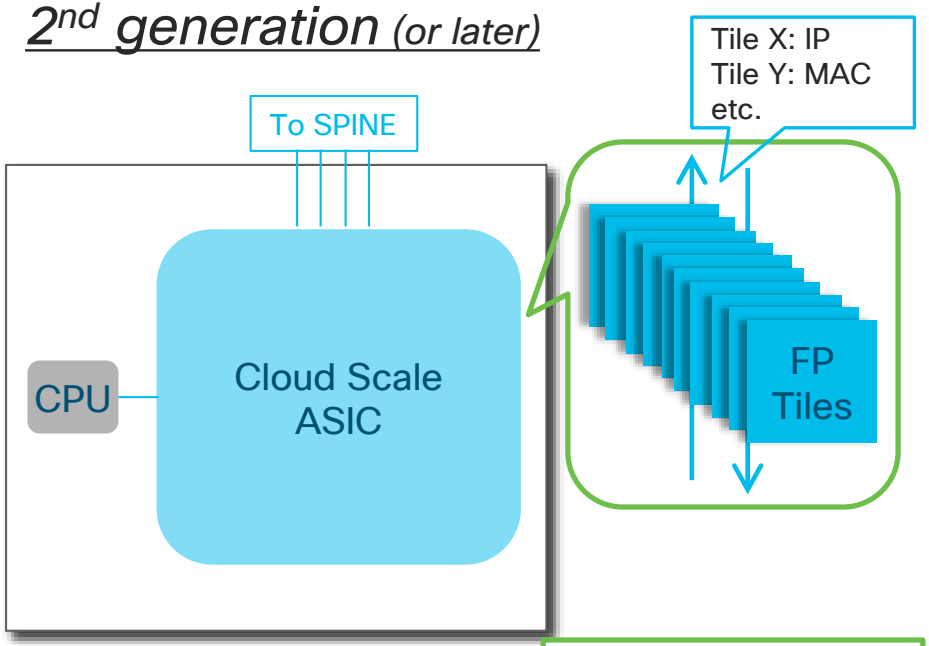
1st generation



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

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

2nd generation (or later)



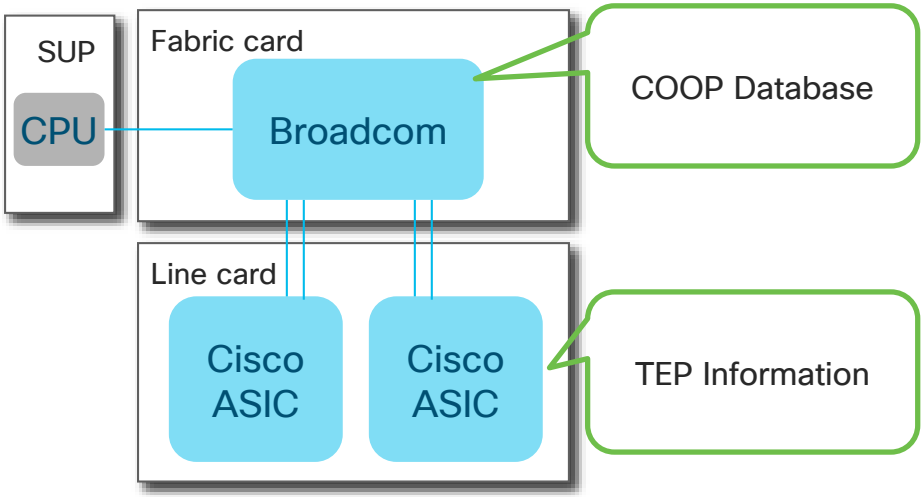
N9K-C93180YC-EX
N9K-C93108TC-EX
N9K-C93180LC-EX
N9K-C93180YC-FX
N9K-C93108TC-FX
N9K-C9348GC-FXP
N9K-C9336C-FX2
N9K-C93240YC-FX2

- 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

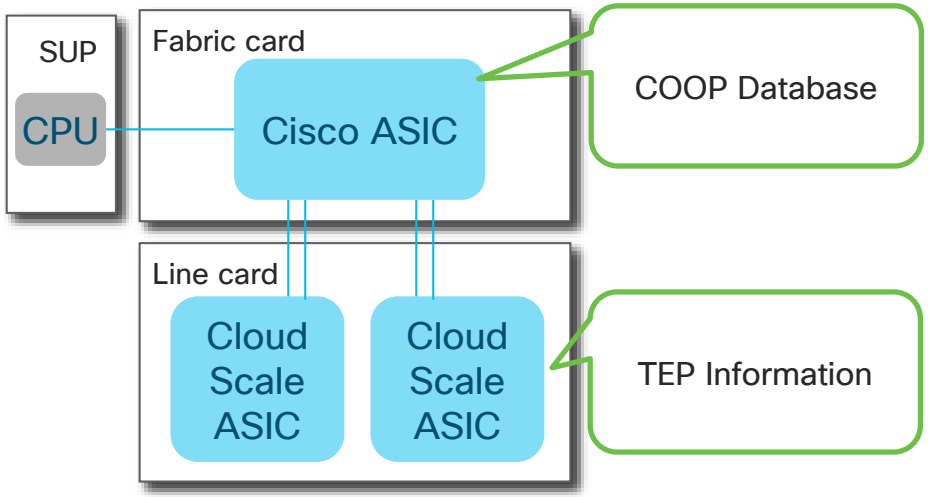


Line card
N9K-X9736PQ

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

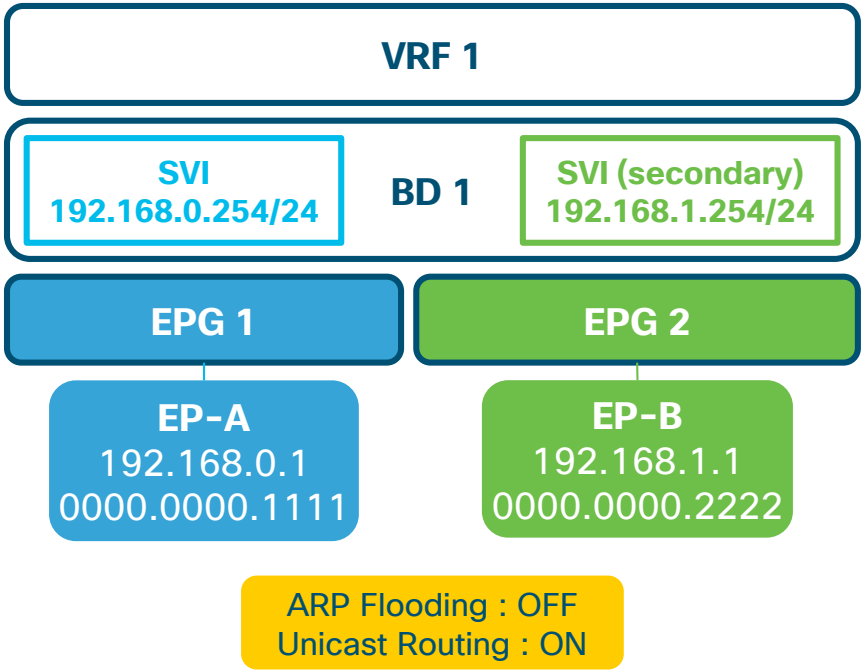
Box spine
N9K-C9364C
N9K-C9332C

Fabric card
N9K-C9504FM-E
N9K-C9508FM-E
N9K-C9508FM-E2
N9K-C9516FM-E2

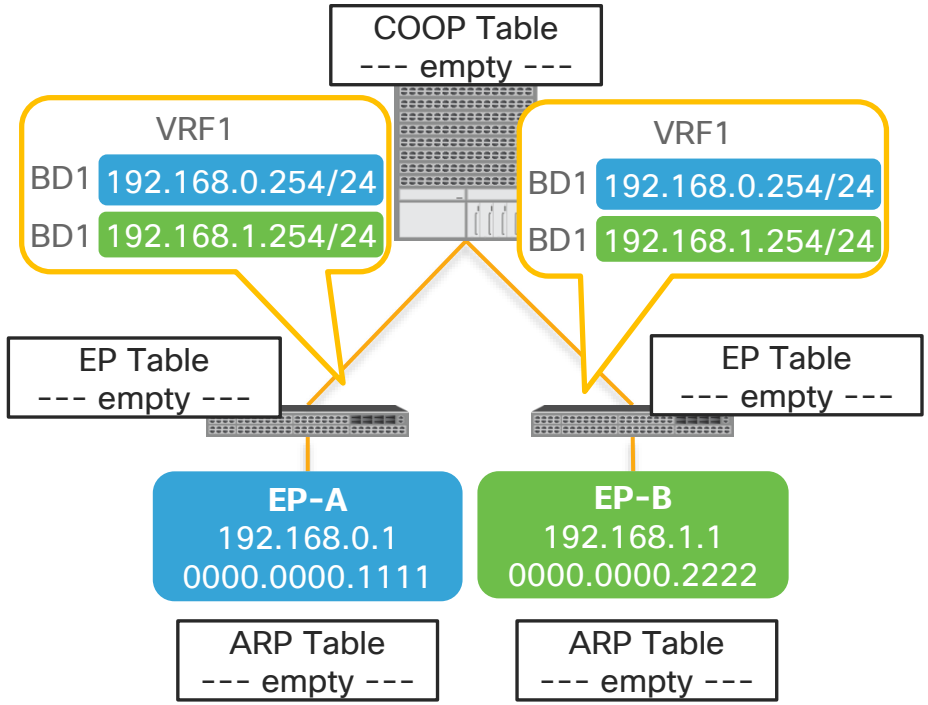
Agenda

- 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

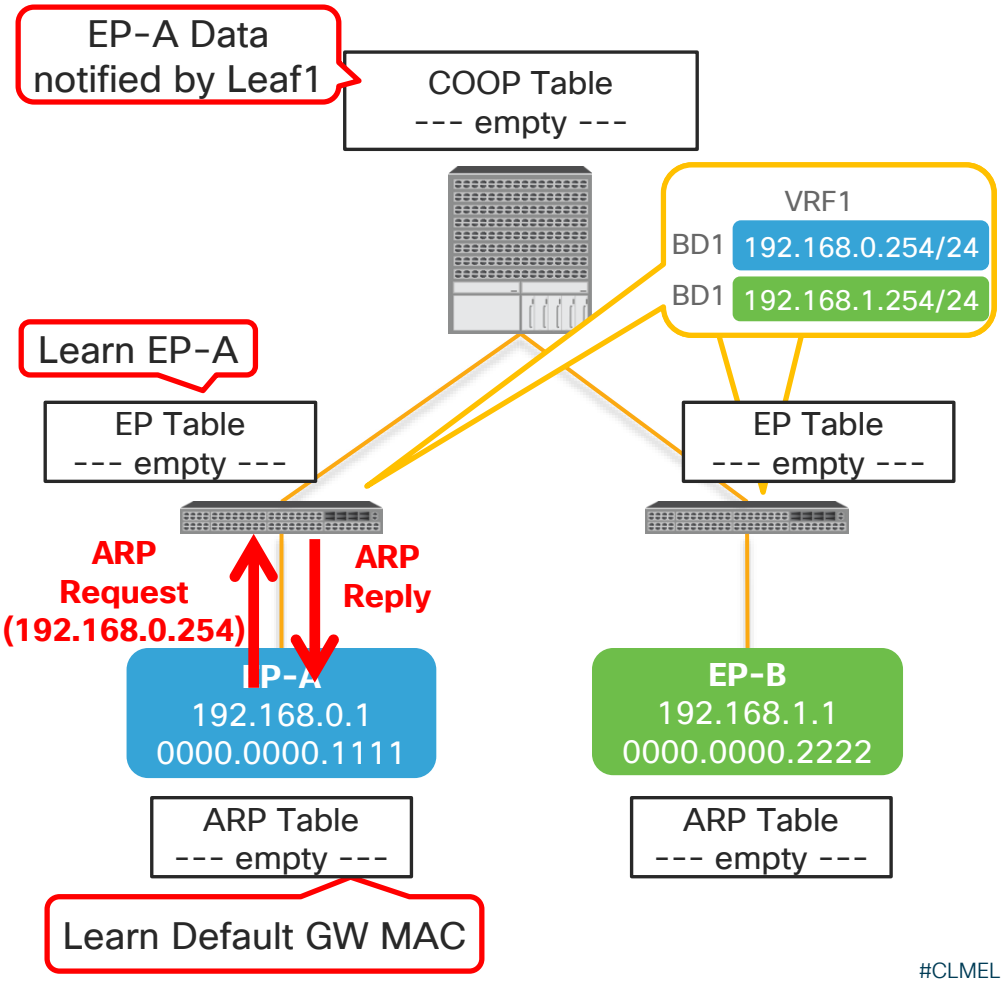
- Logical Topology



- Physical Topology

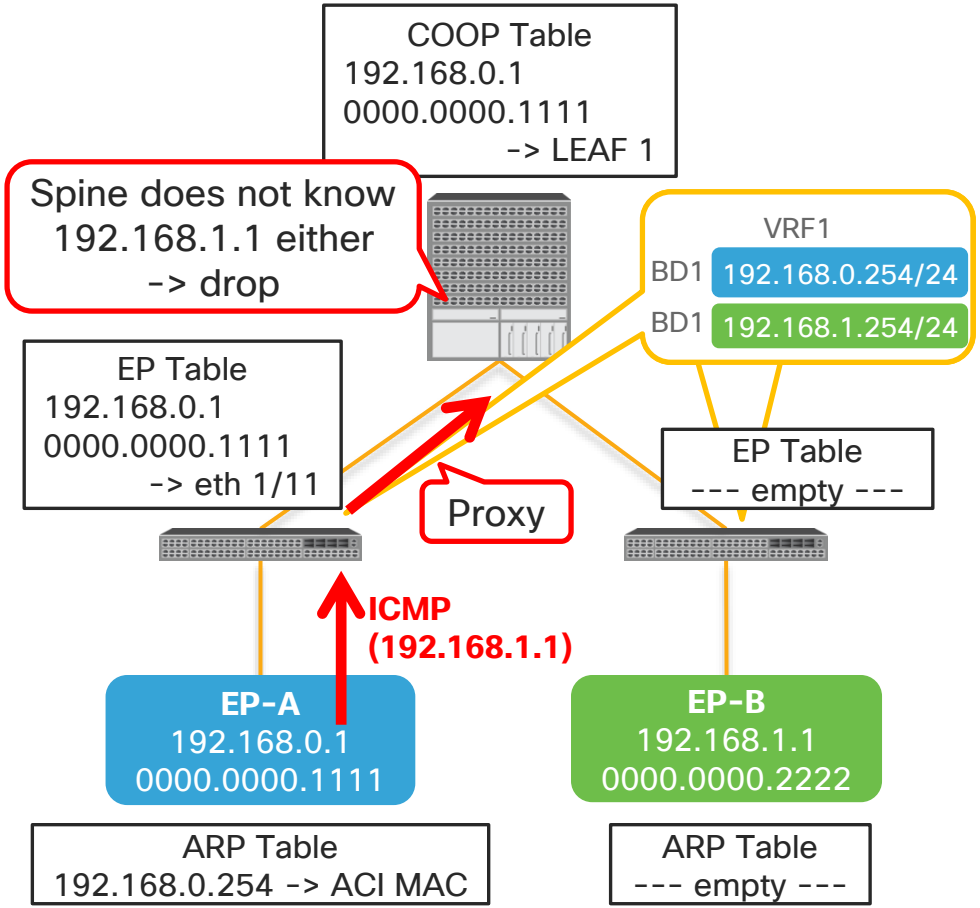


PING : EP-A (192.168.0.1) -> EP-B (192.168.1.1)



1. ARP Request to default GW

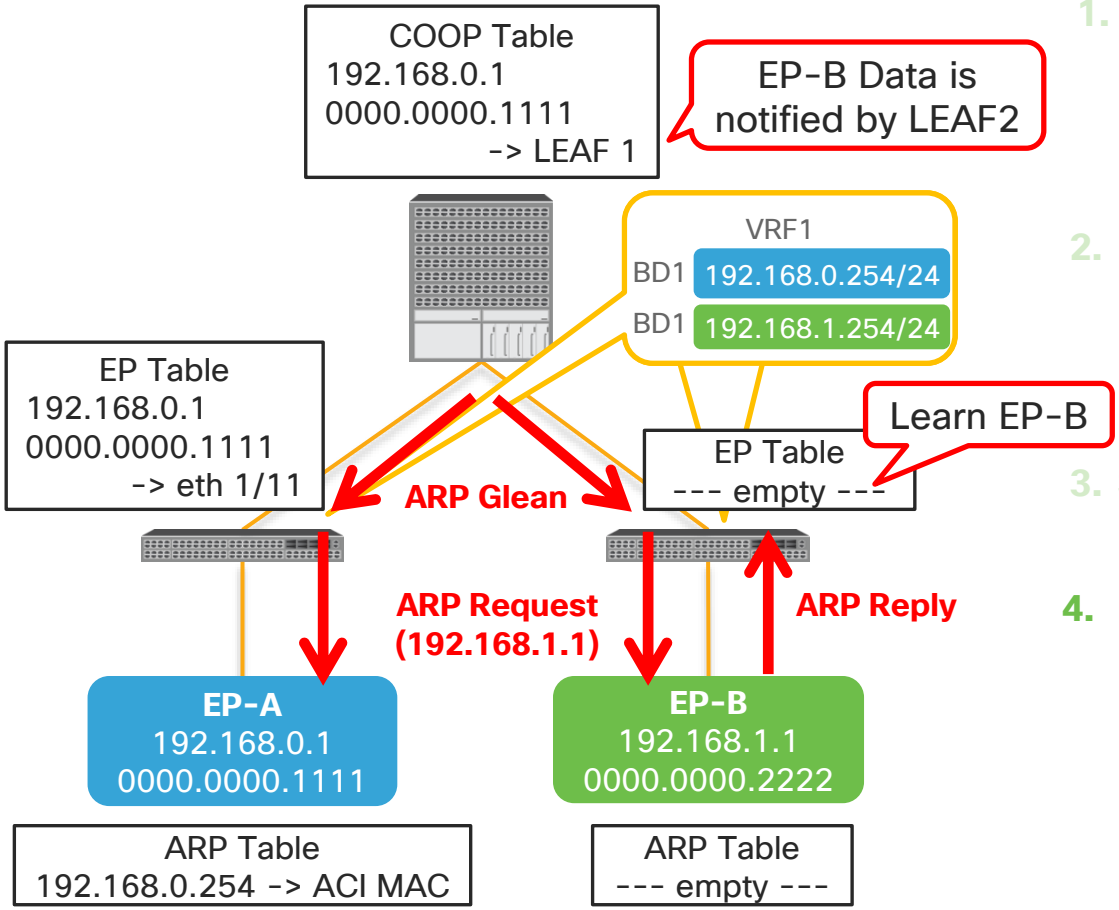
1. ARP Req is sent out to GW (192.168.0.254)
2. LEAF1 learns src IP/MAC from ARP.
   - Leaf1 notify that to Spine COOP
3. LEAF1 sends ARP reply to EP-A.



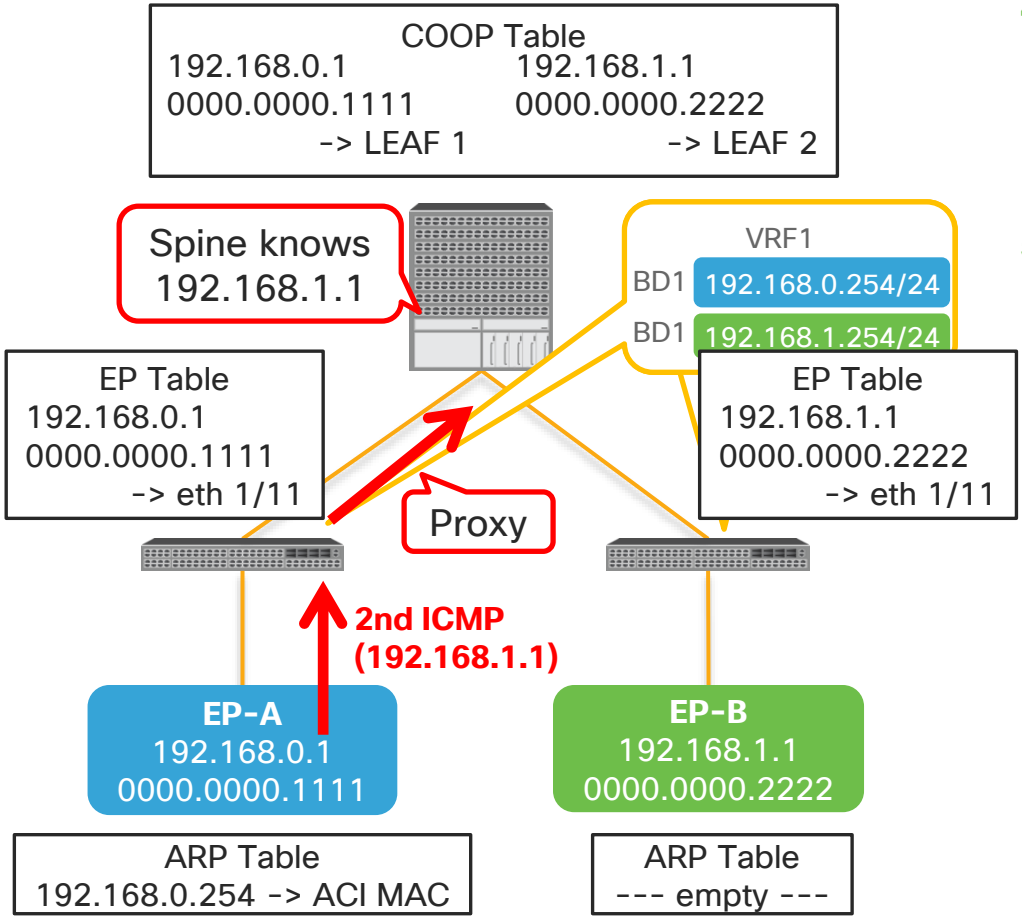
1. **ARP Request to default GW**
   1. ARP Req is sent out to GW (192.168.0.254)
   2. LEAF1 learns src IP/MAC from ARP.
      - Leaf1 notify that to Spine COOP
   3. LEAF1 sends ARP reply to EP-A.

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

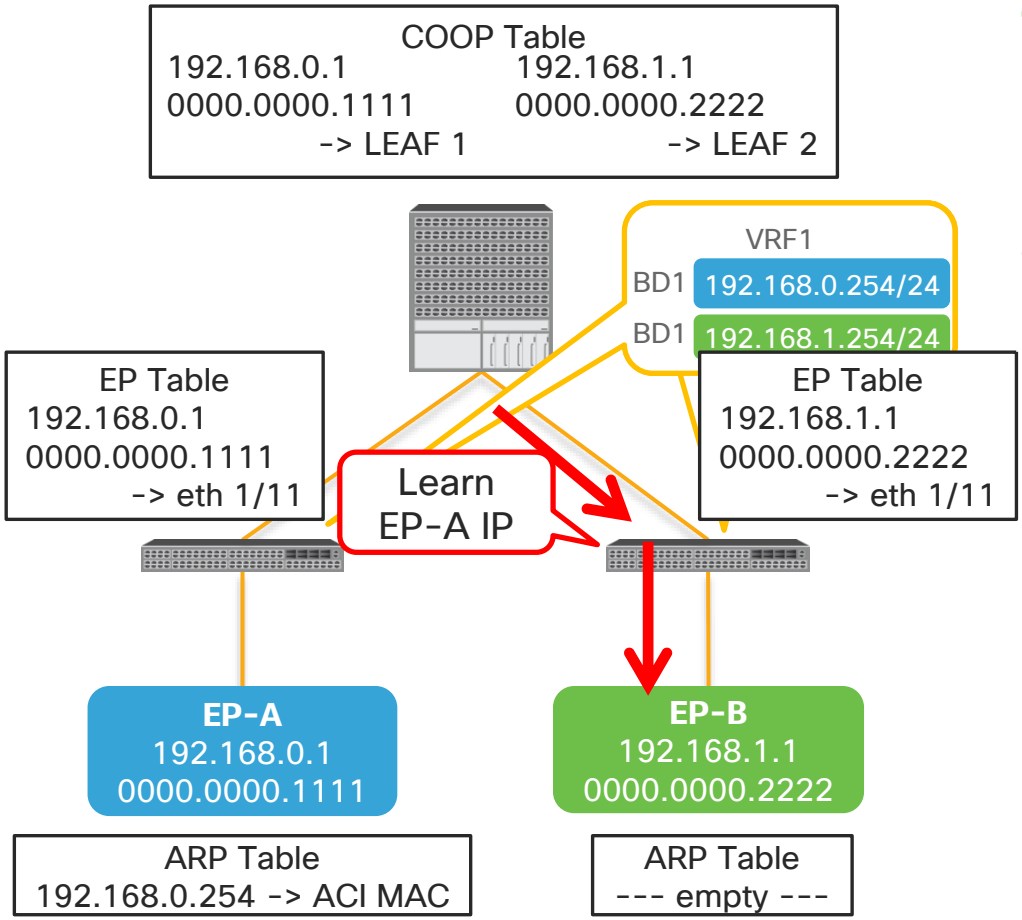
3. **Spine COOP lookup**
   1. COOP doesn't know 192.168.1.1 either
      - drop



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

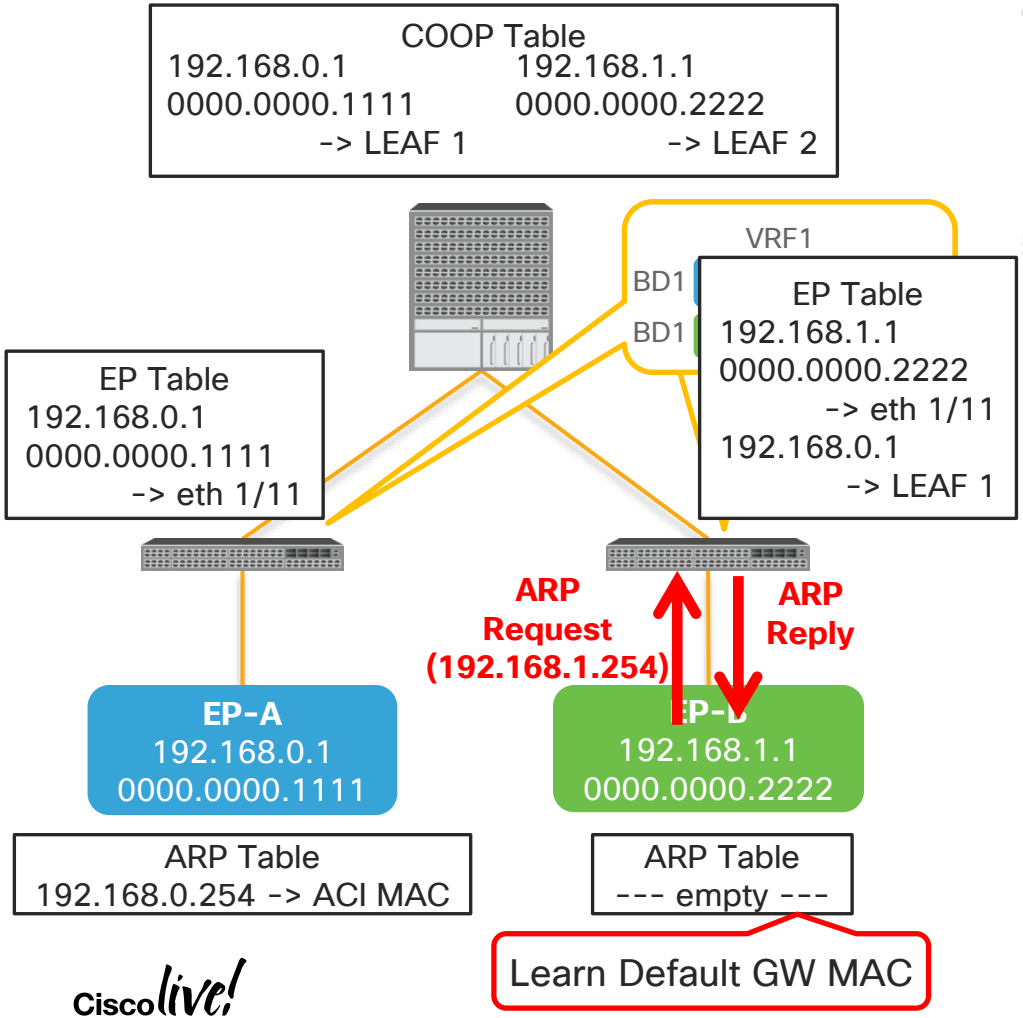


4. **EP-A sends 2nd ICMP to EP-B (192.168.1.1)**
   1. Dst MAC is ACI MAC (BD SVI router-mac)
      - L3 Lookup within VRF
   2. LEAF1 still doesn't know 192.168.1.1 but knows it's subnet (192.168.1.0/254)
      - Spine-Proxy
5. **Spine COOP lookup for 2nd ICMP**
   1. Now COOP knows 192.168.1.1
   2. Spine sends it to Leaf2

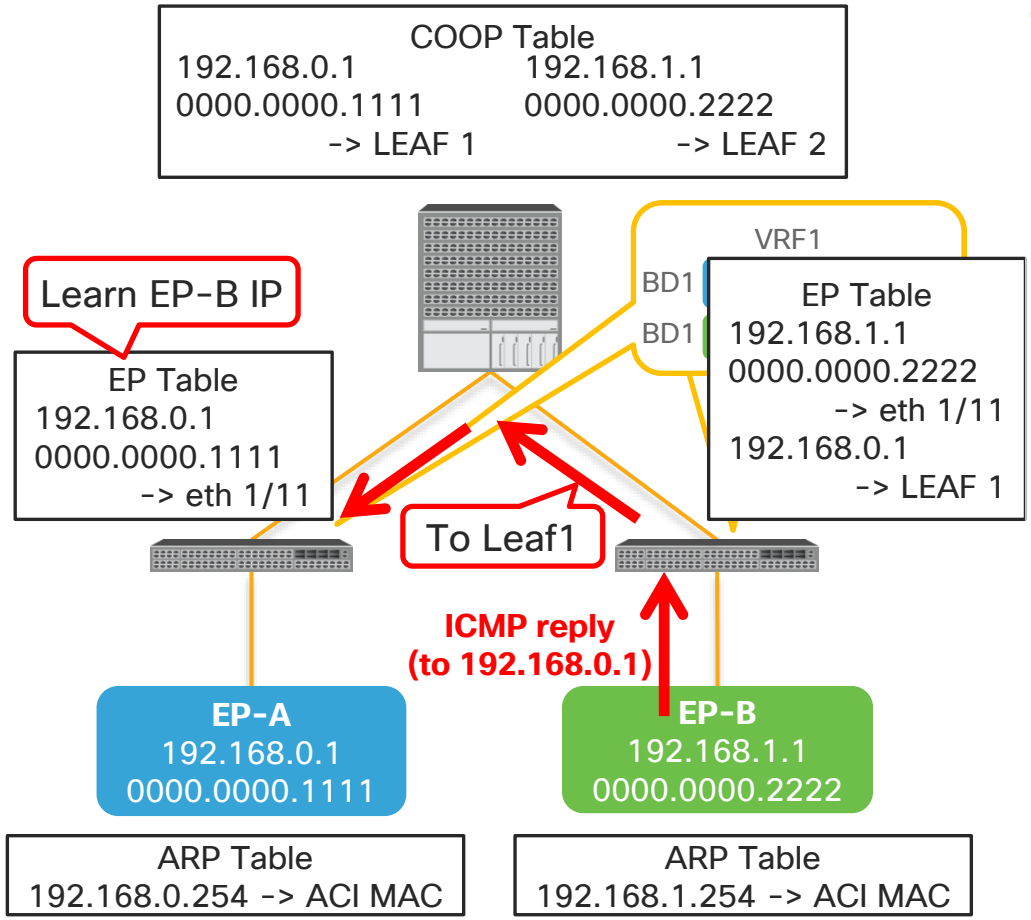


4. EP-A sends 2nd ICMP to EP-B (192.168.1.1)
   1. Dst MAC is ACI MAC (BD SVI router-mac)
      - L3 Lookup within VRF
   2. LEAF1 still doesn't know 192.168.1.1 but knows it's subnet (192.168.1.0/254)
      - Spine-Proxy
5. Spine COOP lookup for 2nd ICMP
   1. Now COOP knows 192.168.1.1
   2. Spine sends it to Leaf2

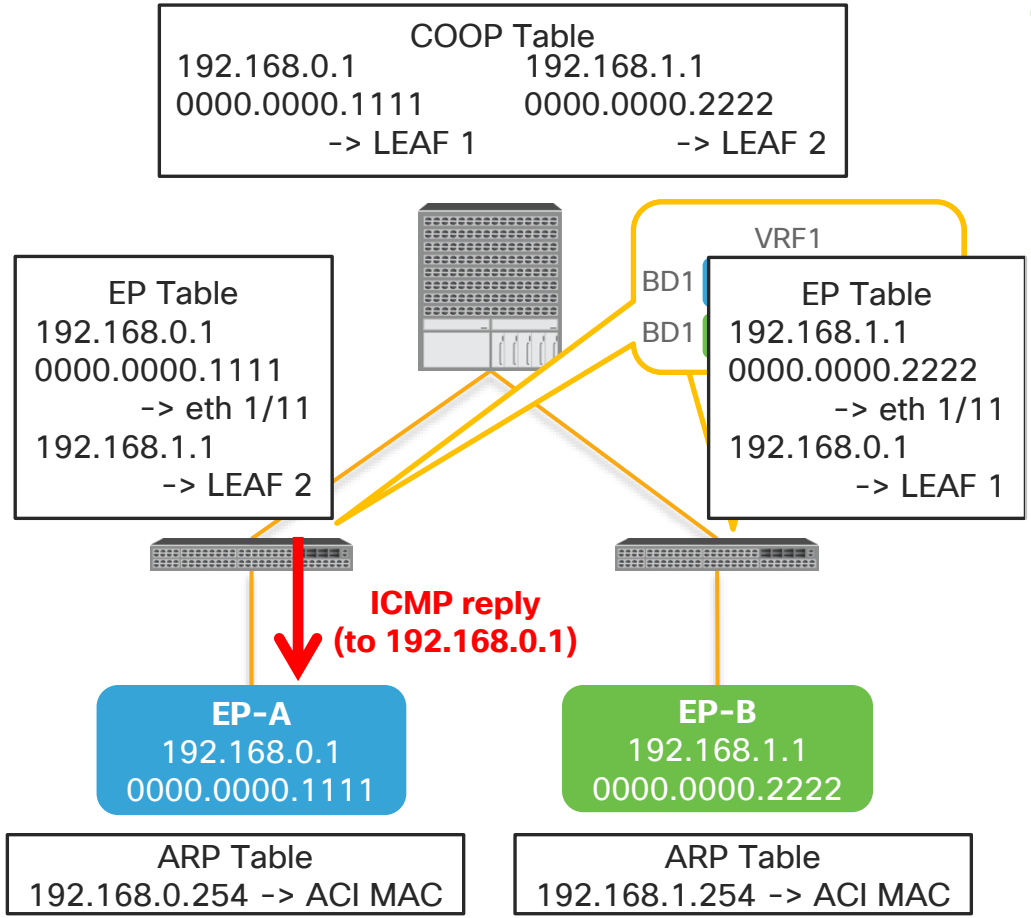
6. **LEAF2 learns EP-A as a remote EP**
   - The packet is routed = sent out with VRF VNID.
     Only IP is learned
7. **LEAF2 sends it out to EP-B**



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)**



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



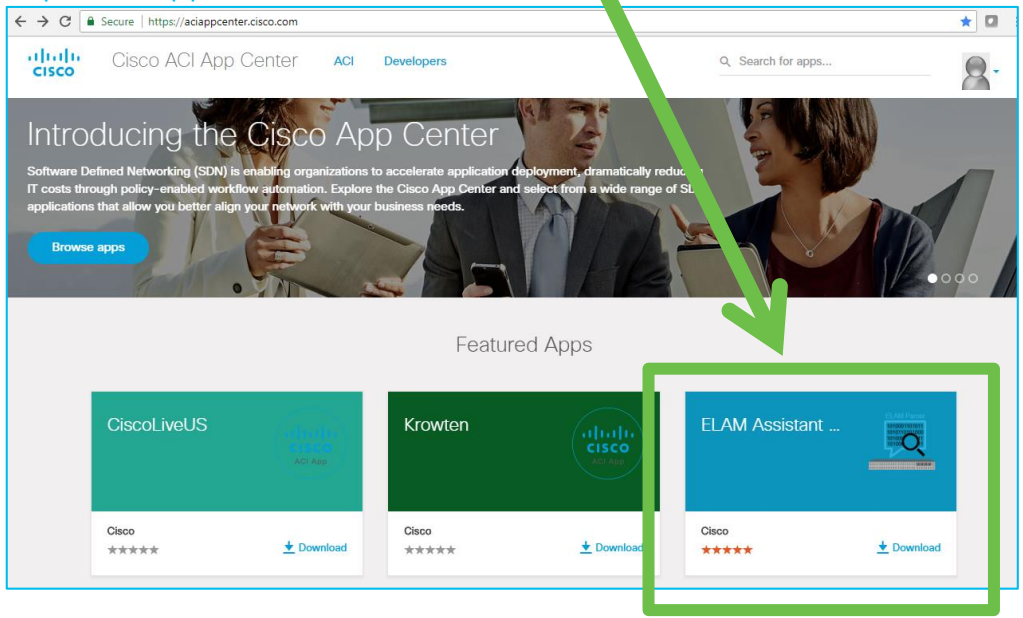
4. **EP-A sends 2nd ICMP to EP-B (192.168.1.1)**
   1. Dst MAC is ACI MAC (BD SVI router-mac)
      - L3 Lookup within VRF
   2. LEAF1 still doesn't know 192.168.1.1 but knows it's subnet (192.168.1.0/254)
      - Spine-Proxy
5. **Spine COOP lookup for 2nd ICMP**
   1. Now COOP knows 192.168.1.1
   2. 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

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

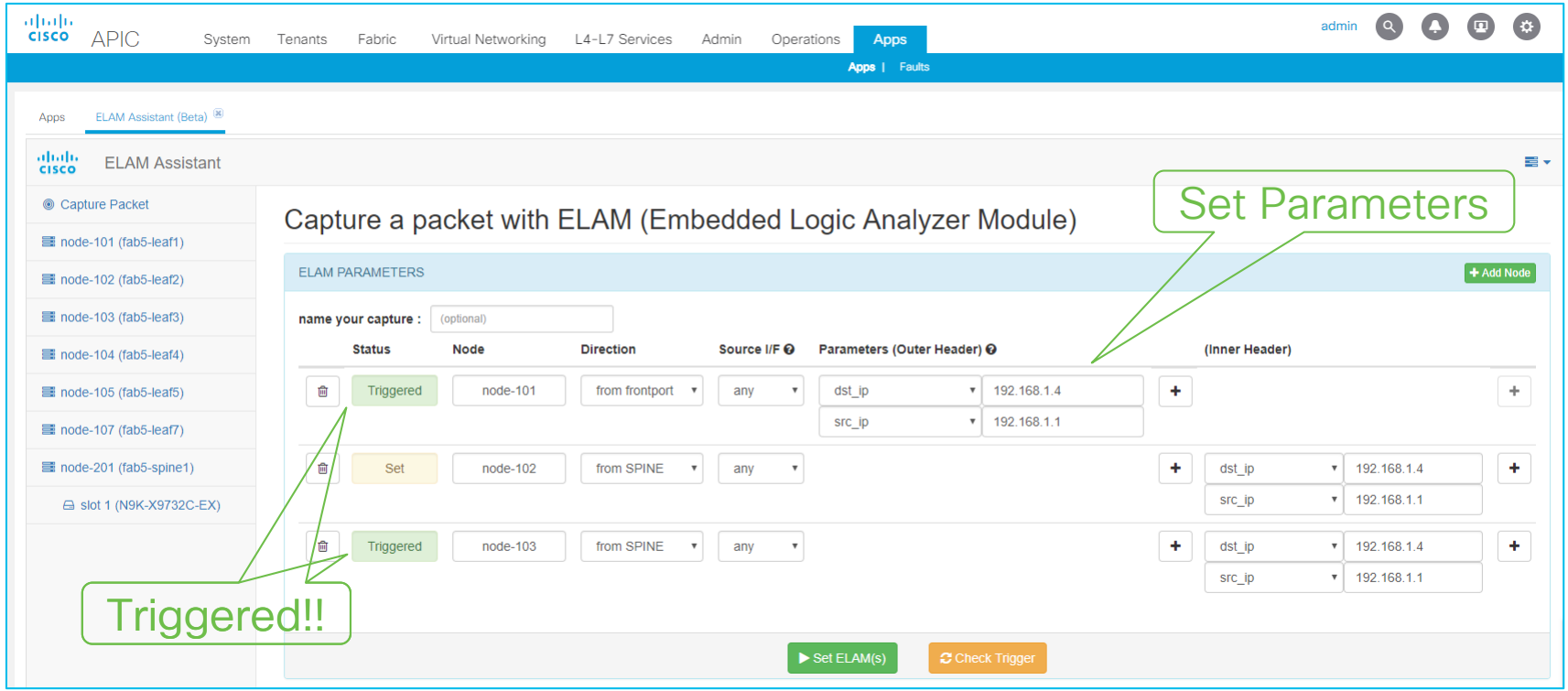
- You can perform ELAM like a TAC engineer!
- With a nicely formatted result report

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



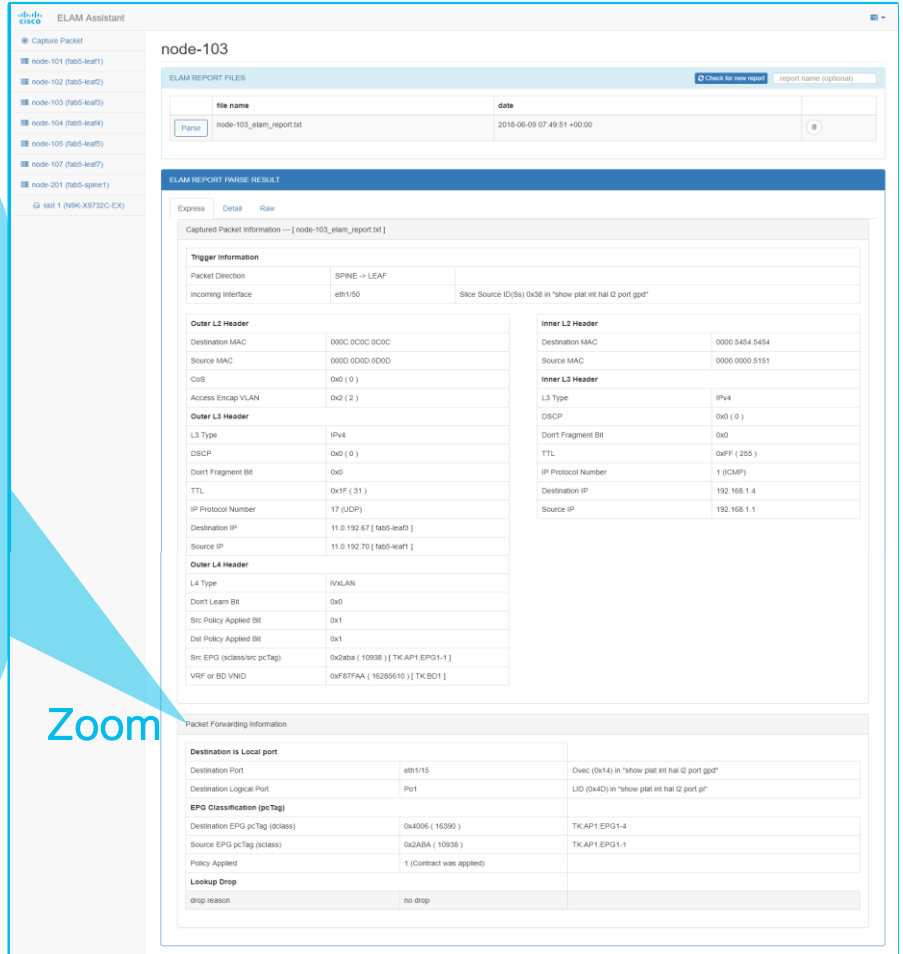
ELAM Assistant in ACI AppCenter (example)

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

Zoom



Continue your education

Complete Your Online Session Evaluation

- Give us your feedback and receive a complimentary **Cisco Live 2019 Power Bank** after completing the overall event evaluation and 5 session evaluations.
- All evaluations can be completed via the Cisco Live Melbourne Mobile App.
- Don’t forget: Cisco Live sessions will be available for viewing on demand after the event at:

  https://ciscolive.cisco.com/on-demand-library/

Thank you

INTUITIVE