ACI Deep Dive

L3 out troubleshooting

ACI Solution TAC team

Roland Ducomble – Technical Leader EMEAR TAC - CCIE 3745

Dec 2019

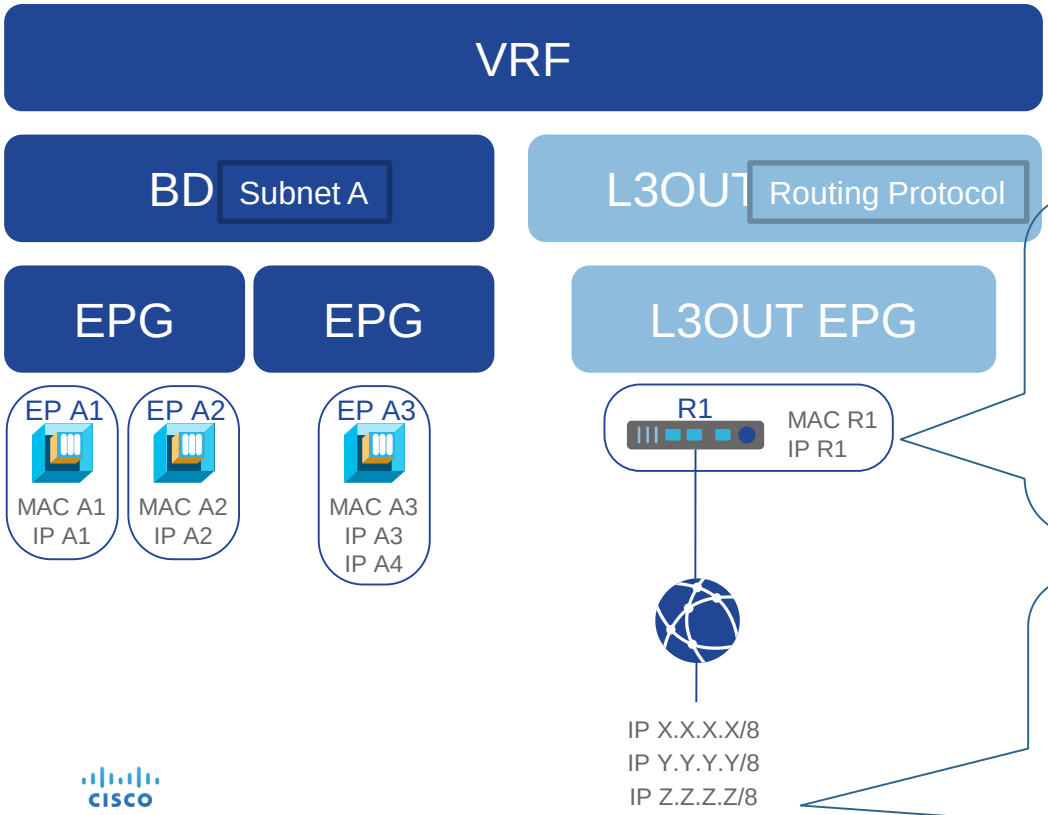
V6.2

L3 out introduction and iBGP route distribution

Why L3OUT?

- What is L3OUT for ?
  - To connect ACI with other network domain
    = devices with multiple subnet behind it
- How is L3OUT different from EPG?
  - Speak Routing Protocol
  - No IP learning as endpoint
  - Next-hop IP is stored in ARP table

= Same as normal routers



Next-hop MAC in endpoint table

```
leaf1# show endpoint vlan 84
84/TK:VRF1      vxlan-14876665    0000.0000.R1R1    L       po3
```

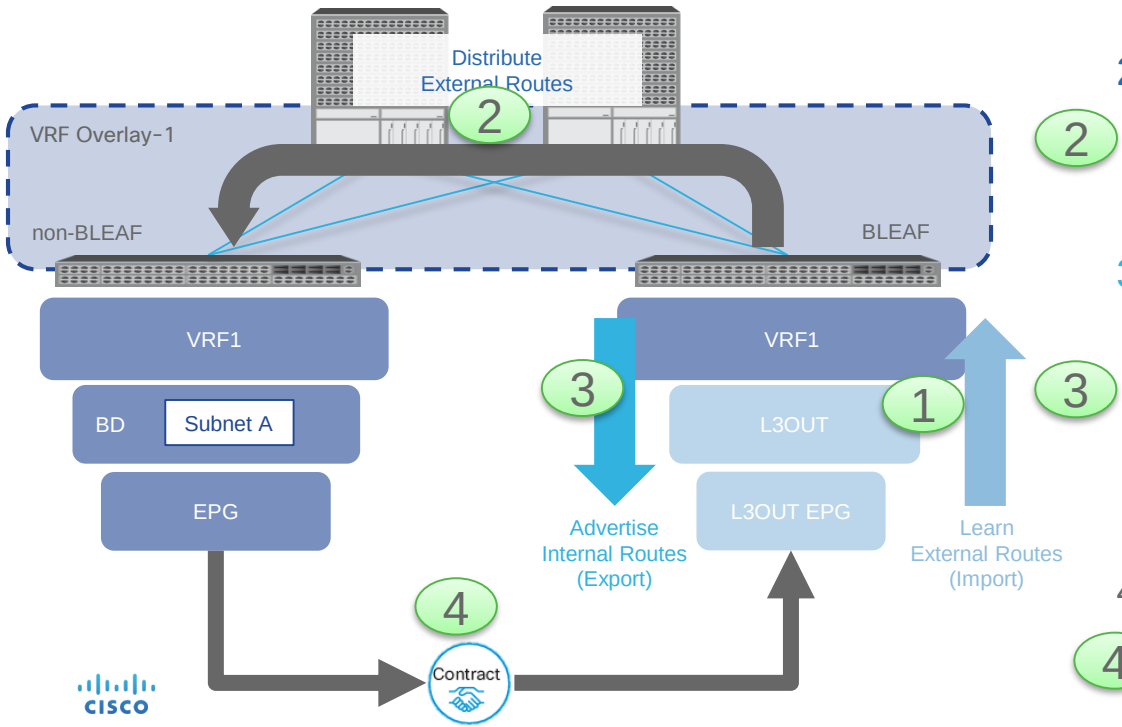
Next-hop IP in ARP table (only for L3OUT)

```
leaf1# show ip arp vlan 84
Address          Age       MAC Address     Interface
R.R.R.1          00:07:51  0000.0000.R1R1  vlan84
```

Routes via Routing Protocol

```
leaf1# show ip route vrf TK:VRF1
X.0.0.0/8, ubest/mbest: 1/0
    *via R.R.R.1, vlan84, [110/5], 2d00h, ospf-default, intra
Y.0.0.0/8, ubest/mbest: 1/0
    *via R.R.R.1, vlan84, [110/5], 2d00h, ospf-default, intra
Z.0.0.0/8, ubest/mbest: 1/0
    *via R.R.R.1, vlan84, [110/5], 2d00h, ospf-default, intra
```

L3OUT Key Components



1. Learn external routes
   - Routing Protocol in L3OUT
2. Distribute external routes to other leaves
   - MP-BGP
3. Advertise internal routes (BD subnet) to outside
   - Redistribution and
   - Contract
4. Allow traffic with contracts
   - L3OUT EPG (Prefix Based EPG)

Infra Setup for iBGP Route Distribution

Intro

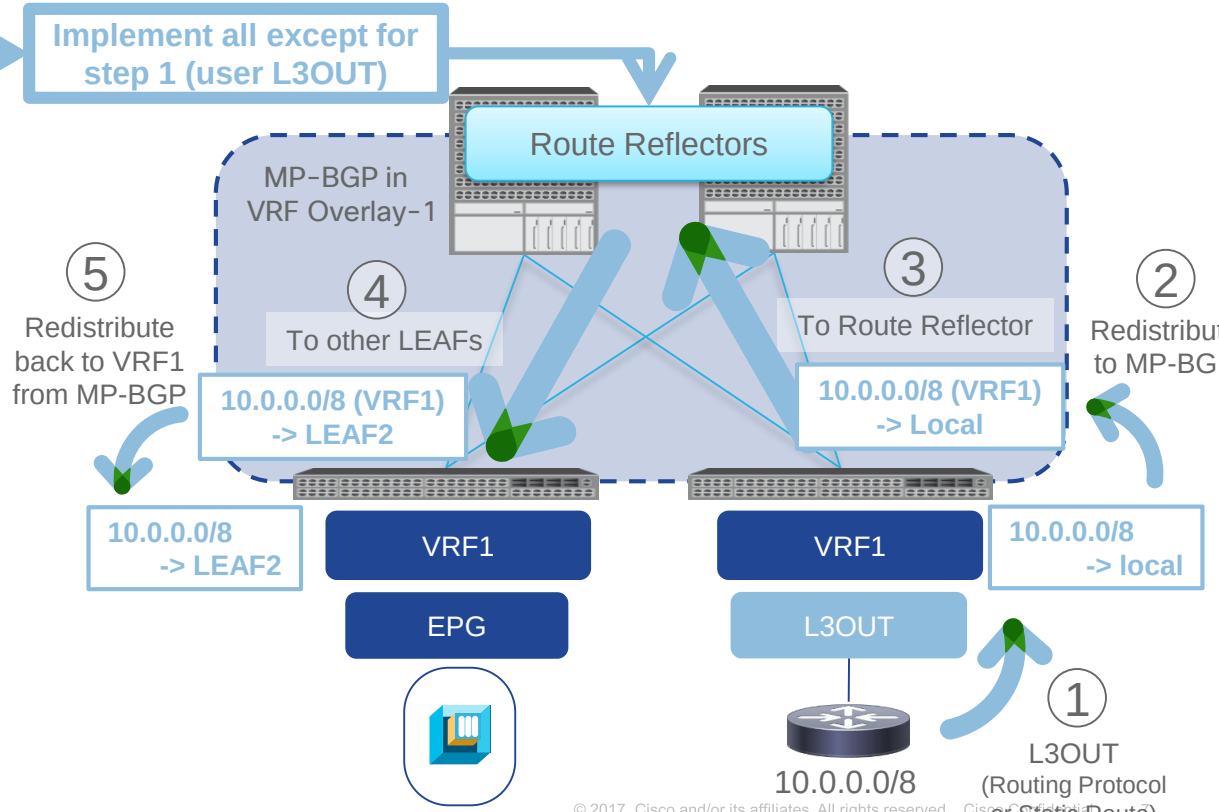
- MP-iBGP VPNv4 is used to distributed external route in the fabric
- Config here is to be done once per fabric

L3OUT Key Components

2. Distribute External Routes = MP-BGP in infra

Configurations

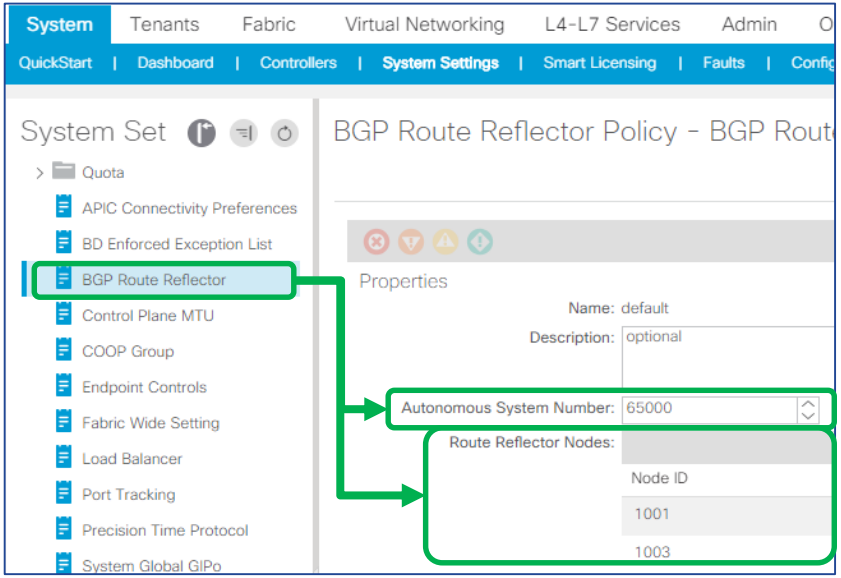
- Pod Profile
- Pod Policy Group
- BGP Route Reflector Policy
  - default
- System Settings
- BGP Route Reflector
  - ACI BGP AS number (for both MP-BGP and L3OUT BGP)
  - MP-BGP Route Reflector Spines



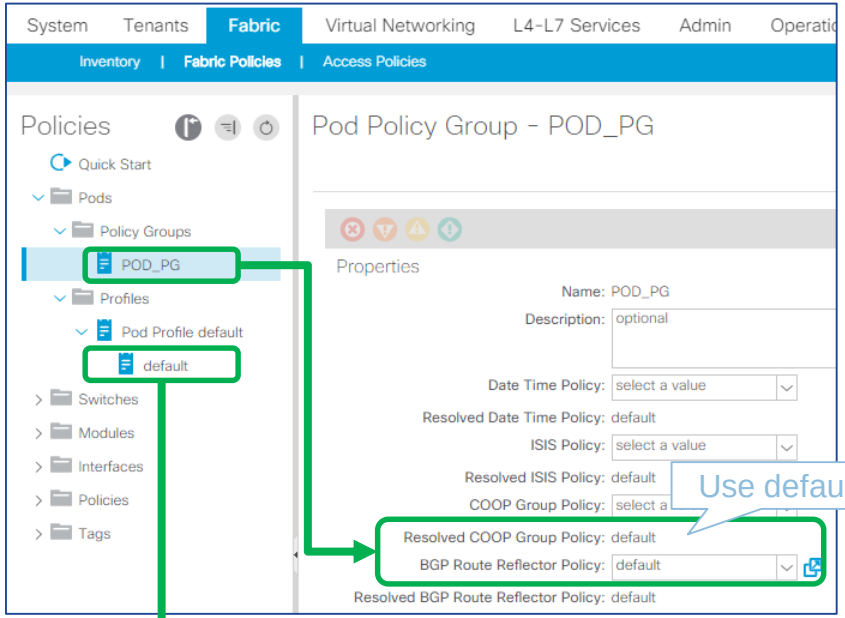
L3OUT Key Components

2. Distribute External Routes = MP-BGP in infra

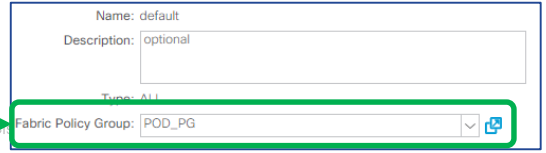
1. Select ACI BGP AS and Route Reflector SPINEs



2. Apply Route Reflector policy to Pod Policy Group



3. Apply Pod Policy Group to Pod Profile



※ L3OUT BGP share this same AS with the internal MP-BGP

CLI Verification

1. Do both border leaf and non-border leaf have BGP sessions with RR spines?

```
leaf# show bgp sessions vrf overlay-1
Neighbor        ASN     Flaps LastUpDn|LastRead|LastWrit St Port(L/R)  Notif(S/R)
10.0.184.65     65003   0     2d07h   |never   |never    E  37850/179  0/0
10.0.184.66     65003   0     2d07h   |never   |never    E  45089/179  0/0

leaf# acidiag fnvread | grep spine
    1001        1       spine1      FGE10000000     10.0.184.65/32   spine         active   0
    1002        1       spine2      SAL10000000     10.0.184.66/32   spine         active   0
```

2. Is the external route learned on a border leaf?

```
border-leaf# show ip route vrf TK:VRF1
10.0.0.0/8, ubest/mbest: 1/0
    *via 15.0.0.1, Vlan58, [110/5], 2d08h, ospf-default, intra
```

3. Does non-border leaf show the expected border leaf as next-hop?

```
non-border-leaf# show ip route vrf TK:VRF1
10.0.0.0/8, ubest/mbest: 2/0
    *via 10.0.184.67%overlay-1, [200/5], 2d08h, bgp-65003, internal, tag 65003
    *via 10.0.184.64%overlay-1, [200/5], 2d08h, bgp-65003, internal, tag 65003

non-border-leaf# acidiag fnvread
      ID   Pod ID                 Name     Serial Number         IP Address     Role        State    LastUpdMsgId
-------------------------------------------------------------------------------------------------------------------
     103        1                leaf3      SAL10000003     10.0.184.64/32     leaf         active   0
     104        1                leaf4      SAL10000004     10.0.184.67/32     leaf         active   0
```

CLI Verification

We have vpnv4 Address-family

```
bdsol-aci32-spine2# show bgp vpnv4 unicast summary vrf overlay-1
BGP summary information for VRF overlay-1, address family VPNv4 Unicast
BGP router identifier 10.10.32.212, local AS number 132
BGP table version is 418, VPNv4 Unicast config peers 10, capable peers 8
144 network entries and 161 paths using 28632 bytes of memory
BGP attribute entries [108/15552], BGP AS path entries [1/6]
BGP community entries [0/0], BGP clusterlist entries [4/24]

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.80.94      4   132   12597   12795      418    0    0     1w1d 24
10.0.88.65      4   132   12577   12785      418    0    0     1w1d 0
10.0.88.90      4   132   12619   13056      418    0    0     1w1d 21
10.0.88.91      4   132   12613   12904      418    0    0     1w1d 27
10.0.88.95      4   132   12615   12873      418    0    0     1w1d 27
```

```
bdsol-aci32-spine2# show bgp vpnv4 unicast neighbor 10.0.80.94  vrf overlay-1
…
 Additional Paths capability: advertised received
  Additional Paths Capability Parameters:
  Send capability advertised to Peer for AF:
    VPNv4 Unicast  VPNv6 Unicast
  Receive capability advertised to Peer for AF:
    VPNv4 Unicast  VPNv6 Unicast
```

L3 out example
Config and Troubleshooting eBGP example

Config Steps

1. Configure
   1. L3 out global (VRF, Phys Dome, which RP to use)
   2. L3 out node (BL) including RID (+ loopback) + if BGP the peering
   3. Configure Logical interface aka if on BL to peer

   Note external prefix will be seen in fabric automatically by default
2. Configure one or more L3 out EPG
   1. Includes subnet in it (or use default 0.0.0.0/0) → assignement of external network to the EPG
   2. Configure Contract
3. Decides what to advertise out
   1. BD subnet
   2. Or other network from other L3 out (transit)

L3OUT Key Components

1. Learn External Routes = Routing Protocol

Configurations

External Routed Networks (L3OUT)

- VRF to deploy Routing Protocol
- Routing Protocol parameters
  ex. OSPF area 0.0.0.1 nssa

Node Profile

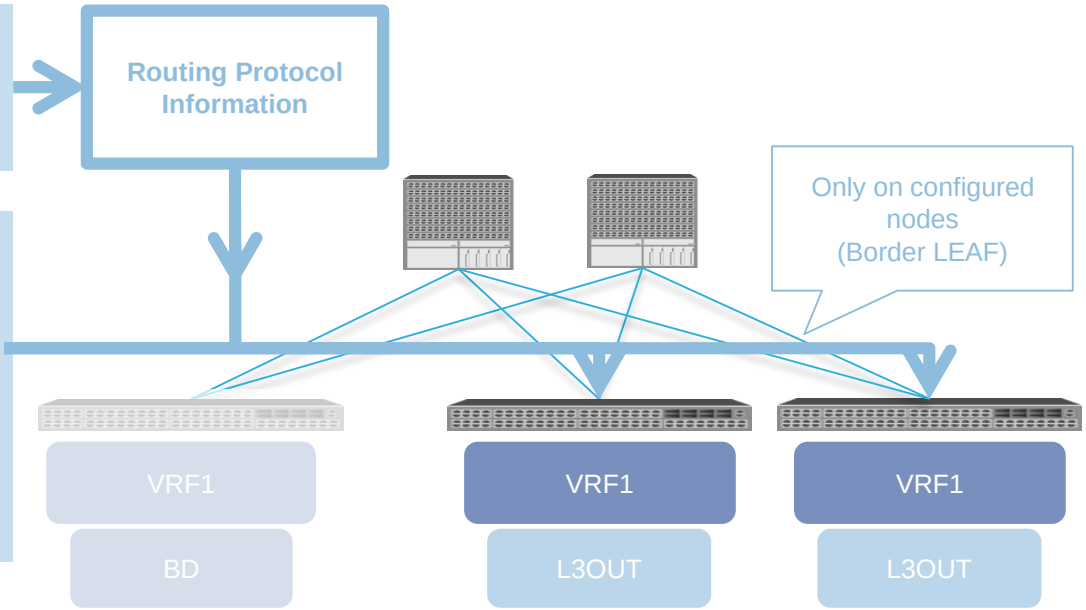
- Node(s) to deploy Routing Protocol
- Static Route (if any)

Interface Profile

- I/F(s) to deploy Routing Protocol
- Routing Protocol I/F parameters
  ex. OSPF hello interval

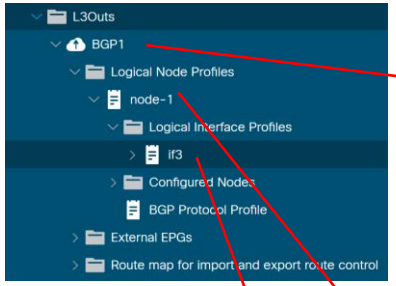
Networks (L3OUT EPG)

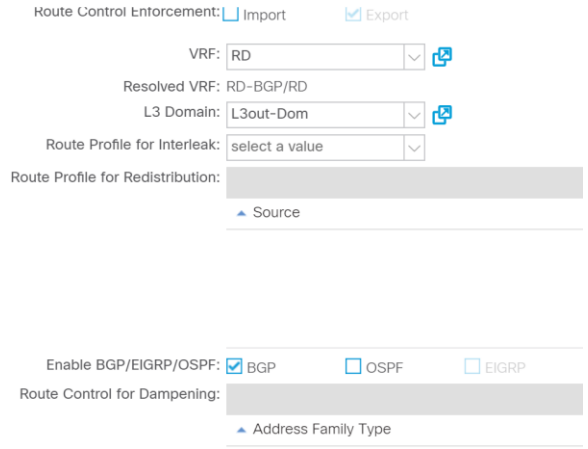
- Contract
- Advanced Route Control
  ex. route-map

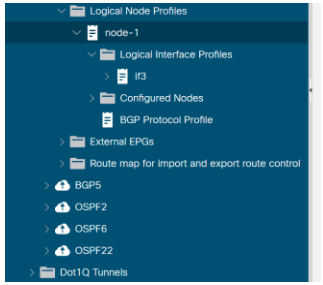


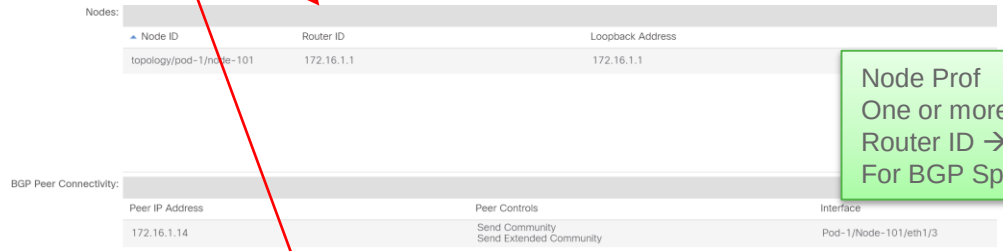
※ Details for each protocol are in later sections
※ Details for L3OUT EPG are in later sections

L3 out – node – if config

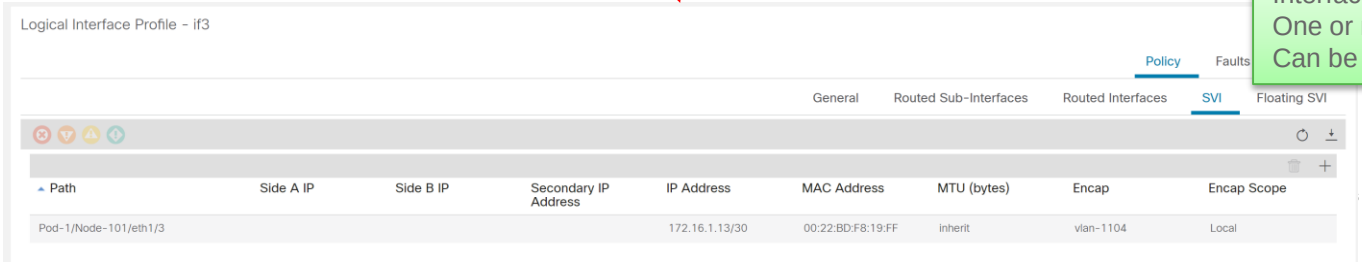






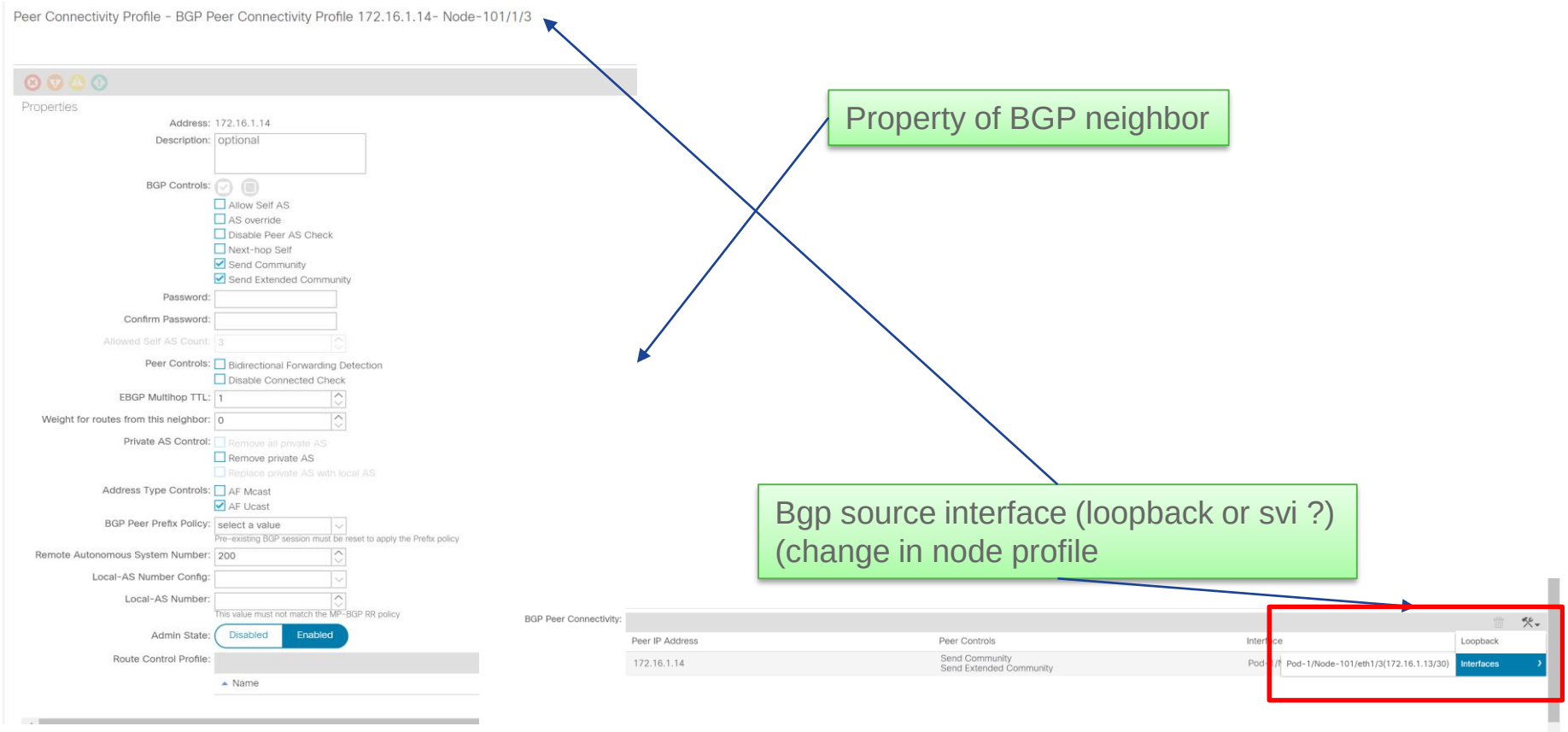


Node Prof
One or more BL
Router ID → by def will be loopback in vrf
For BGP Specify BGP neighbor



Interface profile
One or more per leaf
Can be Routed Subif, Routed If or SVI

Detail of BGP neighbor configuration

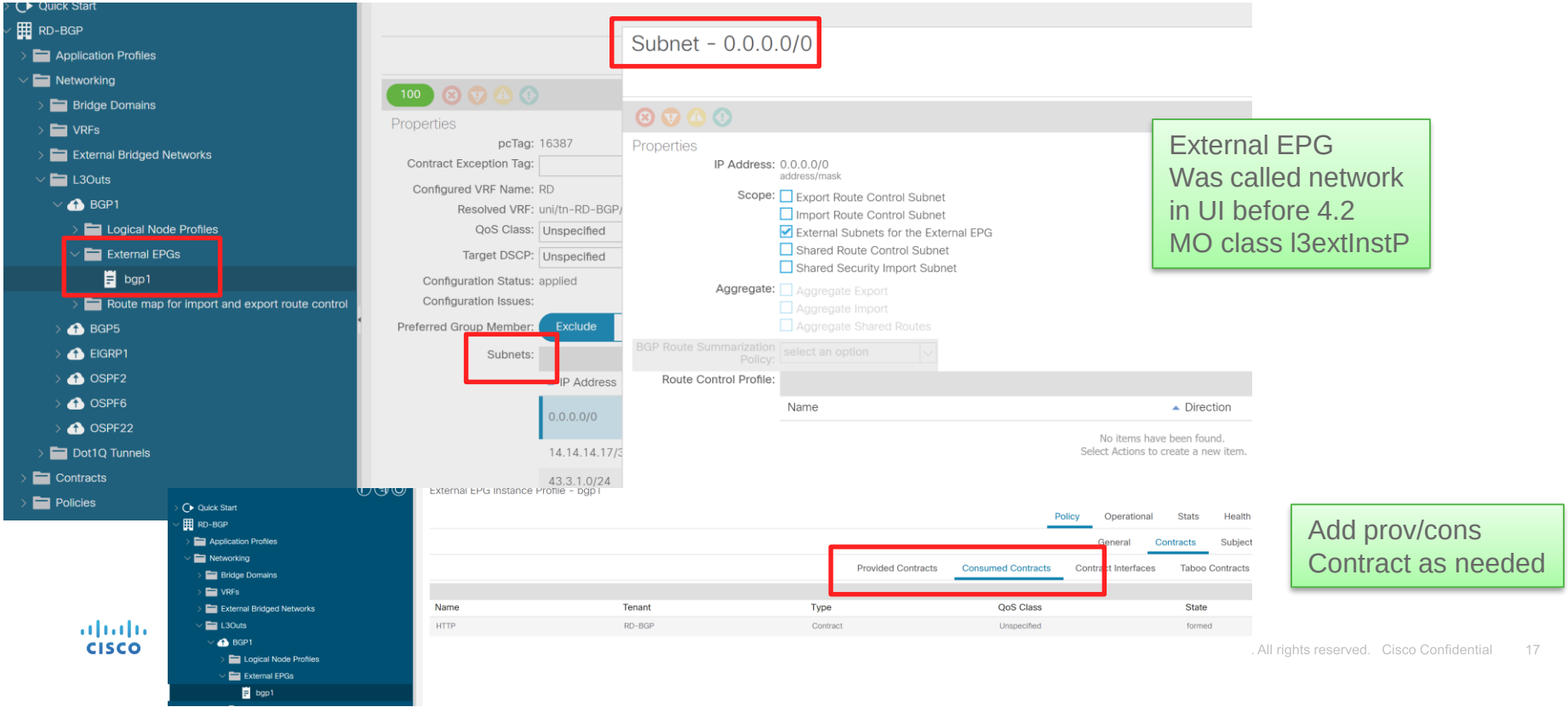


Note on Router ID

- Router ID by default translate to a loopback in the VRF
- If you have multiple L3 out on the same leaf in the same VRF you will need to use the same Router Id, but additional L3 out can’t use the same loopback.
- GUI enforce that and check it
- Ex :
- L3out1- RID 172.16.1.1 Lo 172.16.1.1
- L3out2 – RID 172.16.1.1 Lo 172.16.1.11

2. Configure L3 out EPG

- In case of doubt use 0.0.0.0/0 as external subnet for external epg by default
- Configure a contract on the L3 out to/from where ever if needed



L3OUT Key Components

3. Advertise BD subnet

Configurations

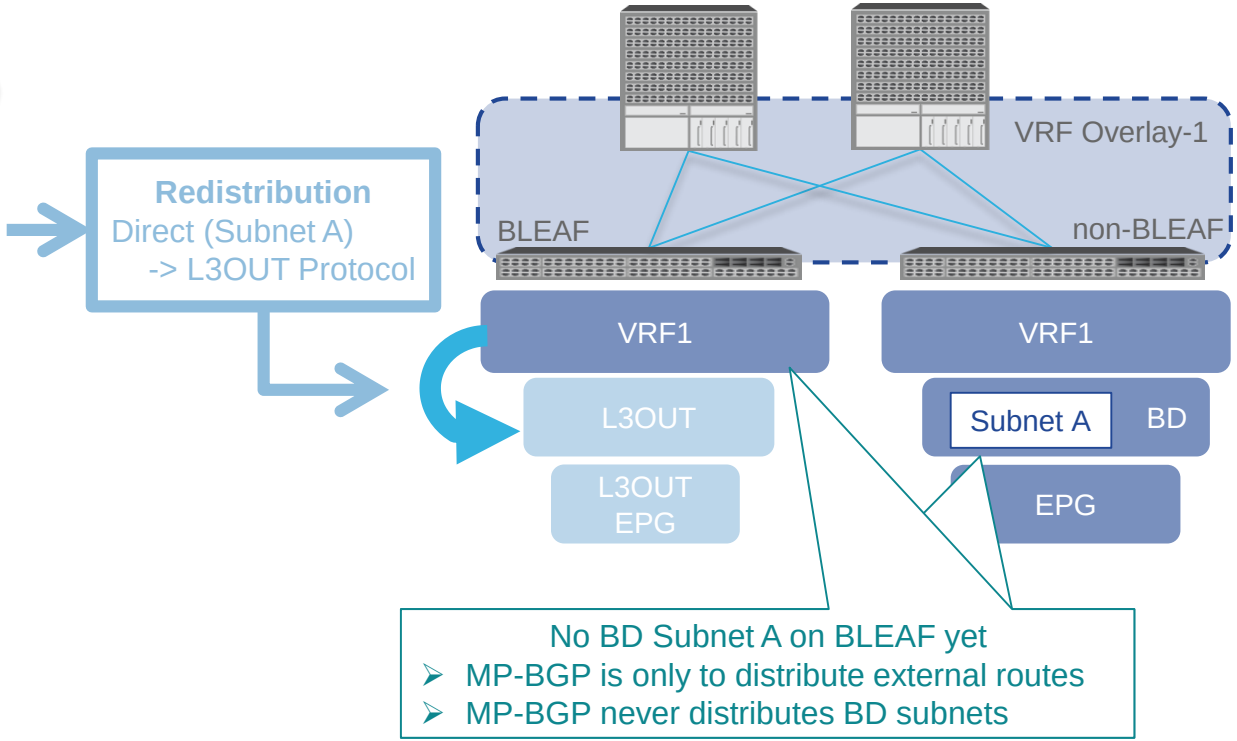
Bridge Domain (BD)

BD Subnet

- Subnet A
  - ✓ "Advertised Externally"

Associated L3OUT

- Target L3OUT(s) to advertise BD subnets



L3OUT Key Components

3. Advertise BD subnet

BD settings needed to update route-map and prefix-list
Contract on L3 EPG needed to ensure BD subnet is on BLeaf

Configurations

Bridge Domain (BD)

BD Subnet

- Subnet A
  - ✓ "Advertised Externally"

Associated L3OUT

- Target L3OUT(s) to advertise BD subnets

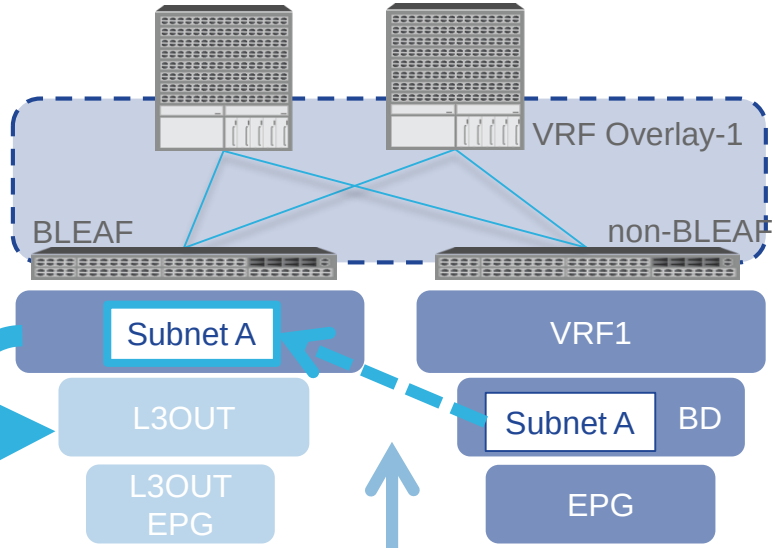
External Routed Networks (L3OUT)

Networks (L3OUT EPG)

- Contract to EPG

Redistribution
Direct (Subnet A)
-> L3OUT Protocol

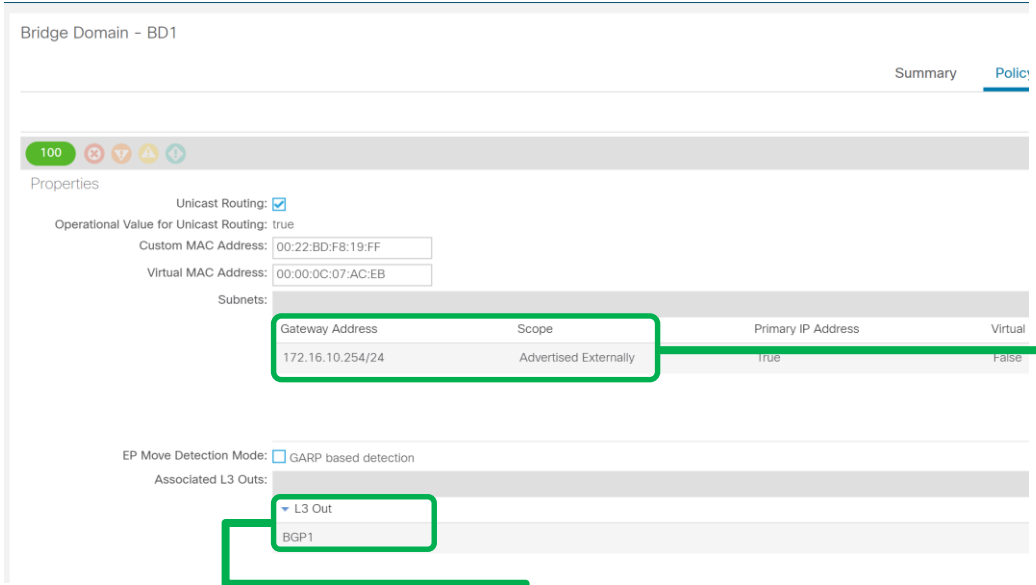
Static Route (subnet A)
on BLEAF via MO (object)



L3OUT Key Components

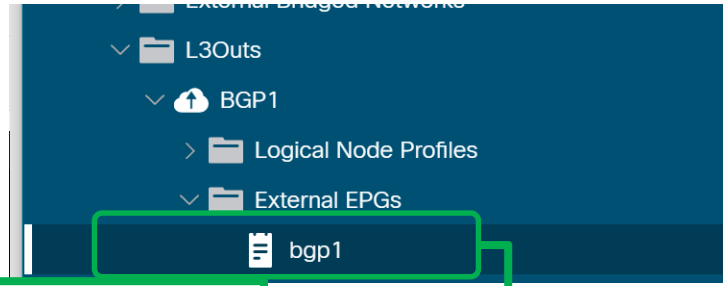
3. Advertise BD subnet

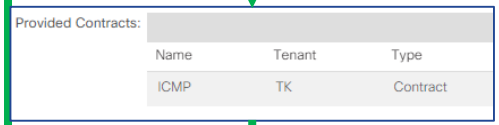
1. L3OUT Association from BD (for redistribution)



2. Contract

Make sure the other end of contract is configured correctly as well






```
→ route-map (eigrp or ospf)
 Redistributing External Routes from
   direct route-map exp-ctx-st-2326530
Or route-map(bgp)
 Outbound route-map configured is exp-l3out-BGP1-peer-2654211, handle obtained
```

```
border-leaf# show ip route vrf TK:VRF1
172.16.0.0/24, ubest/mbest: 1/0, attached, direct, pervasive
    *via 10.0.184.64%overlay-1, [1/0], 04:32:27, static
```

```
ip prefix-list
    ➢ 172.16.10.0/24
```

Check : ip were configured
Peering is configured

```
bdsol-aci32-leaf1# show ip interface vrf RD-BGP:RD
Vlan93, Interface status: protocol-up/link-up/admin-up, iod: 129, mode: external, vrf_vnid: 2654211
  IP address: 172.16.1.13, IP subnet: 172.16.1.12/30
  IP primary address route-preference: 0, tag: 0
loopback5, Interface status: protocol-up/link-up/admin-up, iod: 135, mode: unspecified, vrf_vnid: 2654211
  IP address: 172.16.1.1, IP subnet: 172.16.1.1/32
  IP primary address route-preference: 0, tag: 0
```

```
bdsol-aci32-leaf1# show ip bgp summary vrf RD-BGP:RD
BGP summary information for VRF RD-BGP:RD, address family IPv4 Unicast
BGP router identifier 172.16.1.1, local AS number 132
BGP table version is 71, IPv4 Unicast config peers 1, capable peers 1
12 network entries and 12 paths using 1896 bytes of memory
BGP attribute entries [10/1480], BGP AS path entries [0/0]
BGP community entries [0/0], BGP clusterlist entries [14/136]

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.1.14     4   200     286     287       71    0    0 04:40:02 2
```

BGP is up – what next ?

Check what we receive on BGP session ?

- As soon as BGP is up we have in bgp table
  - Out loopback and peering interface
  - All routes received from bgp peer
  - By default we do not have import route-control (so every routes are accepted, this can be changed using import-route control)

  Note that we do not export anything by default (see later)

```
bdsol-aci32-leaf1#  show ip bgp vrf RD-BGP:RD
BGP routing table information for VRF RD-BGP:RD, address family IPv4 Unicast
BGP table version is 151, local router ID is 172.16.1.1
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-injected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup
*>r172.16.1.1/32      0.0.0.0                  0        100      32768 ?
*>r172.16.1.12/30     0.0.0.0                  0        100      32768
* e0.0.0.0/0          172.16.1.14                       150          0 200 i
*>e172.16.99.0/24     172.16.1.14                       150          0 200 i
```

What do we do with Receive prefix ? iBGP VPNv4

All leaves are automatically configured with the following RD and RT parameter

```
bdsol-aci32-leaf1# show bgp process vrf RD-BGP:RD
BGP Information for VRF RD-BGP:RD
VRF Type                          : System
VRF Id                            : 13
VRF state                         : UP
VRF configured                    : yes
VRF refcount                      : 1
VRF VNID                          : 2654211
Router-ID                         : 172.16.1.1
Configured Router-ID              : 172.16.1.1
Confed-ID                         : 0
Cluster-ID                        : 0.0.0.0
MSITE Cluster-ID                  : 0.0.0.0
No. of configured peers           : 1
No. of pending config peers       : 0
No. of established peers          : 1
VRF RD                            : 101:2654211
VRF EVPN RD                       : 101:2654211
 Export RT list:
        132:2654211
    Import RT list:
        132:2654211
    Label mode: per-prefix
```

Like if we would configure Following in each IOS router Note : RD will differ in each Leaf (more later)

```
vrf RD-BGP:RD
      rd 101:2654211
      route-target 132:2654211 both
```

Received ip BGP path are converted to VPNv4

```
bdsol-aci32-leaf1# show bgp vpnv4 unicast rd 101:2654211 vrf overlay-1
BGP routing table information for VRF overlay-1, address family VPNv4 Unicast
BGP table version is 3292, local router ID is 10.0.88.95
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-injected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup
 Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 101:2654211    (VRF RD-BGP:RD)
* e0.0.0.0/0          172.16.1.14                       150          0 200 i
*>e172.16.99.0/24     172.16.1.14
```

```
bdsol-aci32-leaf1# show bgp vpnv4 unicast rd 101:2654211 172.16.99.0  vrf overlay-1
BGP routing table information for VRF overlay-1, address family VPNv4 Unicast
Route Distinguisher: 101:2654211    (VRF RD-BGP:RD)
BGP routing table entry for 172.16.99.0/24, version 35 dest ptr 0xa6ef26d0
Paths: (1 available, best #1)
Flags: (0x80c001a 00000000) on xmit-list, is in urib, is best urib route, is in HW,
exported
  vpn: version 437, (0x100002) on xmit-list
Multipath: eBGP iBGP

  Advertised path-id 1, VPN AF advertised path-id 1
  Path type: external 0x28 0x0 ref 0 adv path ref 2, path is valid, is best path
  AS-Path: 200 , path sourced external to AS
    172.16.1.14 (metric 0) from 172.16.1.14 (172.16.1.21)
      Origin IGP, MED not set, localpref 150, weight 0
      Extcommunity:
          RT:132:2654211
          VNID:2654211

  VRF advertise information:
  Path-id 1 not advertised to any peer

  VPN AF advertise information:
  Path-id 1 advertised to peers:
    10.0.88.64          10.0.88.94
```

Rx path are Advert
As vpnv4 to spine
With our Route-target

Server leaf received them and insert in RIB

```
bdsol-aci32-leaf4#  show bgp vpnv4 unicast rd 101:2654211   vrf overlay-1
* i172.16.99.0/24     10.0.88.95                      150          0 200 i
*>i                   10.0.88.95                      150          0 200 i
bdsol-aci32-leaf4#  show bgp vpnv4 unicast rd 104:2654211   vrf overlay-1
*>i172.16.99.0/24     10.0.88.95                      150          0 200 i
```

bgp path are Rx with Bgp next-hop PTEP of BL 10.0.88.95

```
bdsol-aci32-leaf4#  show bgp vpnv4 unicast rd 104:2654211  172.16.99.0  vrf overlay-1
BGP routing table information for VRF overlay-1, address family VPNv4 Unicast
Route Distinguisher: 104:2654211    (VRF RD-BGP:RD)
BGP routing table entry for 172.16.99.0/24, version 64 dest ptr 0xa6da68d0
Paths: (1 available, best #1)
Flags: (0x08001a 00000000) on xmit-list, is in urib, is best urib route, is in HW
  vpn: version 1341, (0x100002) on xmit-list
Multipath: eBGP iBGP

  Advertised path-id 1, VPN AF advertised path-id 1
  Path type: internal 0xc0000018 0x40 ref 56506 adv path ref 2, path is valid, is
best path
              Imported from 101:2654211:172.16.99.0/24
  AS-Path: 200 , path sourced external to AS
    10.0.88.95 (metric 3) from 10.0.88.64 (10.10.32.214)
      Origin IGP, MED not set, localpref 150, weight 0
      Received label 0
      Received path-id 1
      Extcommunity:
          RT:132:2654211
          VNID:2654211
      Originator: 10.0.88.95 Cluster list: 10.10.32.214

 RF advertise information:
 ath-id 1 not advertised to any peer
```

Route is inserted in RIB

```
bdsol-aci32-leaf4# show ip route 172.16.99.0 vrf RD-BGP:RD
IP Route Table for VRF "RD-BGP:RD"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

172.16.99.0/24, ubest/mbest: 1/0
    *via 10.0.88.95%overlay-1, [200/0], 02:05:07, bgp-132,
internal, tag 200 (mpls-vpn)
```

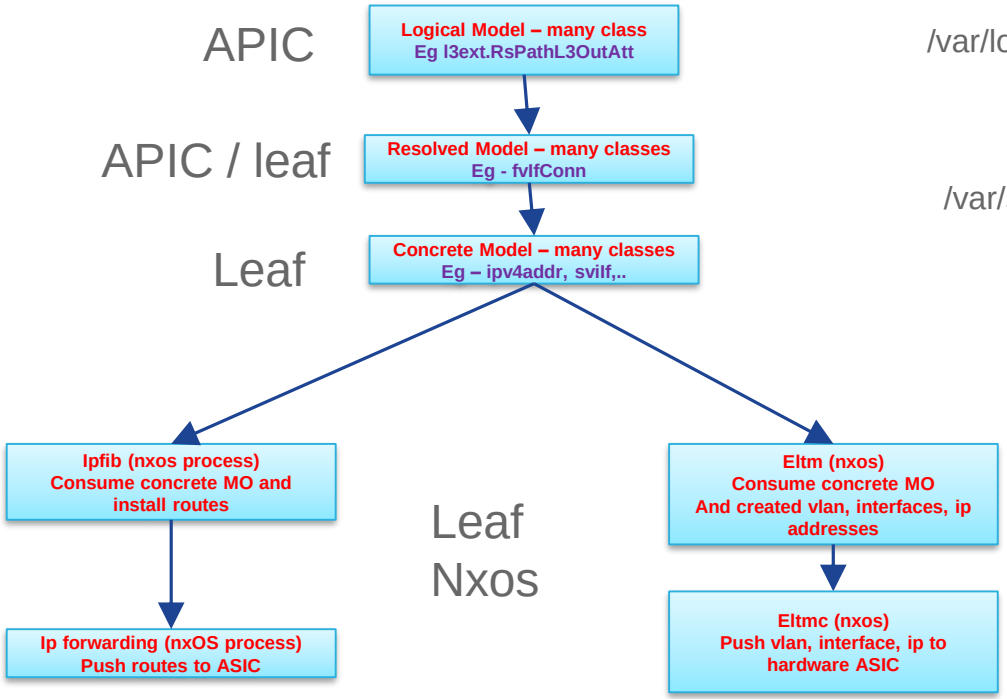
Check BD subnet advertisement

```
bdsol-aci32-leaf1# show ip bgp neighbors 172.16.1.14 vrf RD-BGP:RD | egrep "Out.*route-map"
  Outbound route-map configured is exp-l3out-BGP1-peer-2654211, handle obtained


bdsol-aci32-leaf1# show route-map exp-l3out-BGP1-peer-2654211
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15804
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 0


bdsol-aci32-leaf1# show ip prefix-list IPv4-peer16387-2654211-exc-int-inferred-export-dst
ip prefix-list IPv4-peer16387-2654211-exc-int-inferred-export-dst: 1 entries
   seq 1 permit 172.16.10.0/24
```

Does BGP actually send out the BD subnet

```
bdsol-aci32-leaf1# show ip bgp neighbors 172.16.1.14 advertise  vrf RD-BGP:RD

Peer 172.16.1.14 routes for address family IPv4 Unicast:
BGP table version is 162, local router ID is 172.16.1.1
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-injected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup

   Network            Next Hop            Metric     LocPrf     Weight Path
*>r172.16.10.0/24     0.0.0.0                  0        100      32768 ?

bdsol-aci32-leaf1# show ip bgp 172.16.10.0  vrf RD-BGP:RD
BGP routing table information for VRF RD-BGP:RD, address family IPv4 Unicast
BGP routing table entry for 172.16.10.0/24, version 152 dest ptr 0xa6eef550
Paths: (1 available, best #1)
Flags: (0x80c0002 00000000) on xmit-list, is not in urib, exported
  vpn: version 3293, (0x100002) on xmit-list
Multipath: eBGP iBGP

  Advertised path-id 1, VPN AF advertised path-id 1
  Path type: redist 0x408 0x404001 ref 0 adv path ref 2, path is valid, is best path
  AS-Path: NONE, path locally originated
    0.0.0.0 (metric 0) from 0.0.0.0 (172.16.1.1)
      Origin incomplete, MED 0, localpref 100, weight 32768
      Extcommunity:
          RT:132:2654211
          VNID:2654211

  VRF advertise information:
  Path-id 1 advertised to peers:
    172.16.1.14
```

Troubleshooting – Object Model
what if vlan or ip or routing table is not deployed on leaf ?



/var/log/dme/log/svc_ifc_policymgr.*

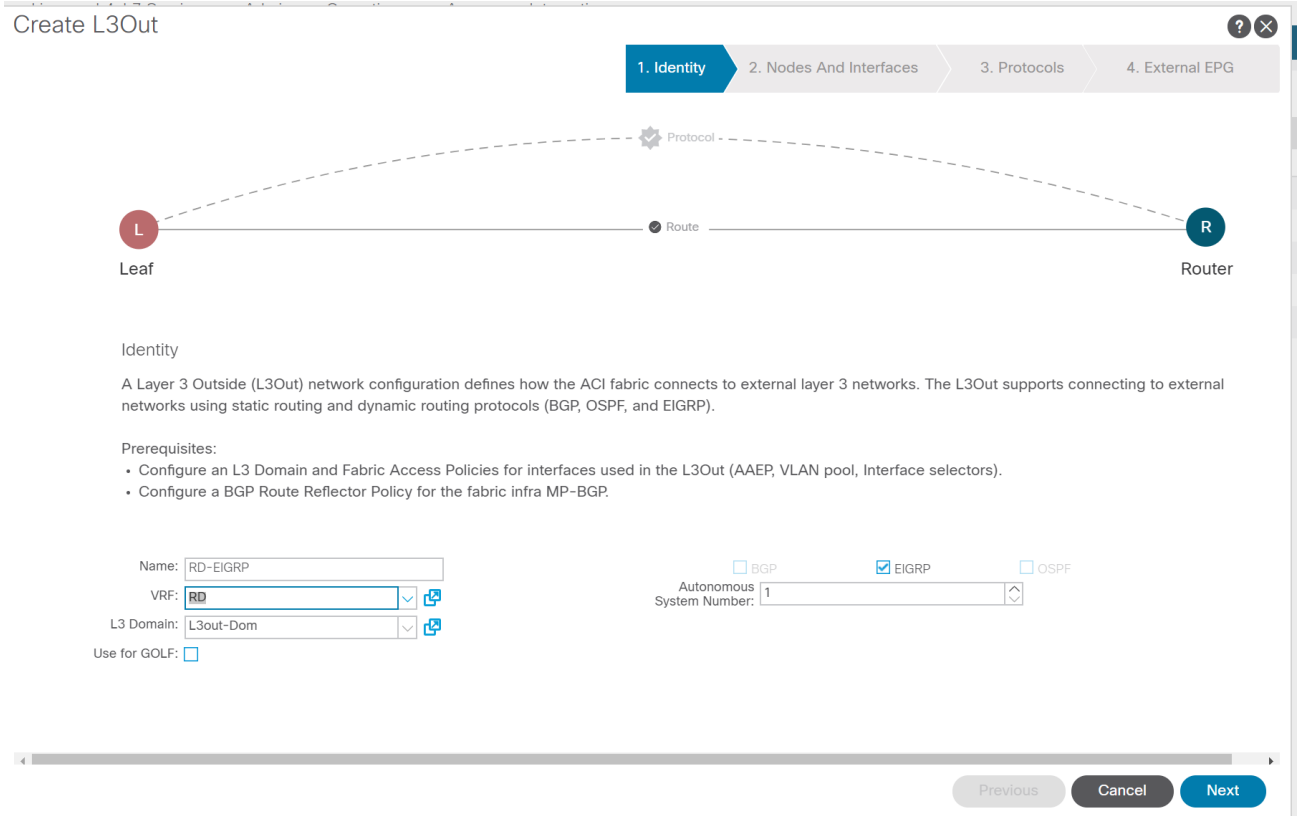
/var/sysmgr/tmp_logs/svc_ifc_policyelem.*

/var/sysmgr/tmp_logs/eltm*

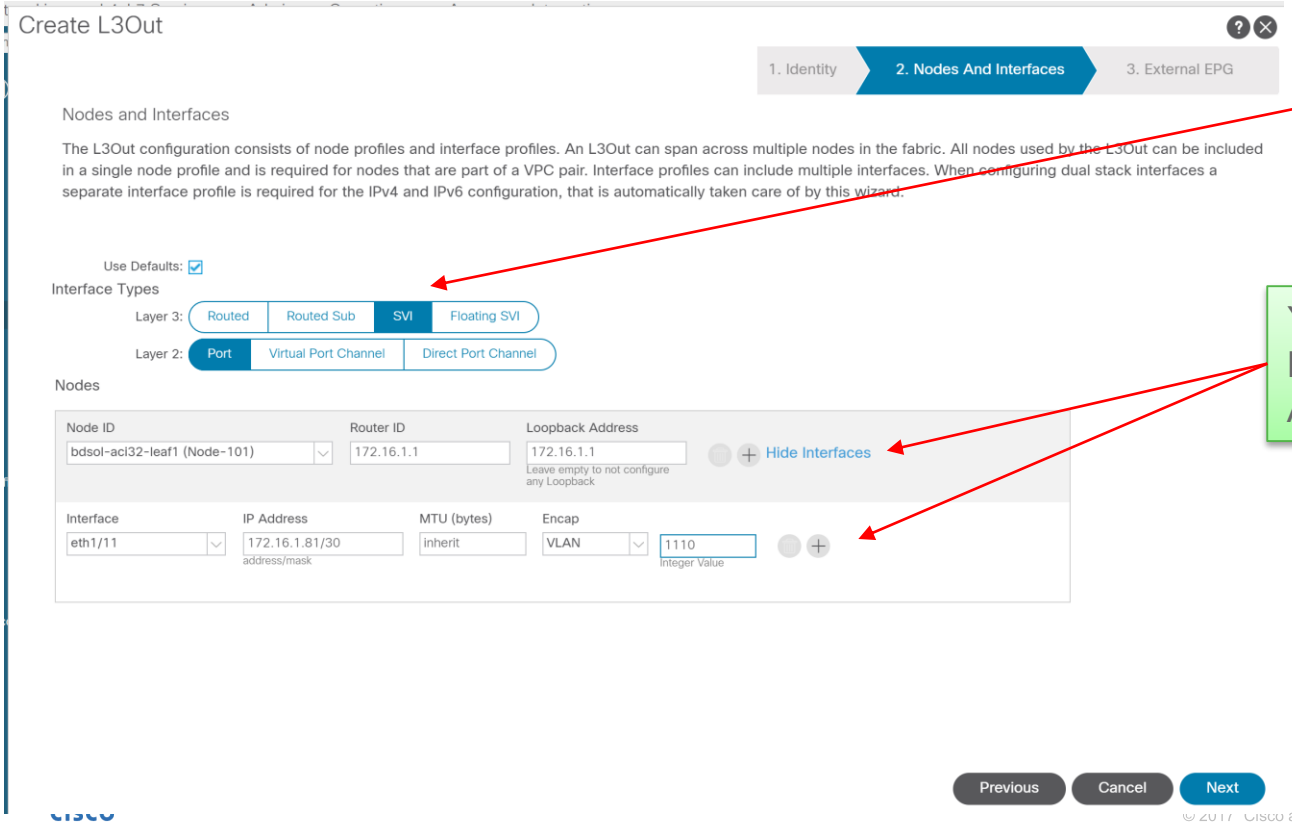
Vsh – show ipmgr internal traces
Vsh_lc – show forwarding traces module 1

L3 out config Made Easy Wizard in 4.2 EIGRP example

L3 out Wizard (Revamped in 4.2) – Step 1



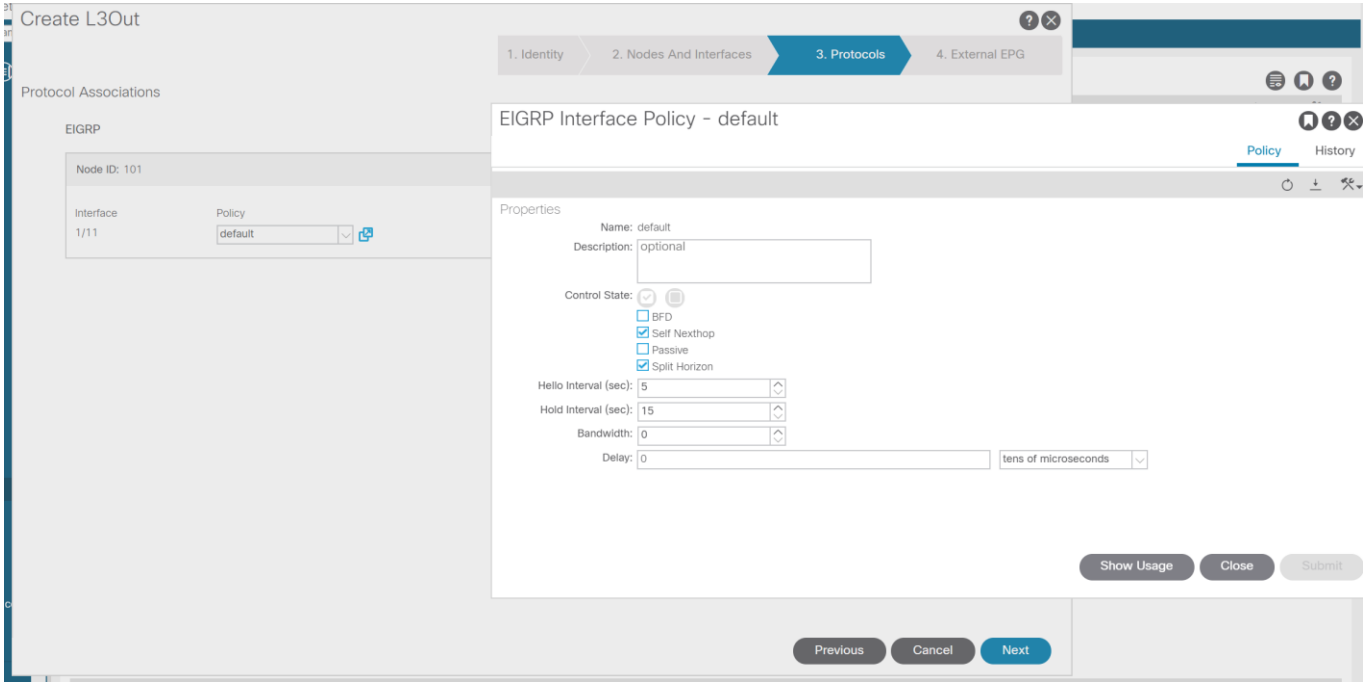
L3 out Wizard (Revamped in 4.2) – Step 2



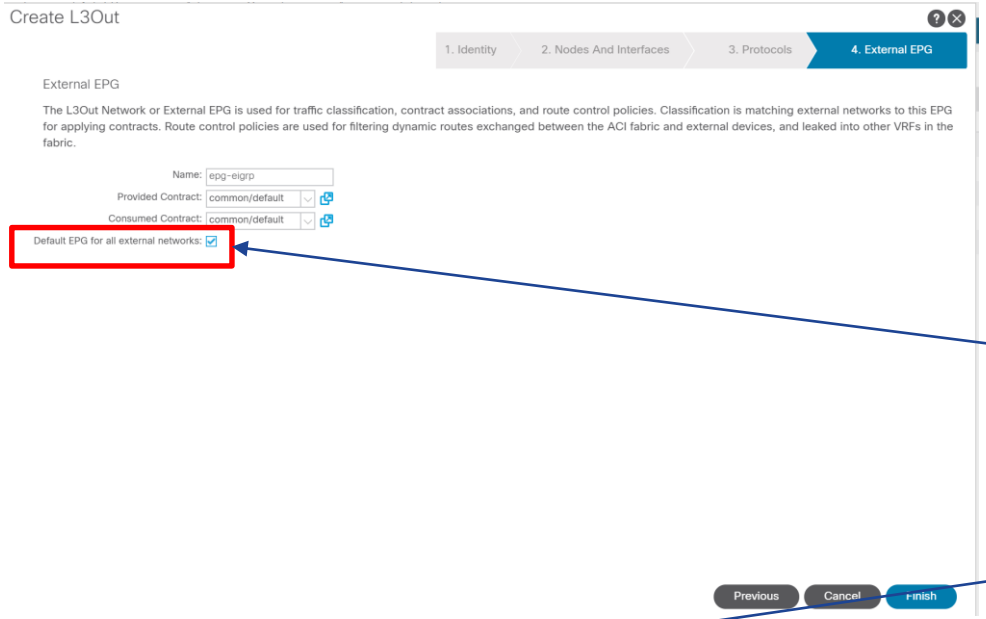
Select your type of L3 interface and l2 path

You can add more Node (BL) And/or more If per BL

Step 3 – protocol policies



L3 out Wizard (Revamped in 4.2) – Step 4



If you unclick use Default EPG for ext Network. You will be Able configure subnet One by one

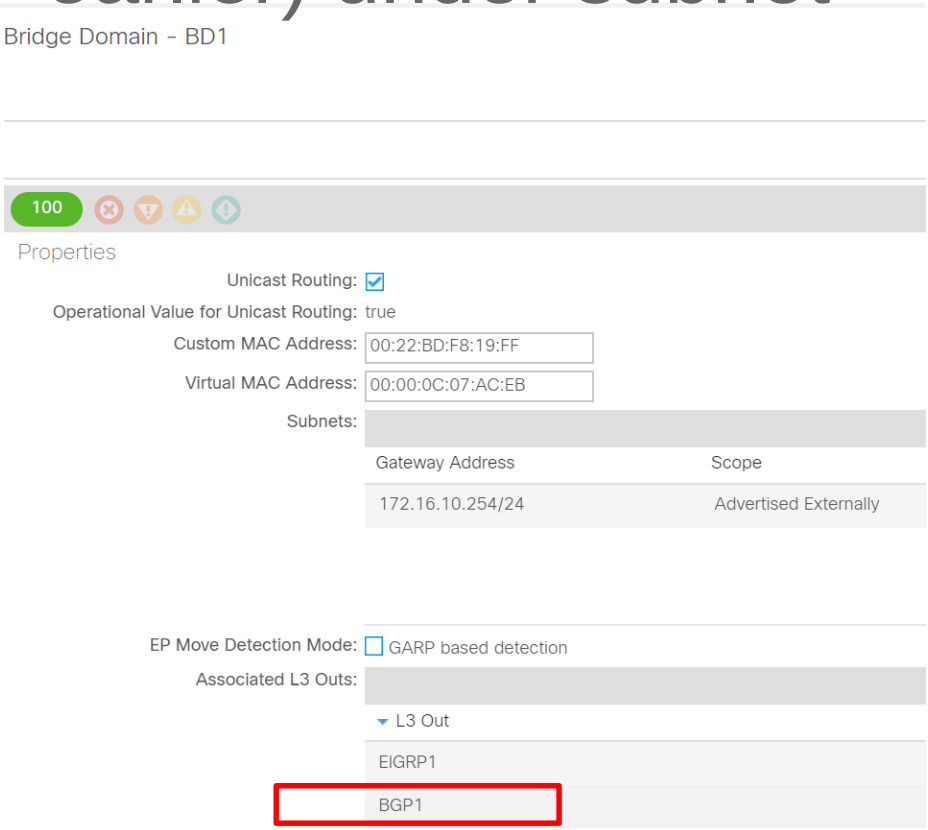
Consumed Contract: common/default

Default EPG for all external networks:

Subnets

IP Address	Scope	Name	Aggregate	Route Control Profile	Route Summarization Policy

Advertise BD (similar to BGP example earlier) under subnet



EIGRP basic check

```
bdsol-aci32-leaf1# show ip eigrp vrf RD-BGP:RD
IP-EIGRP AS 1 ID 172.16.1.1 VRF RD-BGP:RD
  Process-tag: default
  Instance Number: 1
  Status: running
  Authentication mode: none
  Authentication key-chain: none
  Metric weights: K1=1 K2=0 K3=1 K4=0 K5=0
  metric version: 32bit
  IP proto: 88 Multicast group: 224.0.0.10
  Int distance: 90 Ext distance: 170
  Max paths: 8
  Active Interval: 3 minute(s)
  Number of EIGRP interfaces: 2 (1 loopbacks)
  Number of EIGRP passive interfaces: 0
  Number of EIGRP peers: 1
  Redistributing:
    static route-map exp-ctx-st-2654211
    ospf-default route-map exp-ctx-proto-2654211
    direct route-map exp-ctx-st-2654211
    coop route-map exp-ctx-st-2654211
    bgp-132 route-map exp-ctx-proto-2654211
  Tablemap: route-map exp-ctx-2654211-deny-external-tag , filter-configured
  Graceful-Restart: Enabled
  Stub-Routing: Disabled
  NSF converge time limit/expiries: 120/0
  NSF route-hold time limit/expiries: 240/0
```

Check
Eigrp process is running
And has some interface running eigrp

Note as well all Route-map used
To redistribute to eigrp

EIGRP interface check

```
bdsol-aci32-leaf1# show ip eigrp  interface vrf RD-BGP:RD
IP-EIGRP interfaces for process 1 VRF RD-BGP:RD

                        Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Lo11               0        0/0         0       0/0          0            0
  Hello interval is 5 sec
  Holdtime interval is 15 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/0  Un/reliable ucasts: 0/0
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 0
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is not set
  Use multicast
  Classic/wide metric peers: 0/0
Vlan129            1        0/0         1       0/0          50           0
  Hello interval is 5 sec
  Holdtime interval is 15 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/3  Un/reliable ucasts: 2/2
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 1
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is not set
  Use multicast
  Classic/wide metric peers: 1/0
```

Check
The interface runs eigrp
(at least the l3 out logical interface and the loopback)

Checking EIGRP neighbor

```
bdsol-aci32-leaf1# show ip eigrp  neigh det vrf RD-BGP:RD
IP-EIGRP neighbors for process 1 VRF RD-BGP:RD
H   Address                 Interface       Hold  Uptime   SRTT   RTO  Q  Seq
                                            (sec)          (ms)       Cnt Num
0   172.16.1.82             Vlan129         12   00:26:44  1    50    0   3
   Version 8.0/1.2, Retrans: 0, Retries: 0, BFD state: N/A, Prefixes: 1
```

Check if we have an EIGRP neighbor

EIGRP sending BD subnet

```
bdsol-aci32-leaf1# show ip eigrp vrf RD-BGP:RD | egrep route-ma
    static route-map exp-ctx-st-2654211
    ospf-default route-map exp-ctx-proto-2654211
    direct route-map exp-ctx-st-2654211
    coop route-map exp-ctx-st-2654211
    bgp-132 route-map exp-ctx-proto-2654211
  Tablemap: route-map exp-ctx-2654211-deny-external-tag , filter-configured
bdsol-aci32-leaf1# show route-map exp-ctx-st-2654211
..
route-map exp-ctx-st-2654211, permit, sequence 15804
  Match clauses:
    ip address prefix-lists: IPv4-st16387-2654211-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 0
..
bdsol-aci32-leaf1# show ip prefix-list IPv4-st16387-2654211-exc-int-inferred-export-dst
ip prefix-list IPv4-st16387-2654211-exc-int-inferred-export-dst: 1 entries
   seq 1 permit 172.16.10.0/24
```

EIGRP BD subnet in EIGRP topology DB

```
bdsol-aci32-leaf1# show ip eigrp topology  172.16.10.0/24 vrf RD-BGP:RD

IP-EIGRP (AS 1): Topology entry for 172.16.10.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 51200
  Routing Descriptor Blocks:
  0.0.0.0, from Rconnected, Send flag is 0x0
      Composite metric is (51200/0), Route is External
      Vector metric:
        Minimum bandwidth is 100000 Kbit
        Total delay is 1000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1492
        Hop count is 0
        Internal tag is 0
      External data:
        Originating router is 172.16.1.1 (this system)
        AS number of route is 0
        External protocol is Static, external metric is 0
        Administrator tag is 0 (0x00000000)
```

L3 out Cli check for OSPF

OSPF basic

Check OSPF is running verify
Area matching, mtu, Network type (P2P or broadcast)

```
bdsol-aci32-leaf6# show ip ospf interface vrf RD-BGP:RD
 loopback3 is up, line protocol is up
    IP address 172.16.1.6/32
Process ID default VRF RD-BGP:RD, area 0.0.0.1
    Enabled by interface configuration
    State LOOPBACK, Network type LOOPBACK, cost 1
    Index 3
 Vlan70 is up, line protocol is up
    IP address 172.16.1.33/30
Process ID default VRF RD-BGP:RD, area 0.0.0.1
    Enabled by interface configuration
    State P2P, Network type P2P, cost 4
    Index 4, Transmit delay 1 sec
    1 Neighbors, flooding to 1, adjacent with 1
    Timer intervals: Hello 10, Dead 40, Wait 40, Retransmit 5
      Hello timer due in 00:00:04
    No authentication
    Number of opaque link LSAs: 0, checksum sum 0
bdsol-aci32-leaf6# show ip ospf neigh vrf RD-BGP:RD
 OSPF Process ID default VRF RD-BGP:RD
 Total number of neighbors: 1
 Neighbor ID     Pri State            Up Time  Address         Interface
 172.16.1.26       1 FULL/ -          10w1d    172.16.1.34     Vlan70
bdsol-aci32-leaf6#
```

Checking BD subnet advert in OSPF

```
bdsol-aci32-leaf6# show ip ospf vrf RD-BGP:RD

 Routing Process default with ID 172.16.1.6 VRF RD-BGP:RD
 Stateful High Availability enabled
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Table-map using route-map exp-ctx-2654211-deny-external-tag
 Redistributing External Routes from
   static route-map exp-ctx-st-2654211
   direct route-map exp-ctx-st-2654211
   bgp route-map exp-ctx-proto-2654211
   eigrp route-map exp-ctx-proto-2654211
   coop route-map exp-ctx-st-2654211
..
bdsol-aci32-leaf6# show route-map exp-ctx-st-2654211
..
route-map exp-ctx-st-2654211, permit, sequence 15803
  Match clauses:
    ip address prefix-lists: IPv4-st32770-2654211-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 0
bdsol-aci32-leaf6# show ip prefix-list IPv4-st32770-2654211-exc-int-inferred-export-dst
ip prefix-list IPv4-st32770-2654211-exc-int-inferred-export-dst: 1 entries
   seq 1 permit 172.16.10.254/24
bdsol-aci32-leaf6#
```

BD subnet in OSPF DB

- Appears as LSA type 5 with Tag 0 (redistributed from static to OSPF

```
bdsol-aci32-leaf2# show ip ospf database external 172.16.10.0  vrf RD-BGP:RD
        OSPF Router with ID (172.16.1.2) (Process ID default VRF RD-BGP:RD)

                Type-5 AS External Link States

Link ID         ADV Router      Age         Seq#       Checksum Tag
172.16.10.0     172.16.1.2      18          0x80000002 0x1027    0
```

Troubleshooting Routing protocol behavior

TCPDUMP

- RP traffic is targeted to cpu you can always use tcpdump to see what you receive (on kpm_inb)
  - `bdsol-aci32-leaf1# tcpdump -ni kpm_inb proto eigrp`
  - `bdsol-aci32-leaf1# tcpdump -ni kpm_inb proto ospf`
  - `bdsol-aci32-leaf1# tcpdump -ni kpm_inb -f port 179`
- You can add extra filter such as :
  - `bdsol-aci32-leaf1# tcpdump -ni kpm_inb -f port 179 and host 1.1.1.1`
- Or get more verbose :
  - `bdsol-aci32-leaf1# tcpdump -nxxvvi kpm_inb -f port 179 and host 1.1.1.1`

Tcpdump example

```
bdsol-aci32-leaf1# tcpdump -nxxvvi kpm_inb -f port 179 and host 1.1.1.1
tcpdump: listening on kpm_inb, link-type EN10MB (Ethernet), capture size 65535 bytes
15:23:56.623184 IP (tos 0xc0, ttl 64, id 8202, offset 0, flags [none], proto TCP (6), length 52)
    1.1.1.1.44547 > 1.1.1.2.179: Flags [.], cksum 0x0ddb (correct), seq 3740755854, ack 272594056, win
14600, options [nop,nop,TS val 2364232682 ecr 154912704], length 0
        0x0000:  0022 bdf8 19ff 00fe c862 b941 0800 45c0
        0x0010:  0034 200a 0000 4006 55f6 0101 0101 0101
        0x0020:  0102 ae03 00b3 def7 678e 103f 7488 8010
        0x0030:  3908 0ddb 0000 0101 080a 8ceb 53ea 093b
        0x0040:  c7c0
```

Protocol Log

- Highly depend on the software running on the fabric
  - Either event-history or trace show command in old soft (in vsh)
- Or plain file with linux file rotation (3.1+)
- Or linux file binary encoded (4.2)

Nxos event or trace (vsh only)

```
show bgp event-history detail

show ip ospf event-history adjacency
show ip ospf event-history event
show ip ospf event-history lsa
show ip ospf event-history spf
show ip ospf event-history redistribution
show ip ospf event-history ldp
show ip ospf event-history te
show ip ospf event-history rib
show ip ospf event-history hello

show ip eigrp event-history fsm
show ip eigrp event-history packet
show ip eigrp event-history rib
show ip eigrp event-history bfd
```

Plain text debug in linux file (3.1 to 4.1)

```
bdsol-aci32-leaf6# ls -al | egrep  "bgp.*txt"
-rw-rw-rw-  1 root root        3325952 Sep 12 15:02 bgp_trace.txt
bdsol-aci32-leaf6# ls -al | egrep  "osp.*txt"
-rw-rw-rw-  1 root root       34549233 Sep 12 15:31 ospfv2_1_trace.txt
-rw-rw-rw-  1 root root       56651776 Sep 12 15:31 ospfv2_2_trace.txt
-rw-rw-rw-  1 root root          47265 Sep 11 11:47 ospfv3_1_trace.txt
```

```
Running log in :
/var/sysmgr/tmp_logs

Old log (gzip):
/var/log/dme/oldlog
```

4.2 and above

- Bgp, eigrp, isis, ospf and some other protocol traces are binary encoded.
- File end up with .bl
- All can be decoded using : “log_trace_bl_print_tool” <file name>

```
bdsol-aci32-leaf1# ls -al *.bl
-rw-rw-rw- 1 root root 43439057 Oct 19 10:53 bgp_trace.bl
-rw-rw-rw- 1 root root 39618433 Oct 19 10:53 coop_trace.bl
-rw-rw-rw- 1 root root 59710790 Oct 19 10:53 isis_trace.bl
-rw-rw-rw- 1 root root 37771710 Oct 19 10:53 ospfv2_1_trace.bl
-rw-rw-rw- 1 root root     6671 Oct  3 14:38 ospfv2_2_trace.bl
-rw-rw-rw- 1 root root     2666 Sep 22 13:20 ospfv3_1_trace.bl
-rw-rw-rw- 1 root root   374065 Oct 19 10:53 rpm_trace.bl

bdsol-aci32-leaf1# log_trace_bl_print_tool bgp_trace.bl | more
version: 1, pid: 60215
[2019 Sep 11 07:13:23.632899106:main:4257] (0) OBJ: kcache lib initialized succesfully in BGP
[2019 Sep 11 07:13:23.634434168:main:4298] BGP process bgp-132 startup, reason: configuration
```

Import Route-Control (BGP and OSPF)

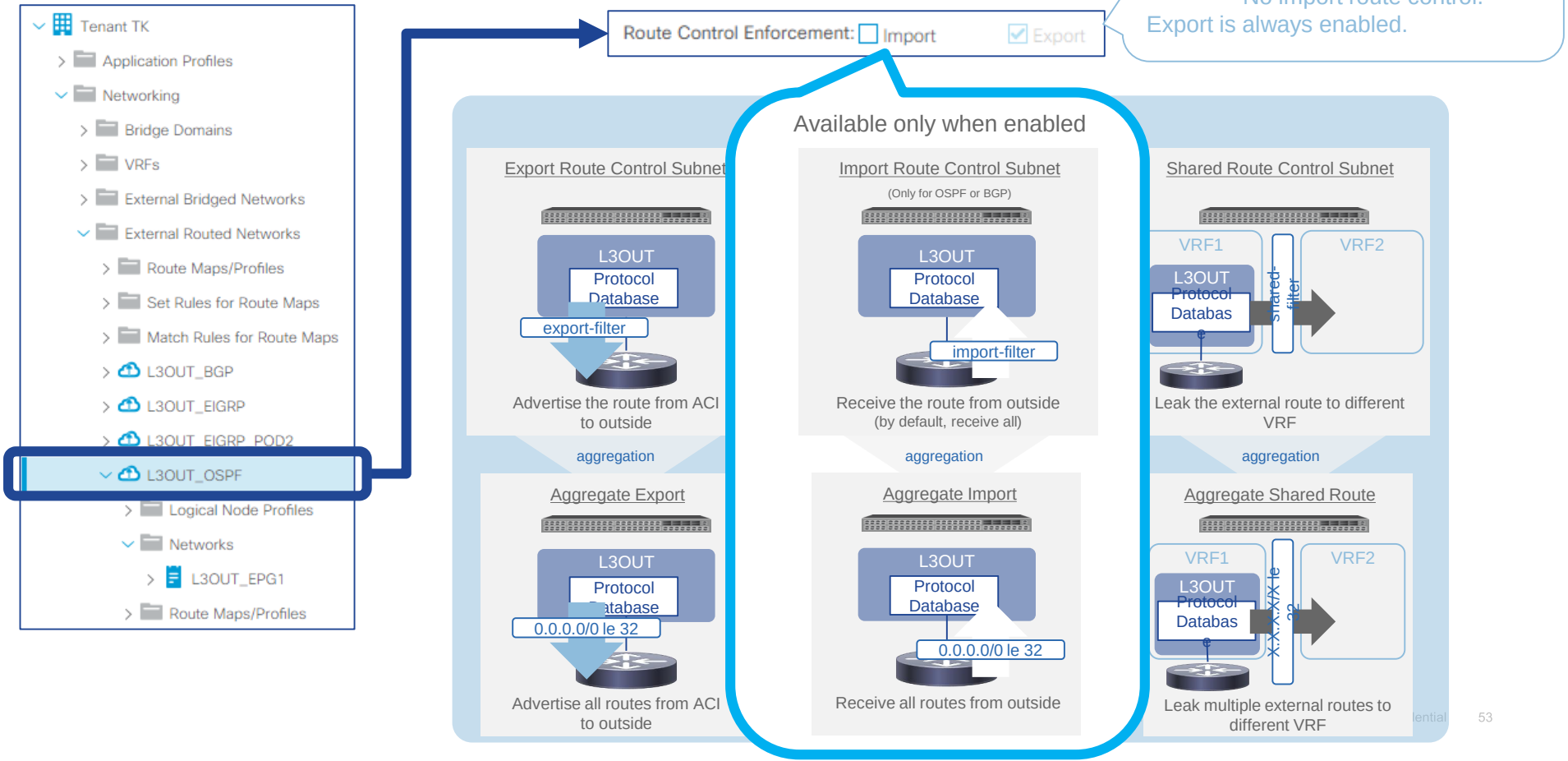
Import route-control

- By default there is no import route-control
- Import route-control can be enabled for OSPF and BGP L3 out Not for EIGRP.
- Enabling it is done in global L3 out screen
- Once enable, implicit import deny is set and user must configure import subnet

Import Route Control Enforcement

Import is disabled by default.
- Receive all routes by default. No import route control.

Export is always enabled.



BGP import route-control

- When import route-control not enable, there is no inbound route-map
- We enable import route-control on BGP layer 3 out.
- Route-map is applied but do not exist

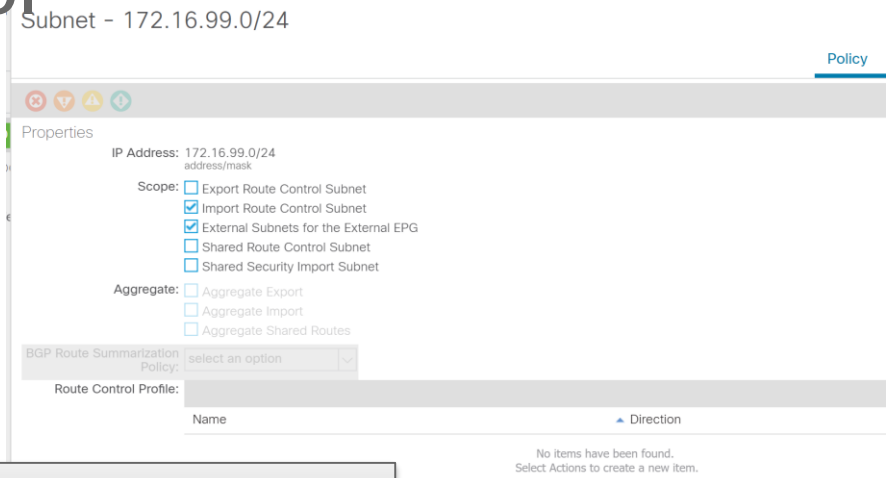
```
bdsol-aci32-leaf1# show ip bgp neighbors  172.16.1.14 vrf RD-BGP:RD | egrep route-map
  Inbound route-map configured is permit-all, handle obtained
```


```
bdsol-aci32-leaf1# show ip bgp neighbors  172.16.1.14 vrf RD-BGP:RD | egrep route-map
  Inbound route-map configured is imp-l3out-BGP1-peer-2654211, handle obtained
```

```
bdsol-aci32-leaf1# show route-map imp-l3out-BGP1-peer-2654211
% Policy imp-l3out-BGP1-peer-2654211 not found
```

BGP import route-control

- If we had some subnet as import route-control subnet under that l3 out the route-map gets created with a permit statement
- Prefix-list contains this subnet




```
bdsol-aci32-leaf1# show ip bgp neighbors  172.16.1.14 vrf RD-BGP:RD | egrep route-map
  Inbound route-map configured is imp-l3out-BGP1-peer-2654211, handle obtained
  Outbound route-map configured is exp-l3out-BGP1-peer-2654211, handle obtained
```

```
bdsol-aci32-leaf1# show route-map imp-l3out-BGP1-peer-2654211
route-map imp-l3out-BGP1-peer-2654211, permit, sequence 15801
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-ext-inferred-import-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
```

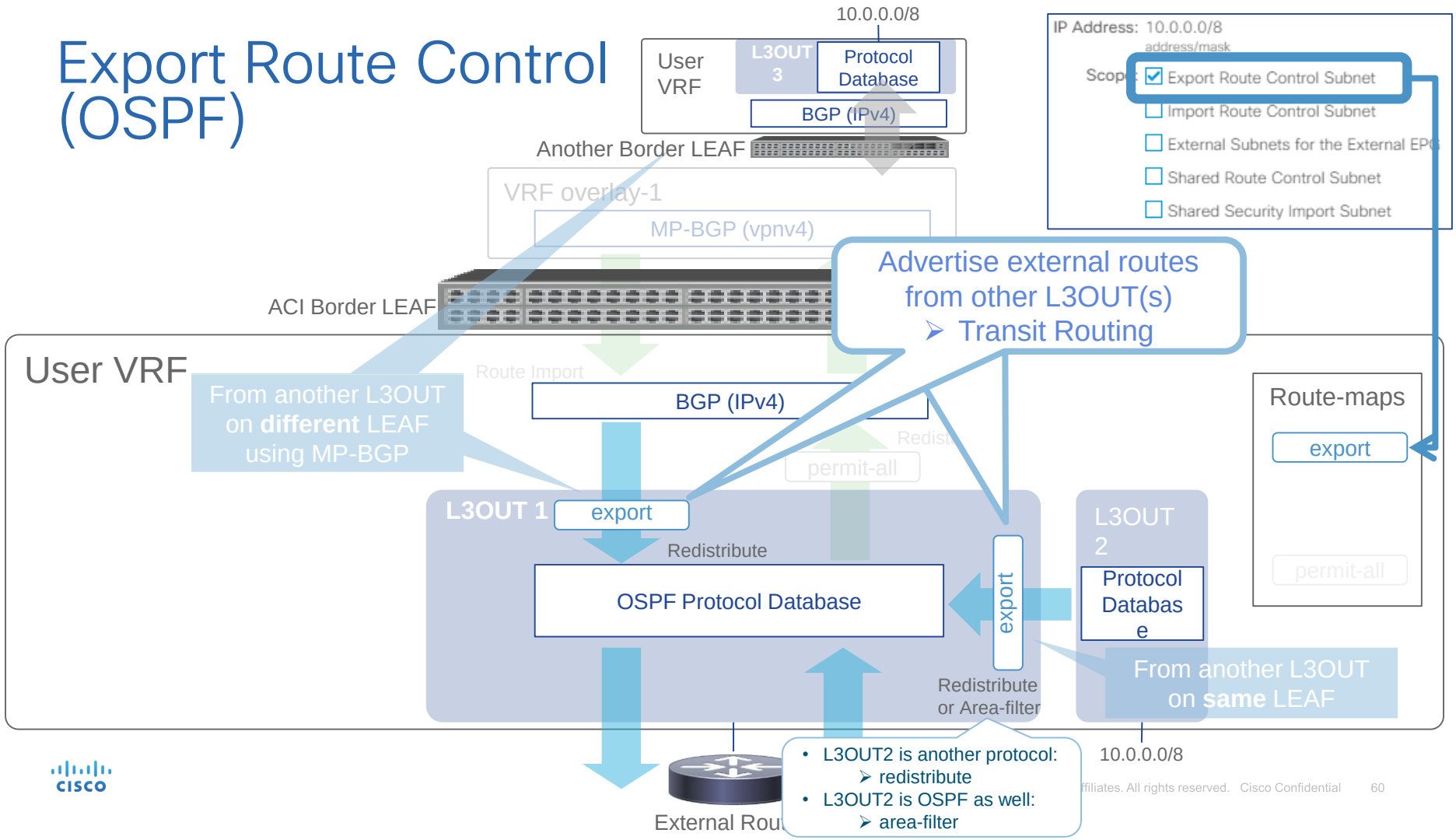
```
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer16387-2654211-exc-ext-inferred-import-dst
ip prefix-list IPv4-peer16387-2654211-exc-ext-inferred-import-dst: 1 entries
   seq 1 permit 172.16.99.0/24
```

Export Route-Control Transit Routing

Export Route Control Subnet

- Primary use is for transit routing
  - Aka Layer 3 out 1 routes to be transmitted on Layer 3 out 2 in same VRF (on same leaf or different leaf)
- This can also be used as an alternative way to advertise BD subnet to a layer 3

Export Route Control (OSPF)



CLI

1. OSPF/EIGRP Redistribution route-map

```
border-leaf# show ip ospf vrf TK:VRF1
 Redistributing External Routes from
   static route-map exp-ctx-st-2097152
   direct route-map exp-ctx-st-2097152
   bgp route-map exp-ctx-proto-2097152
   eigrp route-map exp-ctx-proto-2097152
   Area (backbone)
        Area-filter in 'exp-ctx-proto-2097152'
```

It shares the same route-map with other protocols in the same VRF on the same LEAF

route-map naming:
exp-ctx-st-<vrf vnid> or
exp-ctx-proto-<vrf vnid>

```
border-leaf# show ip eigrp vrf TK:VRF1
  Redistributing:
    static route-map exp-ctx-st-2097152
    ospf-default route-map exp-ctx-proto-2097152
    direct route-map exp-ctx-st-2097152
    bgp-65003 route-map exp-ctx-proto-2097152
```

EIGRP doesn't support Transit Routing on a same LEAF.
- No equivalent filter like OSPF area-filter in EIGRP

2. route-map and ip prefix-list

```
border-leaf# show route-map exp-ctx-proto-2097152
route-map exp-ctx-proto-2097152, permit, sequence 15801
  Match clauses:
    ip address prefix-lists: IPv4-proto49158-2097152-exc-ext-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
```

All Export Route Control subnet on a same LEAF is added here

Same goes to exp-cxt-st-2097152

```
border-leaf# show ip prefix-list IPv4-proto49158-2097152-exc-ext-inferred-export-dst
ip prefix-list IPv4-proto49158-2097152-exc-ext-inferred-export-dst: 1 entries
   seq 1 permit 10.0.0.0/8
```

cisco

CLI

1. BGP outbound route-map

```
border-leaf# show ip bgp neighbors vrf TK:VRF1
BGP neighbor is 17.0.0.1,  remote AS 65001, ebgp link,  Peer index 1
  Outbound route-map configured is exp-l3out-L3OUT_BGP-peer-2097152, handle obtained
```

BGP has a route-map per L3OUT
- A bit more granular control

route-map naming:
exp-l3out-<bgp l3out name>-peer-<vrf vnid>

2. route-map and ip prefix-list

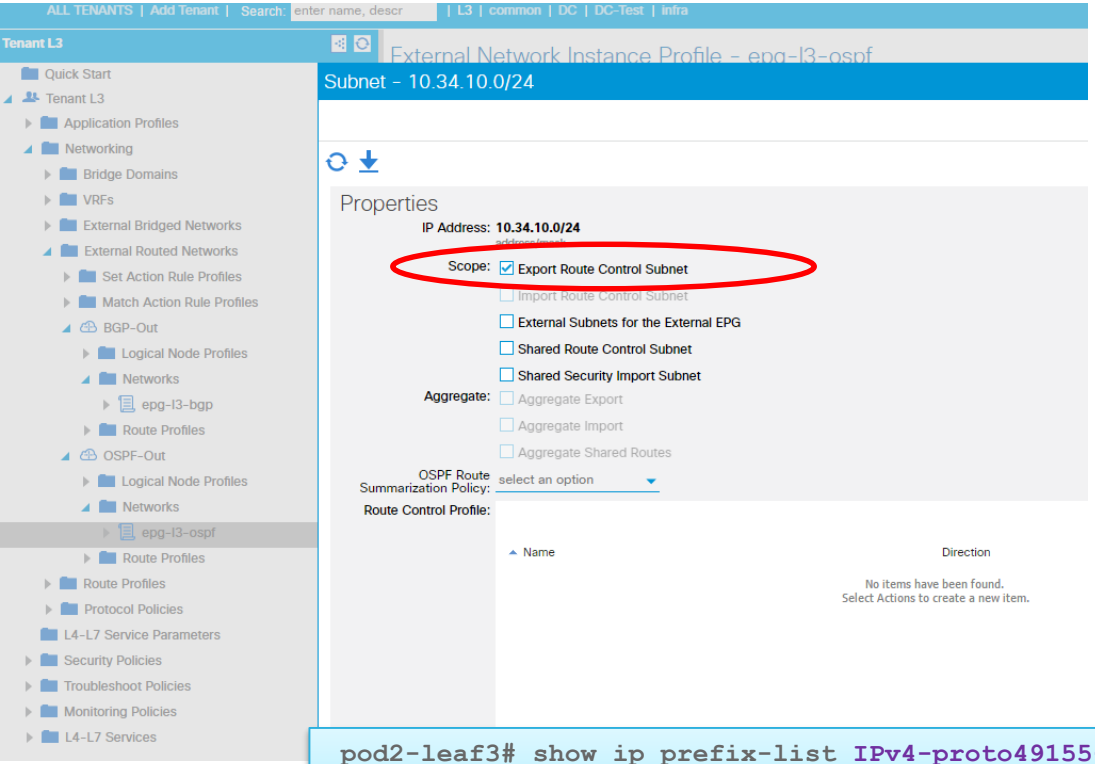
```
border-leaf# show route-map exp-l3out-L3OUT_BGP-peer-2097152
route-map exp-l3out-L3OUT_BGP-peer-2097152, permit, sequence 15804
  Match clauses:
    ip address prefix-lists: IPv4-peer49157-2097152-exc-ext-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
route-map exp-l3out-L3OUT_BGP-peer-2097152, deny, sequence 16000
  Match clauses:
    route-type: direct
  Set clauses:
```

All Export Route Control subnets from the same BGP L3OUT is added here

```
border-leaf# show ip prefix-list IPv4-peer49157-2097152-exc-ext-inferred-export-dst
ip prefix-list IPv4-peer49157-2097152-exc-ext-inferred-export-dst: 4 entries
   seq 1 permit 10.0.0.0/8
```

Transit routing – aggregate route

Aggregate route ?



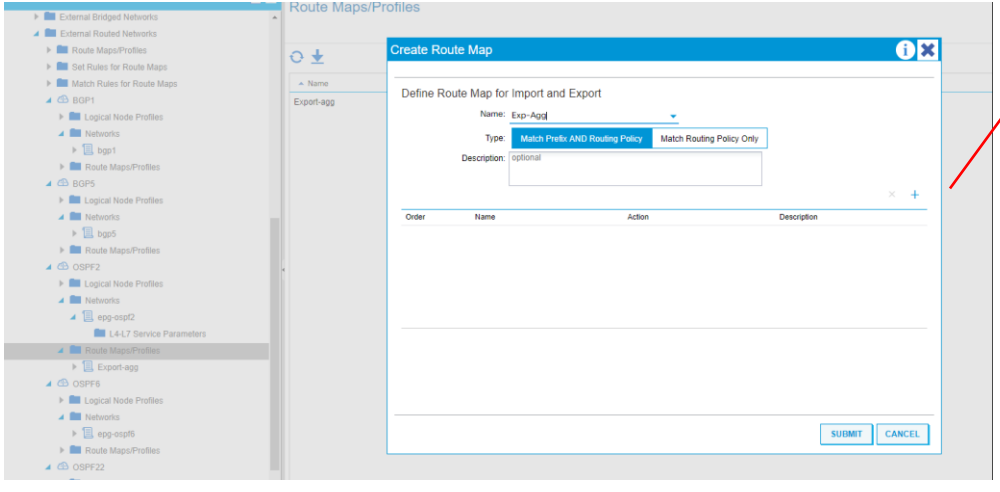
When you mark a subnet as Export Route Control Subnet , see the Aggregate Export is greyed out

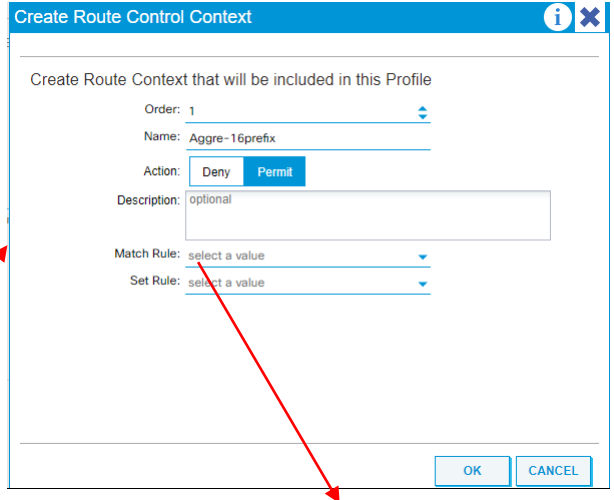
The prefix list is an exact math we do not have the le 32 to match everythinbg between the prefix (/24 and 32)

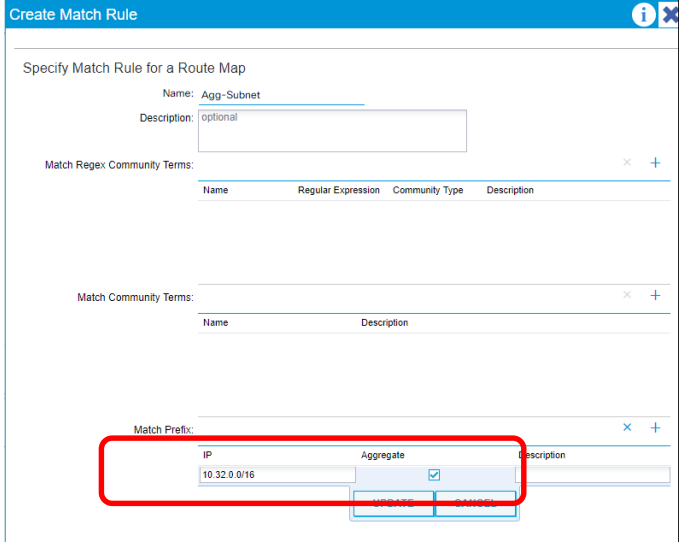
```
pod2-leaf3# show ip prefix-list IPv4-proto49155-2785280-exc-ext-inferred-export-dst
ip prefix-list IPv4-proto49155-2785280-exc-ext-inferred-export-dst: 2 entries
   seq 3 permit 10.34.10.0/24
```

Step 1-

- Configure a route map in the layer 3 to mark the supernet we want as aggregate
- Create new route map
- Create a new sequence in it
- Create the match rule for the supernet
- No need to create set rules

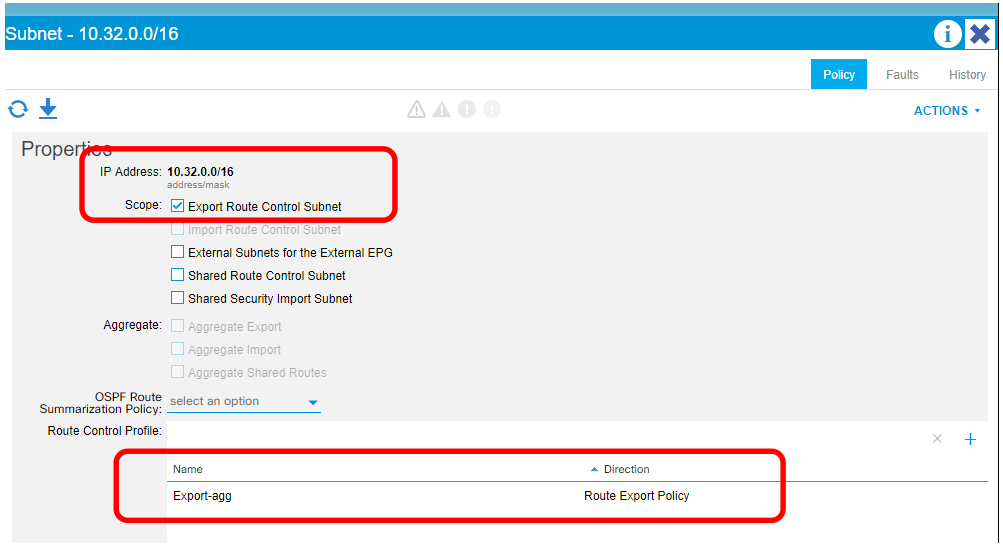

Route Maps/Profiles
Create Route Map
Define Route Map for Import and Export
Name: Exp-Agg
Type: Match Prefix AND Routing Policy | Match Routing Policy Only
Description: optional
Order
Name
Action
Description
SUBMIT
CANCEL


Create Route Control Context
Create Route Context that will be included in this Profile
Order: 1
Name: Aggre-16prefix
Action: Deny | Permit
Description: optional
Match Rule: select a value
Set Rule: select a value
OK
CANCEL


Create Match Rule
Specify Match Rule for a Route Map
Name: Agg-Subnet
Description: optional
Match Regex Community Terms:
Name
Regular Expression
Community Type
Description
Match Community Terms:
Name
Description
Match Prefix:
IP
Aggregate
Description
10.32.0.0/16
UPDATE
CANCEL

Step 2

- In l3 out EPG subnet,
- Create a subnet for the supernet to aggregate,
- Mark it as ‘export route control subnet’
- Add route control profile for export and select the created route map in previous step



Result on the leaf

```
bdsol-aci32-leaf2# show ip ospf vrf RD-BGP:RD | egrep filter
        Area-filter in 'exp-ctx-proto-2654211'

bdsol-aci32-leaf2# show route-map exp-ctx-proto-2654211
route-map exp-ctx-proto-2654211, permit, sequence 4601
route-map exp-ctx-proto-2654211, permit, sequence 4602
  Match clauses:
    ip address prefix-lists: IPv4-proto16388-2654211-exc-ext-out-Export-agg1exp-Agg-32132Subnet-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
bdsol-aci32-leaf2# show ip prefix-list  IPv4-proto16388-2654211-exc-ext-out-Export-agg1exp-Agg-
32132Subnet-dst
ip prefix-list IPv4-proto16388-2654211-exc-ext-out-Export-agg1exp-Agg-32132Subnet-dst: 1 entries
   seq 1 permit 10.32.0.0/16 le 32
```

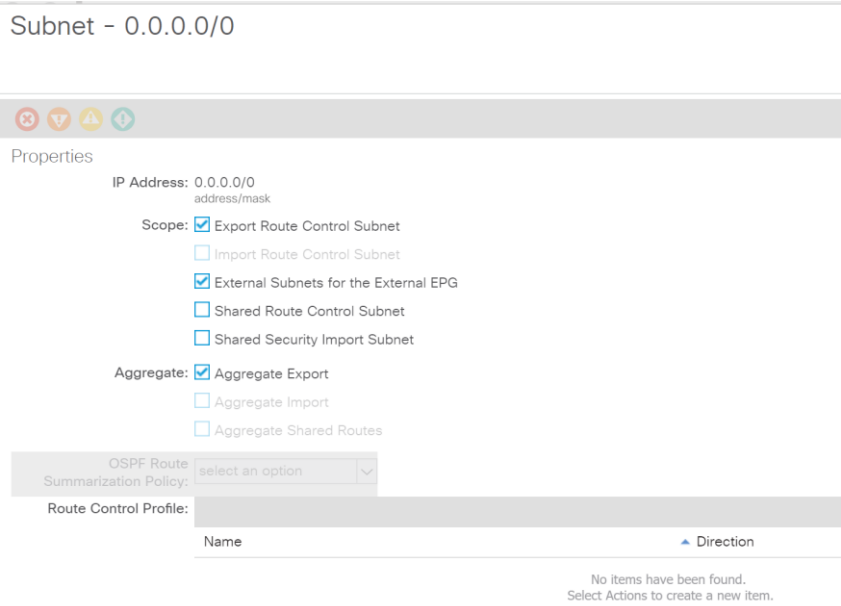
OSPD DB now have subnet of supernet as LSA 5

```
bdsol-aci32-leaf2# show ip ospf database external vrf RD-BGP:RD
        OSPF Router with ID (172.16.1.2) (Process ID default VRF RD-BGP:RD)

                Type-5 AS External Link States

Link ID         ADV Router      Age         Seq#       Checksum Tag
10.32.1.0       172.16.1.2      142         0x80000002 0x36af    4294967295
10.32.2.0       172.16.1.2      142         0x80000002 0x2bb9    4294967295
10.32.7.0       172.16.1.2      142         0x80000002 0xf3eb    4294967295
10.32.9.0       172.16.1.2      142         0x80000002 0xddff    4294967295
```

Aggregate supernet was 10.32.0.0/16
We now have all its underlying /24 subnet in OSPF and they are sent to External router

Transit Aggregate route for default route 0.0.0.0/0

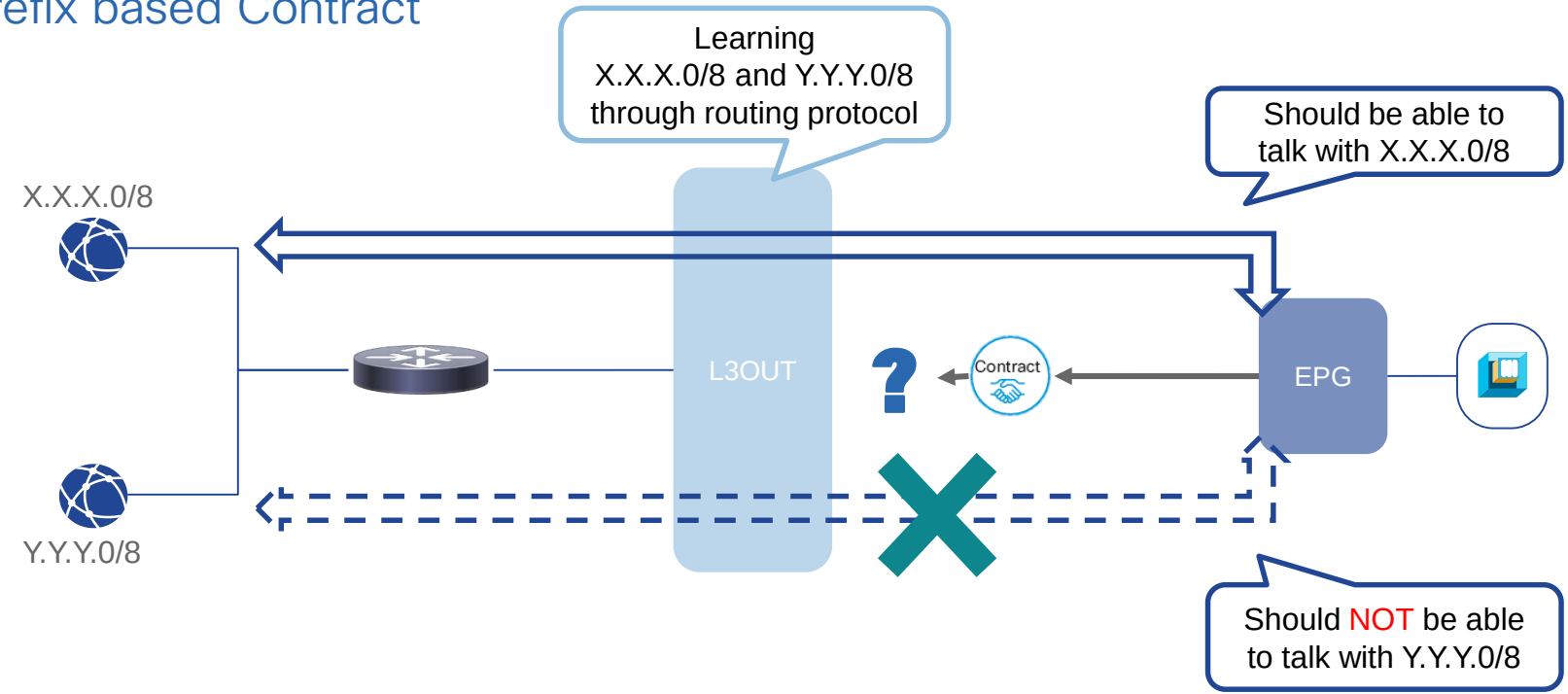
- The only exception is the route 0.0.0.0/0 which can be set as export route control subnet and export aggregate directly
- This will create prefix list with 0.0.0.0/0 le 32
- Note this can only be done for the 0.0.0.0/0
- It is not recommended as it can easily lead to temporarely routing loop in case of reconvergence if L3 out is shared on multiple router
- (see routing loop example later in this pres)



L3 out and contract prefix based contract enforcement

L3OUT Key Components

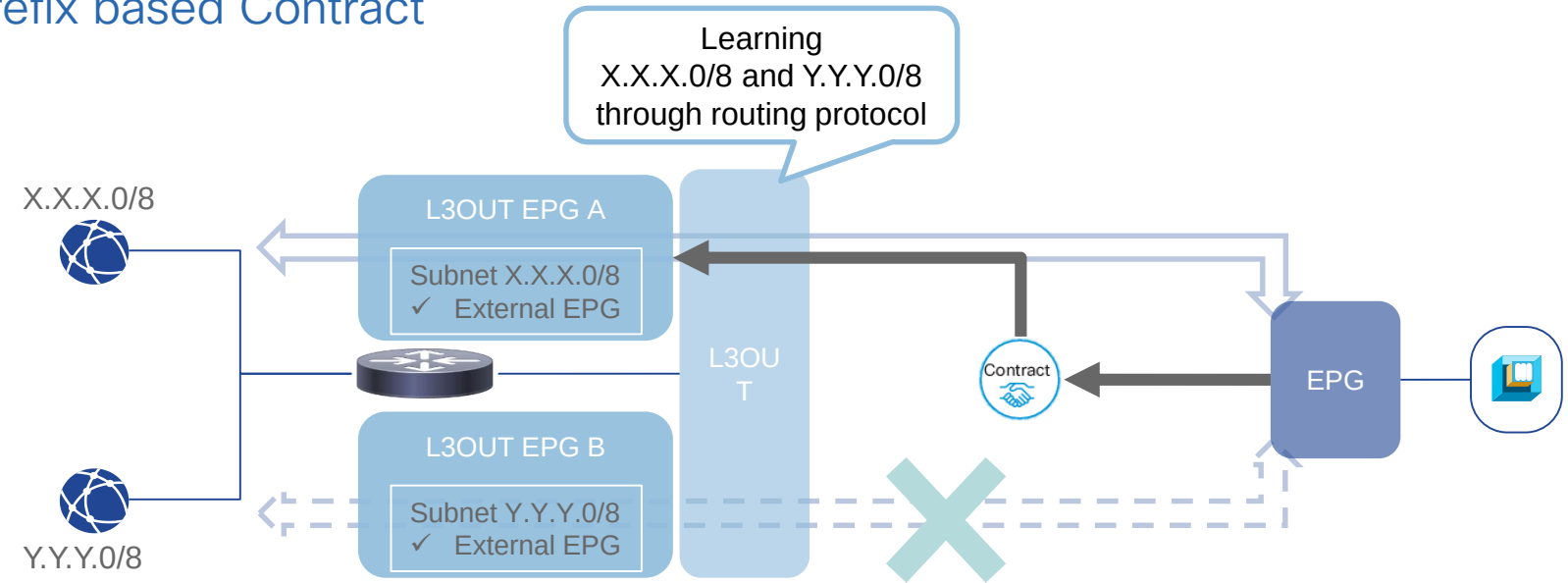
Prefix based Contract



How do we accomplish this ??

L3OUT Key Components

Prefix based Contract

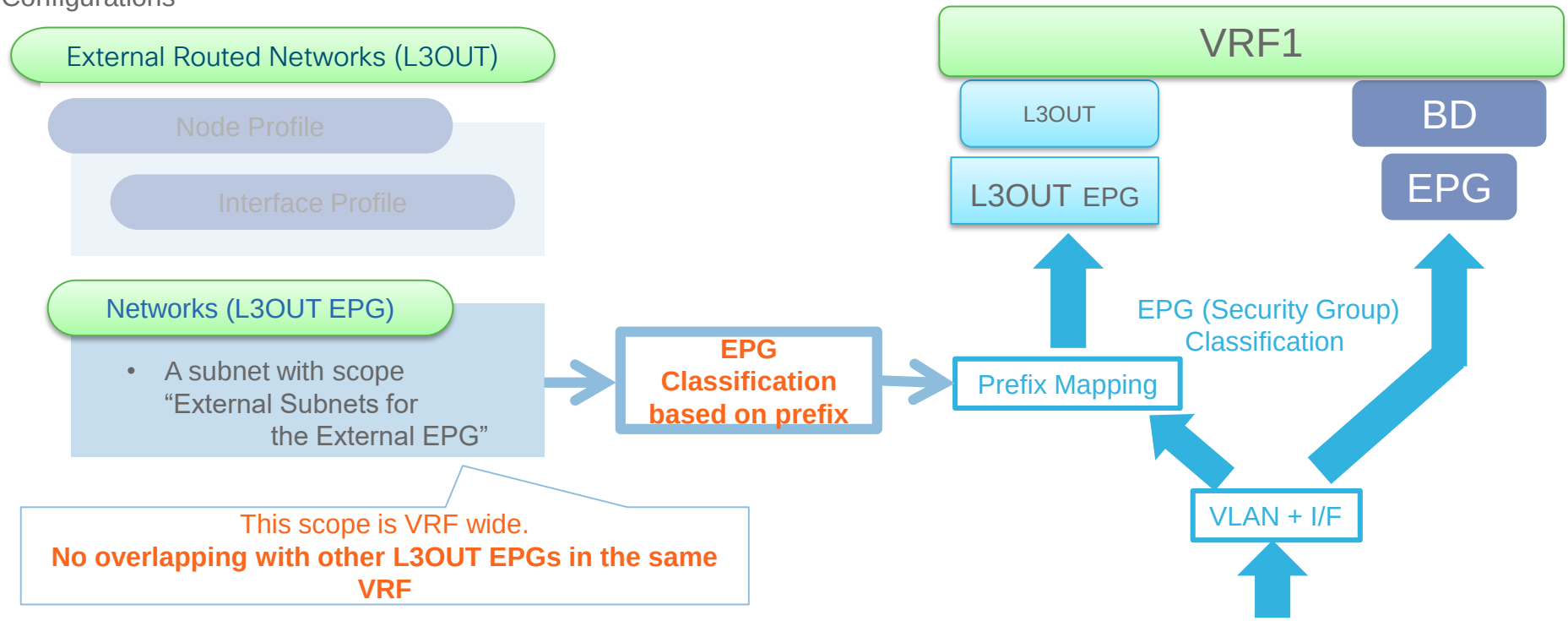


Prefix Based EPG (= L3OUT EPG)

L3OUT Key Components

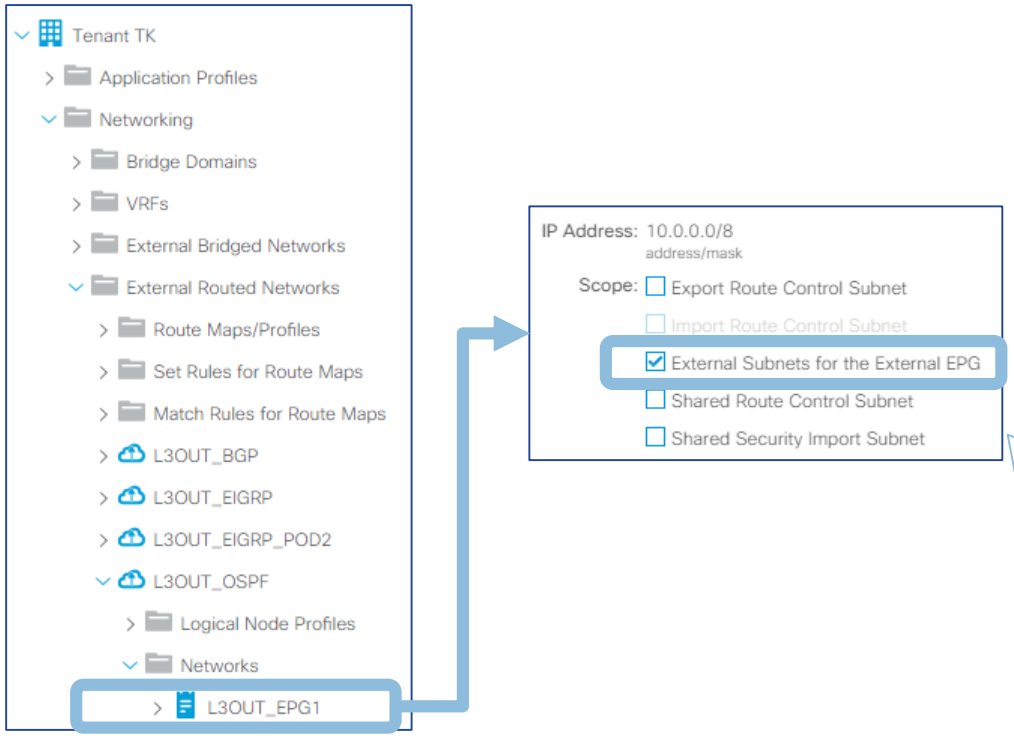
Prefix based Contract

Configurations



L3OUT Key Components

Prefix based Contract



"External Subnets for the External EPG" is to declare this subnet belongs to this L3OUT EPG

- To create prefix to pcTag mapping

NOTE:
It has nothing to do with routing table or routing protocol behavior unlike other Route Control Subnet scopes

A common mistake is selecting both "External Subnets for the External EPG" and "Export Route Control Subnet" for the same subnet, which implies a conflicting situation where the subnet behind the L3OUT but the same L3OUT is also expected to advertise/redistribute the subnet back to where it came from. It may not cause an immediate issue but unnecessary redistribution should always be avoided. Check L3OUT Subnet scope section for details.

CLI Verifications

Contract Drop on this leaf shows up here
Check both ingress/egress leaf just in case
or see appendix for Policy Control Enforcement Direction

1. Check if there is any contract drops

```
leaf# show logging ip access-list internal packet-log deny
[ Wed May  8 18:34:31 2019 155907 usecs]: CName: TK:VRF1(VXLAN: 2719744), VlanType: FD_VLAN, Vlan-Id: 26, SMac: 0x0050569185d1,
DMac:0x0022bdf819ff, SIP: 172.16.200.10, DIP: 35.224.99.156, SPort: 58968, DPort: 80, Src Intf: port-channel1, Proto: 6, PktLen: 74

[ Wed May  8 18:34:22 2019 963462 usecs]: CName: TK:VRF1(VXLAN: 2719744), VlanType: FD_VLAN, Vlan-Id: 26, SMac: 0x0050569185d1,
DMac:0x0022bdf819ff, SIP: 172.16.200.10, DIP: 35.224.99.156, SPort: 58968, DPort: 80, Src Intf: port-channel1, Proto: 6, PktLen: 74
```

2. Check VRF VNID

```
leaf# show vrf TK:VRF1 detail extended | grep vxlan
    Encap: vxlan-2097152
```

pcTag/contract is per VRF
except for shared service (VRF route leaking)

3. Check source (or destination) EPG pcTag

```
leaf# show system internal epm endpoint ip 192.168.1.1 | egrep 'VRF|sclass'
Vlan id : 30 ::: Vlan vnid : 9025 ::: VRF name : TK:VRF1
BD vnid : 16318374 ::: VRF vnid : 2097152
Flags : 0x80005c04 ::: sclass : 49162 ::: Ref count : 5
EP Flags : local|IP|MAC|host-tracked|sclass|timer|
```

If you source/destination is an endpoint, it should be in here.
sclass = pcTag = EPG ID for contract
Make sure the external IP is not here as this pcTag takes precedence over prefix-pcTag mapping table. If it is, check the traffic path that caused ACI to learn the external IP as an endpoint.

4. Check destination (or source) L3OUT prefix based EPG pcTag

```
leaf# vsh_lc -c 'show system internal aclqos prefix' | egrep 'Vrf|10.0.0.0'
Vrf-Vni VRF-Id Table-Id          Addr                    Class  Shared Remote Complete
2097152 8      0x8               10.0.0.0/8              49158  0      1      No

 === use this command from 3.2 ===
leaf# vsh -c 'show sytem internal policy-mgr prefix'
```

"External Subnet for the External EPG" config is reflected here.
This is Longest Prefix Match.

CLI Verifications

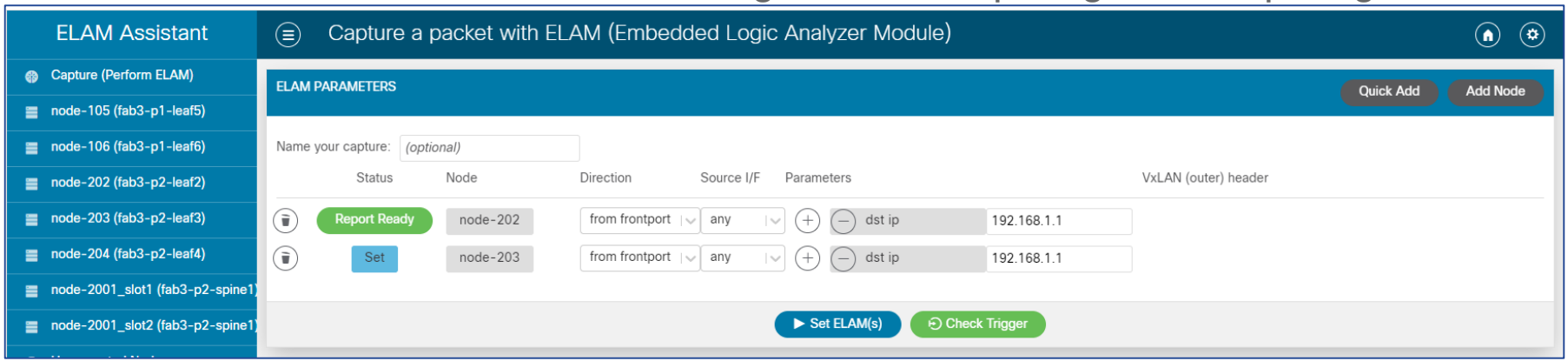
5. Check contracts between two pcTags

scope = VRF VNID

```
leaf# show zoning-rule scope 2097152 | egrep 'Rule|49162|49158'
Rule ID      SrcEPG      DstEPG      FilterID      operSt      Scope      Action      Priority
4165         49162       49158       5             enabled     2097152    permit      fully_qual(7)
4124         49158       49162       5             enabled     2097152    permit      fully_qual(7)
```

```
leaf# show zoning-filter filter 5
FilterId  Name   EtherT  ArpOpc       Prot     MatchOnlyFrag Stateful SFromPort    SToPort      DFromPort    DToPort
~snip~
========  ====== ======  ==========   =======  ====== ======= ========     ====         ====         ====
~snip~
5         5_0    ip      unspecified  icmp     no             no       unspecified unspecified unspecified unspecified
~snip~
```

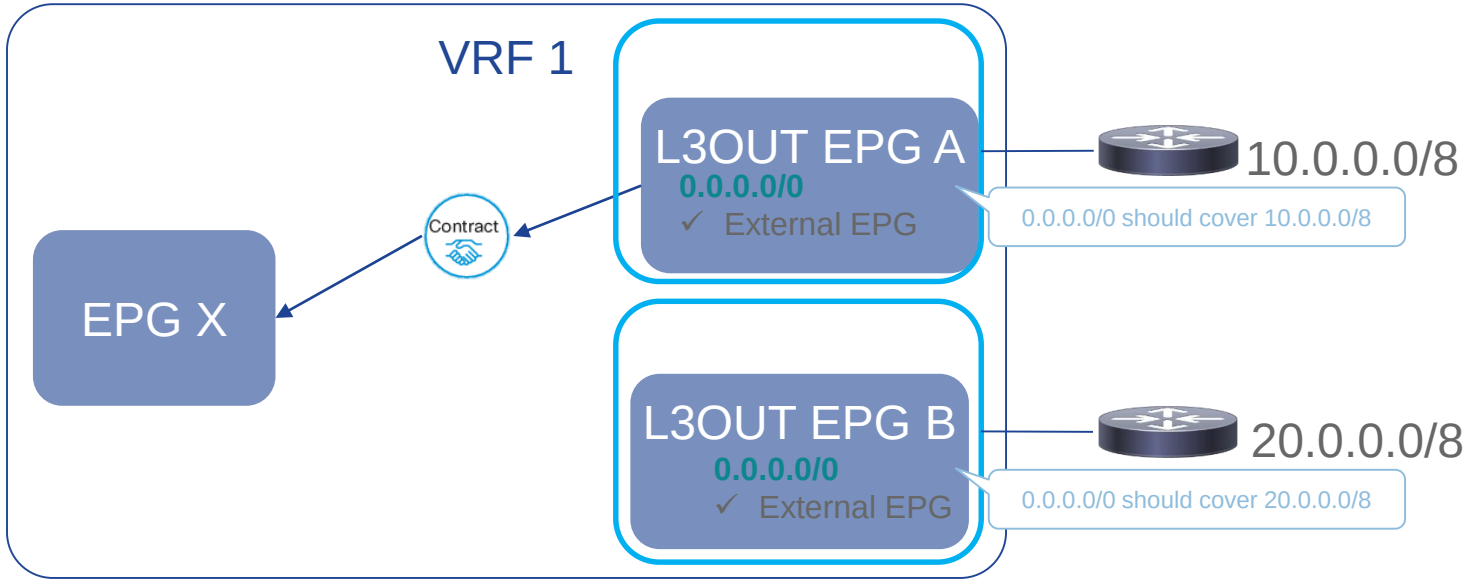
6. Check ELAM to see if the traffic is using correct src pcTag and dst pcTag



https://aciappcenter.cisco.com/elam-assistant-2-2-1n-0-2-262.html

L3OUT Contract

Common Issue (L3OUT EPGs with 0.0.0.0/0)



The problem is both 10.0.0.0/8 and 20.0.0.0/8 can talk to EPG X
even though there is no contract between L3OUT EPG B and EPG X

➢ Let us learn pcTag and L3OUT prefix mapping

L3OUT Contract

Common Issue (L3OUT EPGs with 0.0.0.0/0)

1. Check VRF VNID

```
leaf# show vrf TK:VRF1 detail extended | grep vxlan
    Encap: vxlan-2097152
```

2. Check source (or destination) EPG pcTag

```
leaf# show system internal epm endpoint ip 192.168.1.1 | egrep 'VRF|sclass'
Vlan id : 30 ::: Vlan vnid : 9025 ::: VRF name : TK:VRF1
BD vnid : 16318374 ::: VRF vnid : 2097152
Flags : 0x80005c04 ::: sclass : 49162 ::: Ref count : 5
EP Flags : local|IP|MAC|sclass|timer|
```

3. Check destination L3OUT 0.0.0.0/0 EPG pcTag

```
leaf# vsh_lc -c 'show system internal aclqos prefix' | egrep 'Vrf|0.0.0.0'
Vrf-Vni VRF-Id Table-Id          Addr                    Class  Shared Remote Complete
2097152 8      0x8               0.0.0.0/0                15     0      0      No
```

There is only one entry for each subnet per VRF. Any L3OUT traffic without more granular prefix entry would use pcTag 15 for 0.0.0.0/0.

NOTE: this is not a routing table. It doesn't matter if routing table has more granular route

This contract is due to EPG X <-> L3OUT A But any other L3OUT that hits 0.0.0.0/0 in prefix table will use this rule

4. Check contracts between pcTags

```
leaf# show zoning-rule scope 2097152 | egrep 'Rule|49162'
Rule ID         SrcEPG          DstEPG          FilterID        operSt          Scope           Action          Priority
4165            49162           15              5               enabled         2097152         permit          fully_qual(7)
```

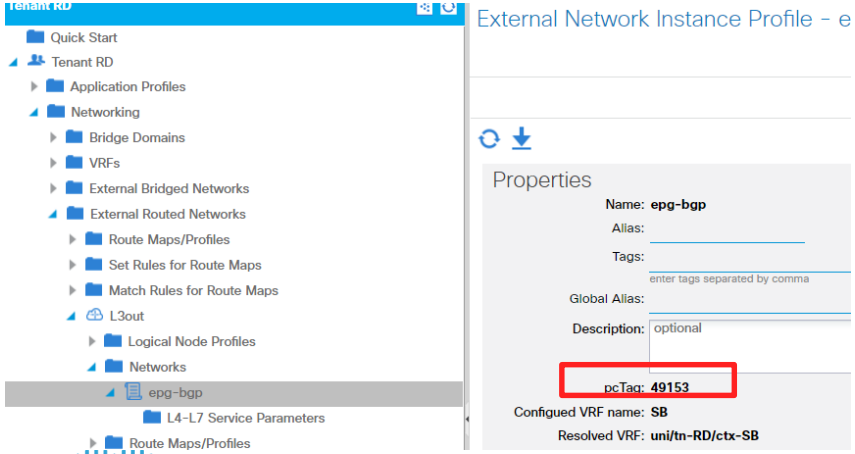
Basic Rule

Only one 0.0.0.0/0 for "External Subnet for the External EPG" per VRF

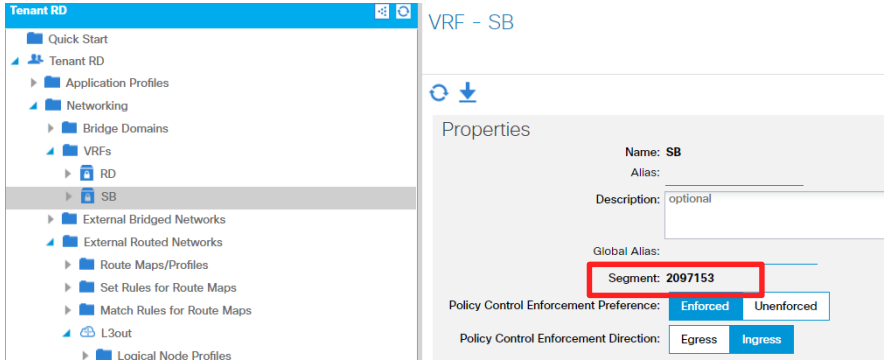
Or you risk to get flow going through contract which is not expecte

3. External subnet for external EPG finding pcTag and vrf scope

- L3 out epg gives pcTag
- Or pcTag parameter of class l3extInstP



- VRF gives vrf scope
- Or scope parameter of class fvCtx



3. External subnet for external EPG Seeing all external subnet list

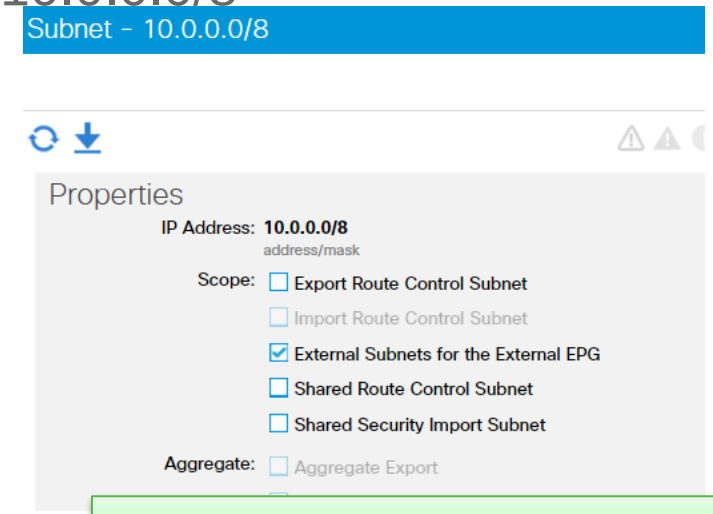
- Here we lookup all aclqos prefix for vrf with scope 3014656
- We have at least 2 l3 out epg 16390 and 32772
- 10.200.2.0/24 would be assigned to 1390
- 10.200.1.0/24 would be assigned to epg 32772

```
module-1# show system internal  aclqos  prefix  | egrep "==|Scope|3014656"
Vrf-Vni VRF-Id Table-Id          Addr                       Scope Class  Shared Remote Complete
======= ====== ============ ============================== ===== ====== ====== ====== ========

3014656 8      0x7           10.200.2.0/24                    7     16390  0      1      No
3014656 8      0x7           172.16.0.0/16                    7     16390  0      1      No
3014656 8      0x7           10.0.0.0/8                       7     16390  0      1      No
3014656 8      0x7           10.200.1.0/24                    7     32772  0      1      No
```

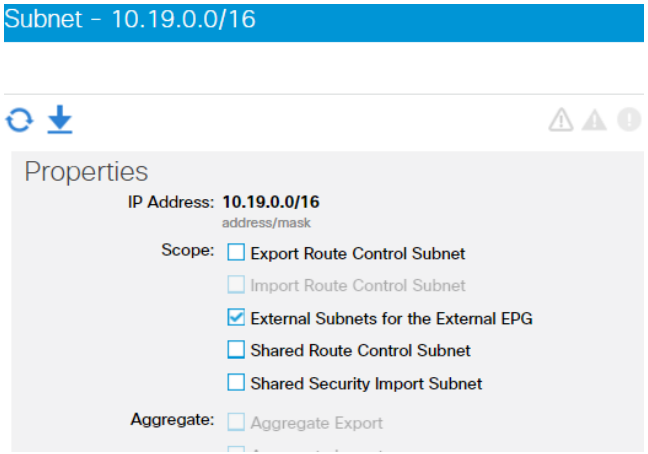
Deprecated in 3.2
For Gen-2, Gen-3 leaf

3. External subnet for external EPG Overlapping subnet

On one of the l3 out epg we set 10.0.0.0/8



On a second l3 out epg we set 10.19.0.0/16



LPM lookup if 10.0.0.0/8 ip we assign to 16390 EPG unless it is a 10.19.0.0/16 which goes to 32772

```
module-1# show system internal  aclqos prefix | egrep "3014656.*10."
3014656 7      0x6            10.0.0.0/8                  6     16390  0      1      No
3014656 7      0x6            10.19.0.0/16                6     32772  0      0      No
```

3.2 and above note

- Starting in 3.2 we do not use anymore aclqos prefix for Gen-2, Gen-3 leaf (EX, FX).
- See CSCvk16258 Aclqos Prefix Output Empty for EX and FX Switches in 3.2
- A replacement Cli is provided in vsh (not vsh_lc) see below

```
bdsol-aci32-leaf2# vsh -c "show system internal policy-mgr prefix"
Requested prefix data

Vrf-Vni VRF-Id Table-Id Table-State  VRF-Name                       Addr                                  Class Shared Remote Complete
======= ======  =========== =======  ============================== ====================================  ====== ====== ====== ========
2654211 7      0x7          Up      RD-BGP:RD                                    170.0.0.0/8   16387   False  True   False
2654211 7      0x7          Up      RD-BGP:RD                                 172.16.99.0/24   16387   False  True   False
2654211 7      0x7          Up      RD-BGP:RD                                 172.16.1.10/32   16387   False  True   False
```

L3OUT Contract

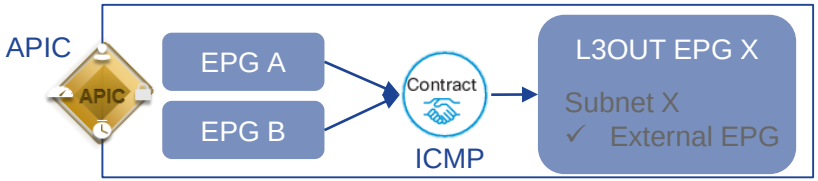
Policy Control Enforcement Direction

Under VRF

Policy Control Enforcement Direction: **Egress** | Ingress

Allows to usually enforce Policy on Server leaf

A feature to save contract TCAM usage on border LEAF



No effects on EPG <-> EPG traffic

Egress Policy Enforcement

Policy Control Enforcement Direction: **Egress** | Ingress

Non-Border LEAF(s)

with EPG A

source	destination	Filter
pcTag A	pcTag X	ICMP

with EPG B

source	destination	Filter
pcTag B	pcTag X	ICMP

Border LEAF(s)

source	destination	Filter
pcTag A	pcTag X	ICMP
pcTag B	pcTag X	ICMP

Ingress Policy Enforcement

default from 1.2

Policy Control Enforcement Direction: Egress | **Ingress**

Non-Border LEAF(s)

with EPG A

source	destination	Filter
pcTag A	pcTag X	ICMP

with EPG B

source	destination	Filter
pcTag B	pcTag X	ICMP

Border LEAF(s)

source	destination	Filter
- none -		

L3OUT Contract

Policy Control Enforcement Direction

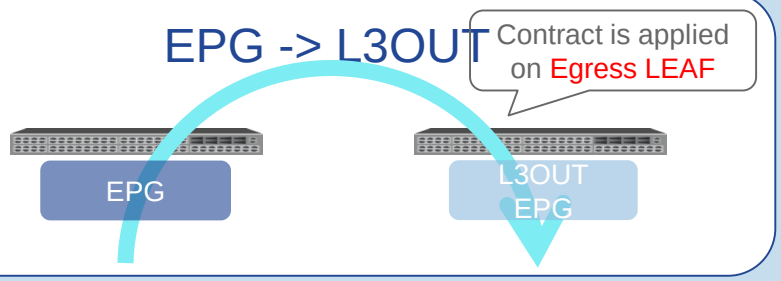
Under VRF

Policy Control Enforcement Direction: **Egress** Ingress

How does it affect traffic flow and contract?

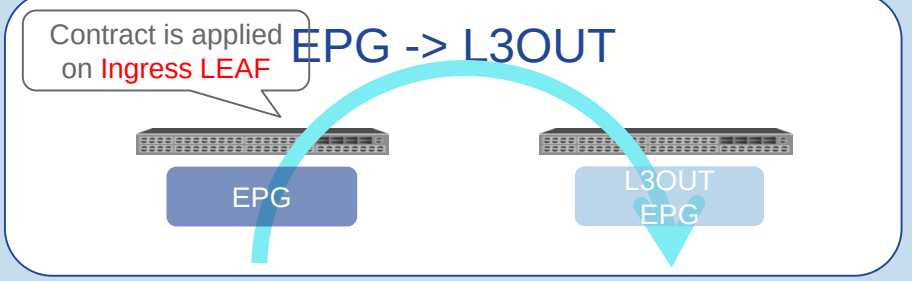
Egress Policy Enforcement

Policy Control Enforcement Direction: **Egress** Ingress



Ingress Policy Enforcement

Policy Control Enforcement Direction: Egress **Ingress**



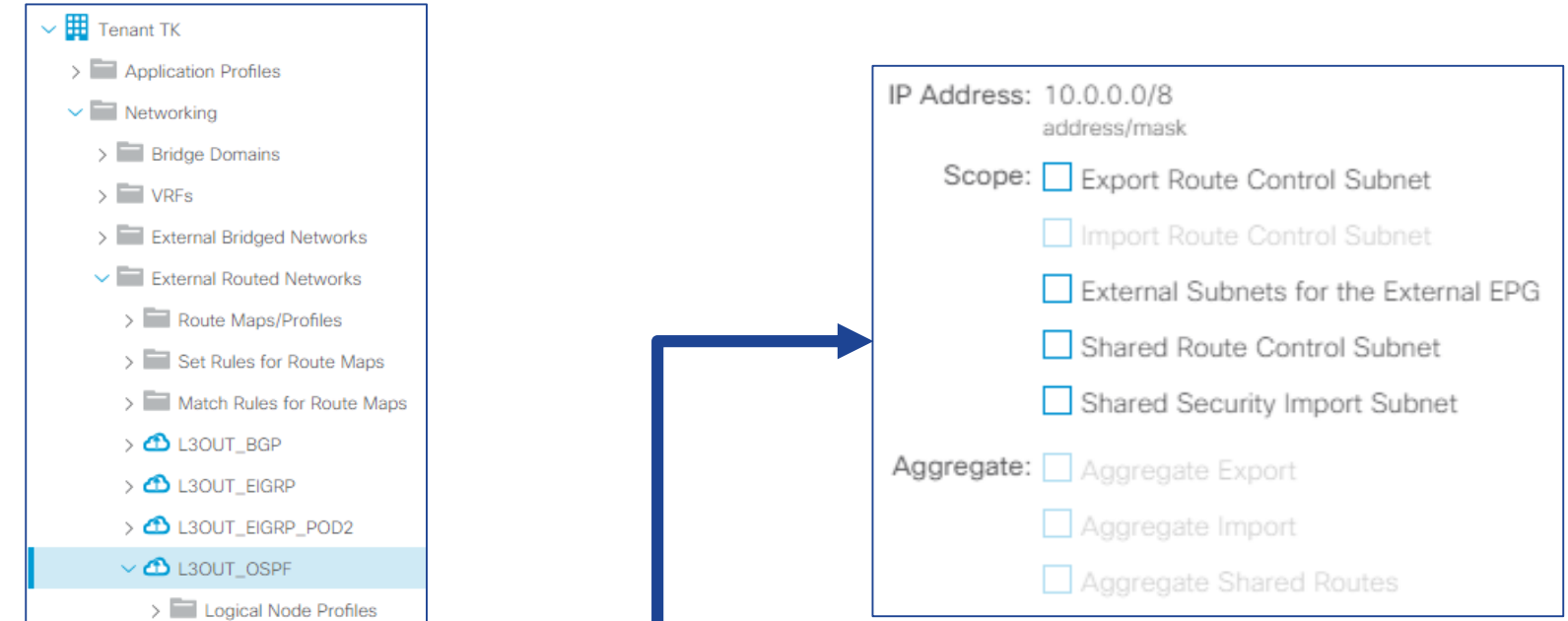
Consequence

- With ingress policy enforcement that means that typically policy-mgr prefix table will be the same on all server leaf
- If a leaf is part of the vrf
  - (vsh) Show system internal policy-mgr prefix (3.2 and after)
  - (vsh_lc) show system internal aclqos prefix (before 3.2)

  Those will be the same and will contains ext subnet for external EPG LPM table merged of all L3 out EPG of the VRF no matter if they are on some BL or not

L3 out – Subnet Flags detail

L3OUT Subnet Scope – When to use what Summary



L3OUT Subnet Scope

Route Control for Routing Protocol

- Export Route Control Subnet
- Import Route Control Subnet
- Shared Route Control Subnet (inter VRF route control – not covered here)

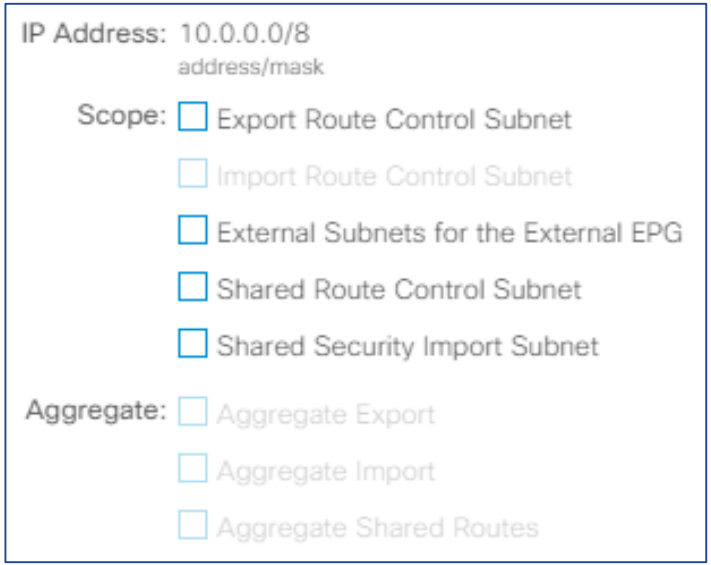
Traffic Classification for Contract

- External Subnets for the External EPG
- Shared Security Import Subnet (inter vrf Classificaton – not covered here)

Aggregate

- Aggregate Export
- Aggregate Import
- Aggregate Shared Routes

Grouping by functionality



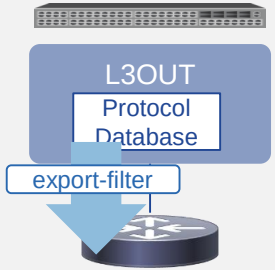
L3OUT Subnet Scope Summary

Only for contracts
No impact in routing table

Route Control for Routing Protocol

Export Route Control Subnet

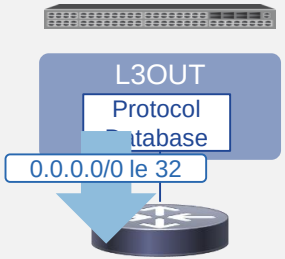
(Mainly for Transit Routing)



Advertise the route from ACI to outside

aggregation

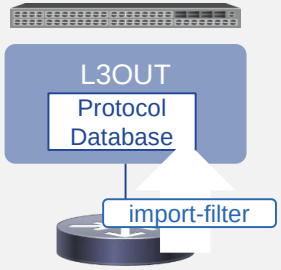
Aggregate Export



Advertise all routes from ACI to outside

Import Route Control Subnet

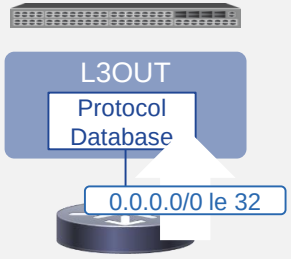
(Only for OSPF or BGP)



Receive the route from outside (by default, receive all)

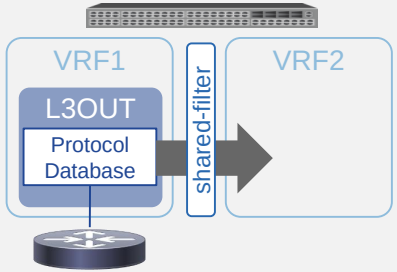
aggregation

Aggregate Import



Receive all routes from outside

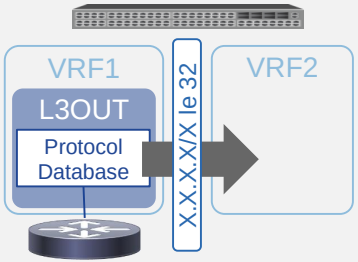
Shared Route Control Subnet



Leak the external route to different VRF

aggregation

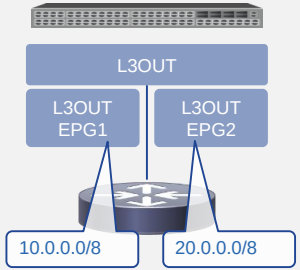
Aggregate Shared Route



Leak multiple external routes to different VRF

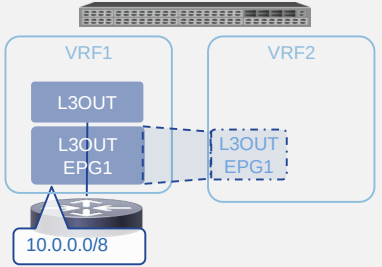
Subnet Classification

External Subnet for the External EPG



Group subnets into each L3OUT EPG (pcTag)

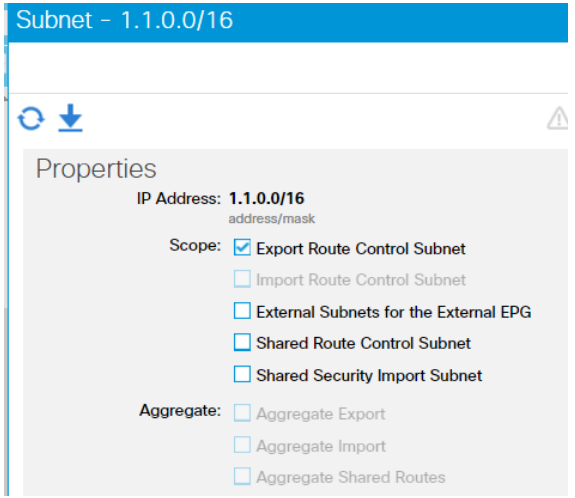
Shared Security Import



Leak prefix-pcTag mapping to different VRF

1. External Route Control Subnet

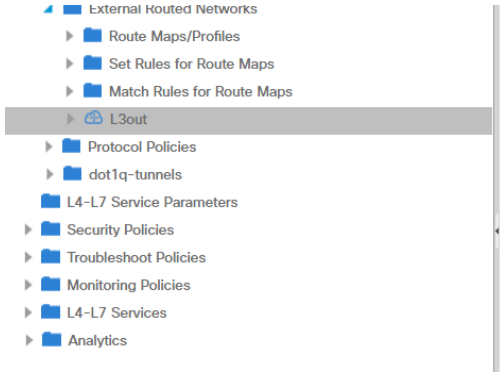
- use in transit routing, to advertise prefix received from a first L3 out (or GOLF) down to a second layer 3 out
- Directly adds entry in prefix-list of the outbound route-map
- Should NOT contain the the prefix you are suppose to receive on that Layer 3 out
- Does not matter whether the prefix exist or is received.
- Do not add routes, only plays with route-map
- Route-map is on BGP neighbor or ospf area-filter
- May be used for BD subnet advertisement

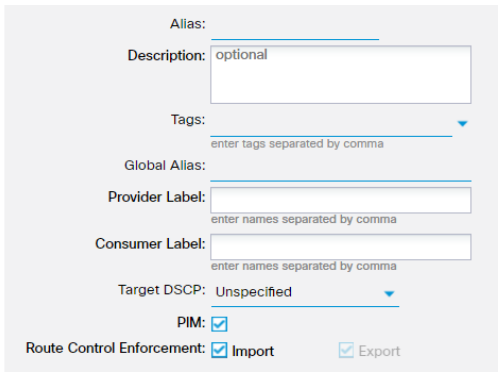


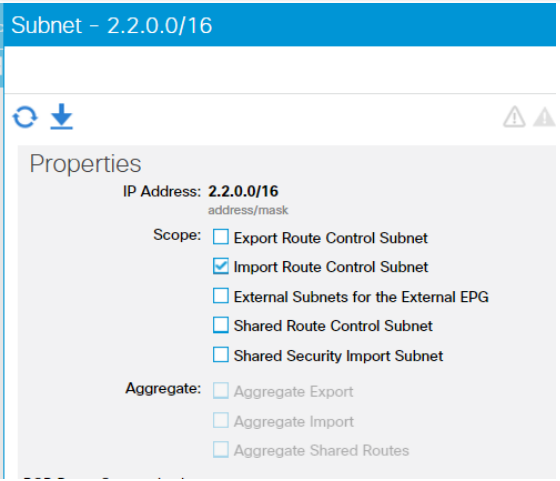

```
bdsol-aci32-leaf2# show ip ospf  vrf DC:DC  | egrep lter
        Area-filter in 'exp-ctx-proto-3014656'
bdsol-aci32-leaf2# show route-map exp-ctx-proto-3014656
route-map exp-ctx-proto-3014656, permit, sequence 7801
  Match clauses:
    ip address prefix-lists: IPv4-proto32772-3014656-exc-ext-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
bdsol-aci32-leaf2# show ip prefix-list IPv4-proto32772-3014656-exc-ext-inferred-export-dst
ip prefix-list IPv4-proto32772-3014656-exc-ext-inferred-export-dst: 1 entries
   seq 1 permit 1.1.0.0/16
```

2. Import Route Control Subnet

- Disabled by default (greyed out)
- Only enable if the L3 out is set with import route control enable
- Works the same as export route control subnet but on the import route-map
- Only supported for BGP or OSPG (not EIGRP)



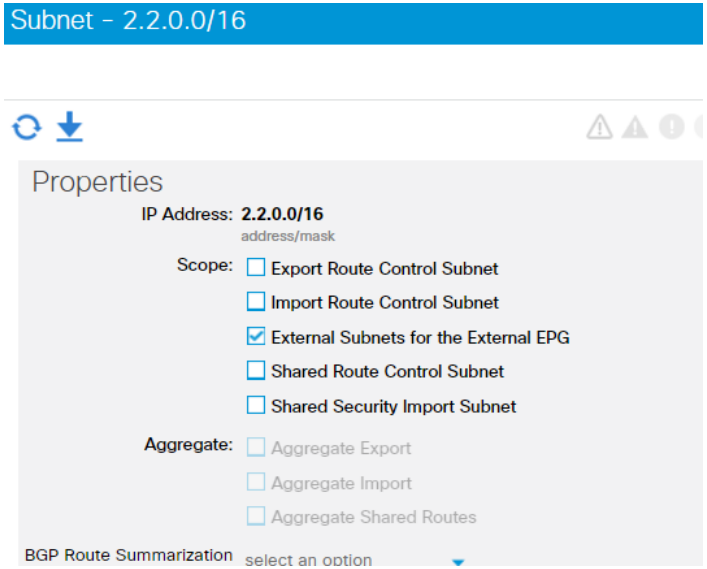





```
bdsol-aci32-leaf2# show ip bgp neigh vrf RD:SB | egrep route-map
  Outbound route-map configured is exp-l3out-L3out-peer-2097153, handle obtained
  Inbound route-map configured is imp-l3out-L3out-peer-2097153, handle obtained
bdsol-aci32-leaf2# show route-map  imp-l3out-L3out-peer-2097153
route-map imp-l3out-L3out-peer-2097153, permit, sequence 7801
  Match clauses:
    ip address prefix-lists: IPv4-peer49153-2097153-exc-ext-inferred-import-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
bdsol-aci32-leaf2# show ip prefix-list IPv4-peer49153-2097153-exc-ext-inferred-import-dst
ip prefix-list IPv4-peer49153-2097153-exc-ext-inferred-import-dst: 1 entries
   seq 1 permit 2.2.0.0/16
```

3. External subnet for external EPG

- By far the most important
- Is used for datapath traffic to assign external prefix to the correct l3 epg.
  - Used both for ingress traffic and egress traffic
- Should typically contains the external prefix we expect to receive on that L3 out
- SHOULD NOT contains, BD subnet or subnet received on a different L3 out
- Might contain default internet route 0.0.0.0/0
- Also act on the routing table :
  - On Gen1 – install a route in LPM (longest prefix match) to redirect routes to broadcom)
  - On Gen-2 – also install a route in LPM of the ASIC, but punt all smaller subnet to the EGP
- Keep in mind EPG assignment is based on LPM lookup



Remark

- Note seems that in 2.3 or 3.x software even if you didn't specify a subnet with Ext Subnet with external EPG we do install always 0.0.0.0/0 in aclqos prefix LPM table by default to sclass 15 even if not configured.

- However it seems we do not push zoning-rule to it . So still need to configure it to get the zoning-rule pushed

3.2 and above note

- Starting in 3.2 we do not use anymore aclqos prefix for Gen-2, Gen-3 leaf (EX, FX).
- See CSCvk16258 Aclqos Prefix Output Empty for EX and FX Switches in 3.2
- A replacement Cli is provided in vsh (not vsh_lc) see below

```
bdsol-aci32-leaf2# vsh -c "show system internal policy-mgr prefix"
Requested prefix data

Vrf-Vni VRF-Id Table-Id Table-State  VRF-Name                       Addr                                  Class Shared Remote Complete
======= ======  =========== =======  ============================== ====================================  ====== ====== ====== ========
2654211 7      0x7          Up      RD-BGP:RD                                    170.0.0.0/8   16387   False  True   False
2654211 7      0x7          Up      RD-BGP:RD                                 172.16.99.0/24   16387   False  True   False
2654211 7      0x7          Up      RD-BGP:RD                                 172.16.1.10/32   16387   False  True   False
```

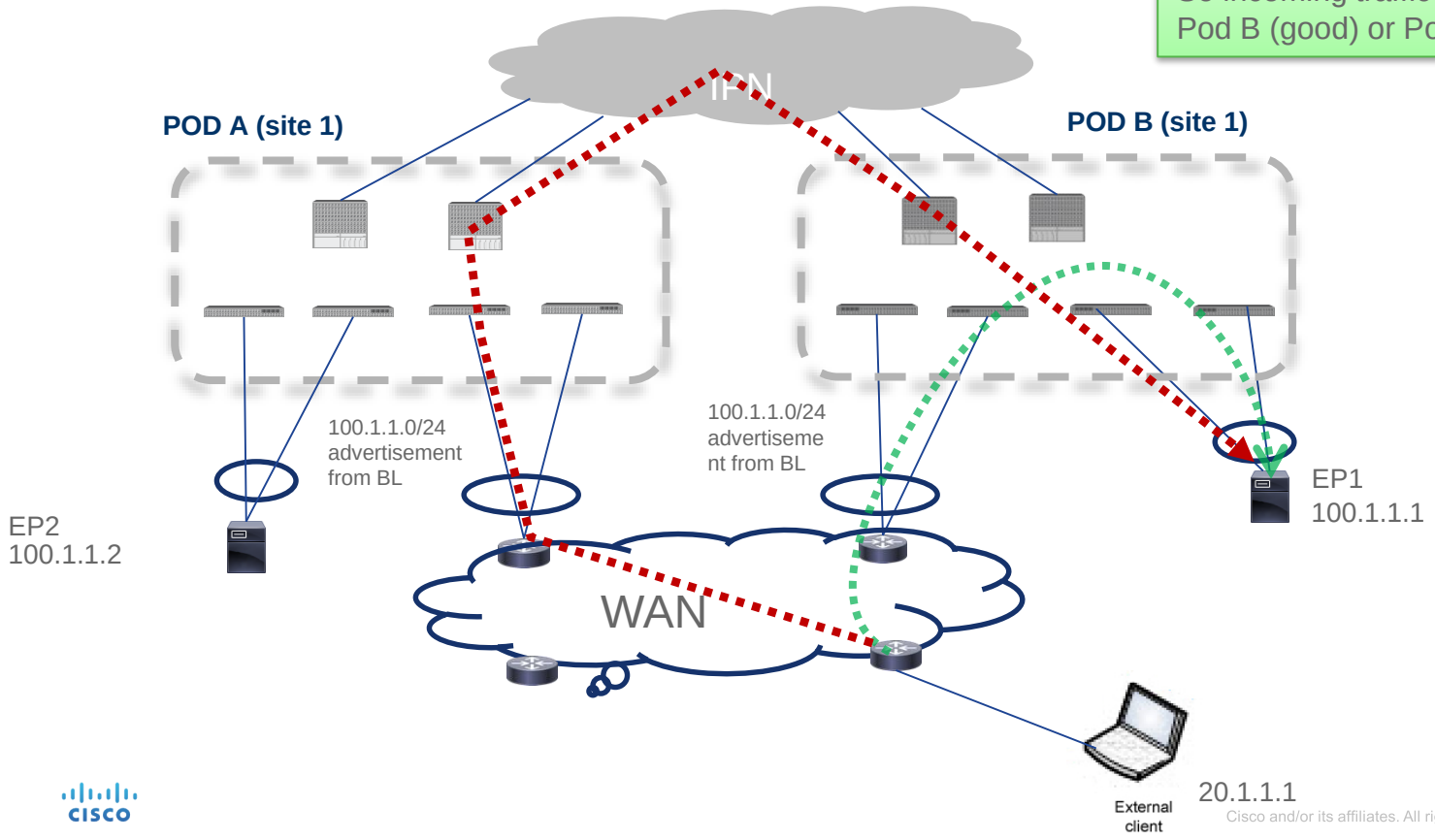
Host Based Routing – ACI 4.0

Why do we need Host based routing ?

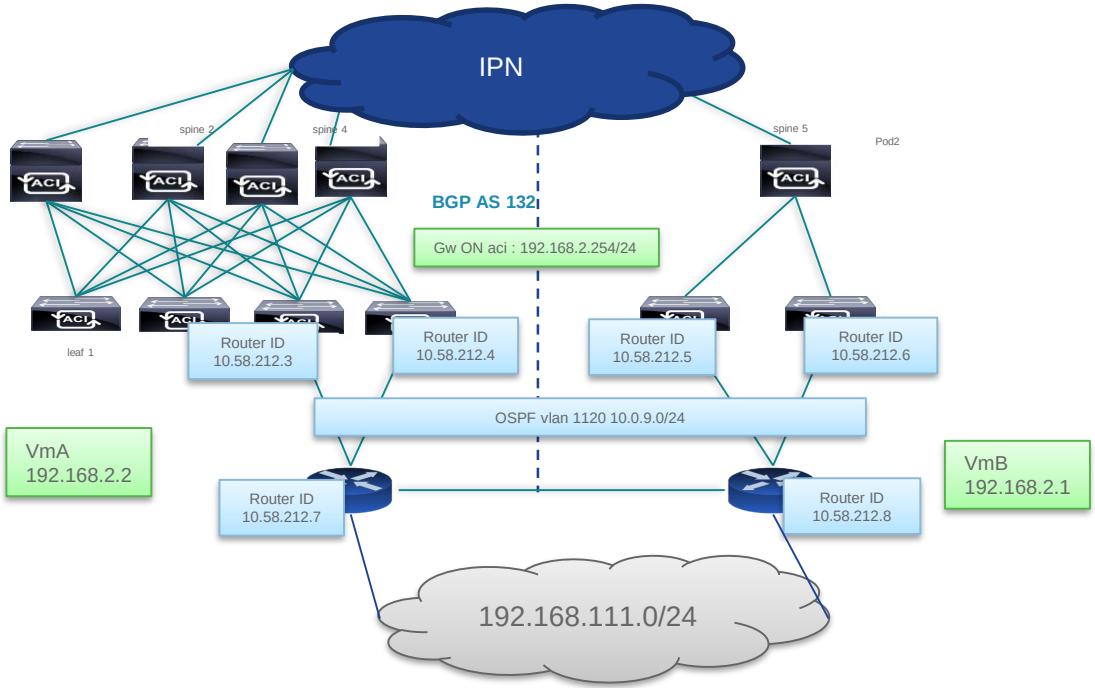
- In case of multipod, we want external traffic that needs to reach an EP in ACI to enter ACI in the Pod where the EP is to avoid tromboning on the IPN
- Before 4.0 , only option to achieve that was using GOLF and advsertise evpn type 2 update for each EP in BD only on local pod spine so that we advertise /32 and golf router would send traffic for EP to the right pod (using /32)
- Host based routing achieve the same goal in traditional distributed l3 out across Pod

Traffic Asymmetry

Same BD subnet 100.1.1.0/24
Advertised from Pod A BL and Pod B BL (stretched L3 out).
So incoming traffic to 100.1.1.1 may enter Pod B (good) or Pod A (require crossing IPN



Lab setup RD-OSPF:RD – stretched L3 out

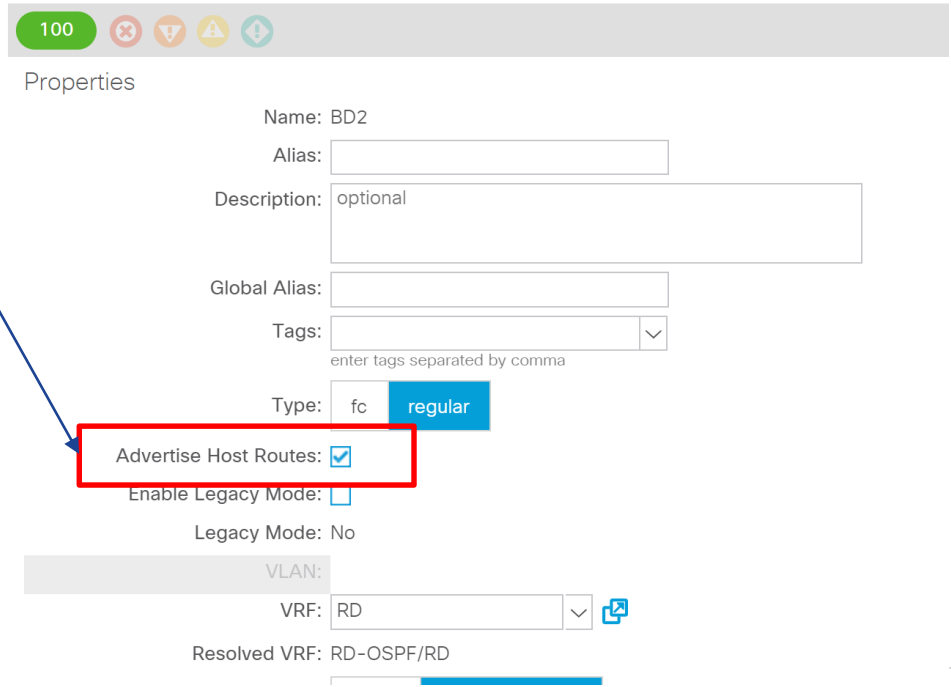


Config

Step 1
Just need to enable
Advertize Host Routes flag
In BD setting

Step 2
BD subnet must be sent out
(like before) either
By RsBDToOut (BD to L3 map)
Or by route-map (exp route control)

Bridge Domain - BD2

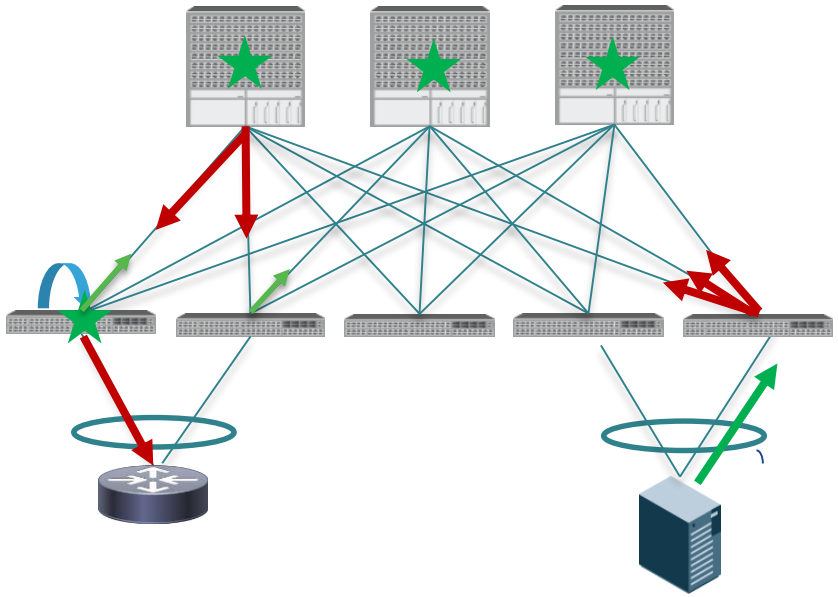


Routing Table on external router

Without host route , both VM are reachable using ECMP between the 4 BL
With Host route, each VM is reachable using the 2 BL local to the pod where it sits

```
192.168.2.0/24, ubest/mbest: 4/0
    *via 10.0.9.3, Vlan1120, [110/20], 00:17:30, ospf-666, type-2
    *via 10.0.9.4, Vlan1120, [110/20], 00:17:29, ospf-666, type-2
    *via 10.0.9.5, Vlan1120, [110/20], 00:17:30, ospf-666, type-2
    *via 10.0.9.6, Vlan1120, [110/20], 00:17:29, ospf-666, type-2
192.168.2.1/32, ubest/mbest: 2/0
    *via 10.0.9.5, Vlan1120, [110/1], 00:00:37, ospf-666, type-2, tag 4294967295
    *via 10.0.9.6, Vlan1120, [110/1], 00:00:38, ospf-666, type-2, tag 4294967295
192.168.2.2/32, ubest/mbest: 2/0
    *via 10.0.9.3, Vlan1120, [110/1], 00:00:40, ospf-666, type-2, tag 4294967295
    *via 10.0.9.4, Vlan1120, [110/1], 00:00:36, ospf-666, type-2, tag 4294967295
```

HBR Design Flow (contd.)



What happens for each host-routes

- EPM Learns the host and updates COOP
- COOP Citizens updates the Host information to Spines (Oracles)
- Spines downloads the Host-Routes to the Border-Leafs.
- Coop updates all the routes in RIB with not-to-fib flag. RIB will stop installing all these routes in FIB.
- Routing Protocol (BGP, OSPF, EIGRP) gets all the Host-Routes.
- Host-routes will be advertised outside the fabric with Transit VRF flag set to prevent loops.

As soon as the Feature is configured on a BD :

1. COOP on border-leafs publishes Host-Route interest to Spine via Mrouter Record
2. On BL coop-rib-leak route-map is configured to leak coop host route to RIB
3. On BL redistribution route-map updates the prefix-list to mark the BD subnet as ‘le 32’

Result of HBR config in a BD Check

BL register BD host route to COOP

- COOP Citizen needs to notify Oracle about Host-Route interest on BD.
- To achieve this, HBR features rides on the existing IGMP MROUTER functionality.
- HOST_ROUTE flag is the key to identify if the Border-Leaf has published Host-Route interest for the BD-VNID to the Oracle.
- spine learns EPs under BD-VNID and it notifies all HOST_ROUTE enabled leafs under that BD-VNID regarding the EPs.

- Only local BL to the pod express interest (here leaf3-4) in pod-1

```
bdsol-aci32-spine1# show coop  internal  info  repo mrouter
key 16089027


Repo Hdr Checksum : 51580
Repo Hdr record timestamp : 12 20 2018 07:02:12 613424930
Repo Hdr last pub timestamp : 12 20 2018 07:02:12 613675835
Repo Hdr last dampen timestamp : 01 01 1970 00:00:00 0
Repo Hdr dampen penalty : 0
Repo Hdr flags : IN_OBJ
BD Vnid : 16089027
flags : 0x2 HOST_ROUTE
num of leafs in record : 2
num of valid leafs in record : 2
Leaf 0 Info :
Leaf Repo Hdr Checksum : 0
Leaf Repo Hdr record timestamp : 12 20 2018 07:02:12
613424930
Leaf Repo Hdr last pub timestamp : 12 20 2018 07:02:12
613675835
Leaf Repo Hdr last dampen timestamp : 01 01 1970 00:00:00 0
Leaf Repo Hdr dampen penalty : 0
Leaf Repo Hdr flags : IN_OBJ
Leaf tep ip : 10.0.88.90
Leaf Flags : 0x2 HOST_ROUTE
Leaf 1 Info :
Leaf Repo Hdr Checksum : 0
Leaf Repo Hdr record timestamp : 12 20 2018 06:23:07
278690455
Leaf Repo Hdr last pub timestamp : 12 20 2018 06:23:07
278956179
Leaf Repo Hdr last dampen timestamp : 01 01 1970 00:00:00 0
Leaf Repo Hdr dampen penalty : 0
Leaf Repo Hdr flags : IN_OBJ
Leaf tep ip : 10.0.88.91
Leaf Flags : 0x2 HOST_ROUTE
Hash: 2532389223 owner: 10.0.128.64
```

Route-map for COOP to RIB leak

- Route-map on Border leaf to allow leak from COOP to RIB for BD subnet with HBR enable

```
bdsol-aci32-leaf3# show coop internal host-route bridge-domain 16089027
Host-Based Routing BD Details:
bd-vnid:16089027, flags:0x1
host-route: Enabled
vrf[0]: RD-OSPF:RD, vnid:2621441 flags:0x1
policy af:IPv4 name:coop-ribleak-2621441 cfg:1 hdl:251567012
policy af:IPv6 name:coop-ribleak-2621441 cfg:1 hdl:251565556

bdsol-aci32-leaf3# show route-map coop-ribleak-2621441
route-map coop-ribleak-2621441, permit, sequence 1
  Match clauses:
    ip address prefix-lists: IPv4-coop-ribleak-2621441-16089027
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
route-map coop-ribleak-2621441, deny, sequence 20000
  Match clauses:
  Set clauses:
bdsol-aci32-leaf3#
bdsol-aci32-leaf3#
bdsol-aci32-leaf3# show ip prefix-list IPv4-coop-ribleak-2621441-16089027
ip prefix-list IPv4-coop-ribleak-2621441-16089027: 1 entries
   seq 1 permit 192.168.2.0/24 le 32
```

Route-map from COOP route in RIB to OSPF (or eigrp or bgp).

```
bdsol-aci32-leaf3# show ip ospf vrf RD-OSPF:RD | egrep coop
   coop route-map exp-ctx-st-2621441
bdsol-aci32-leaf3#
bdsol-aci32-leaf3# show route-map exp-ctx-st-2621441
route-map exp-ctx-st-2621441, deny, sequence 1
  Match clauses:
    tag: 4294967294
  Set clauses:
route-map exp-ctx-st-2621441, permit, sequence 15801
  Match clauses:
    ip address prefix-lists: IPv4-st16386-2621441-exc-ext-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
route-map exp-ctx-st-2621441, permit, sequence 15802
  Match clauses:
    ip address prefix-lists: IPv4-st16386-2621441-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
bdsol-aci32-leaf3# show ip prefix-list  IPv4-st16386-2621441-exc-int-inferred-export-dst
ip prefix-list IPv4-st16386-2621441-exc-int-inferred-export-dst: 3 entries
   seq 1 permit 7.7.7.254/24
   seq 2 permit 192.168.2.254/24 le 32
```

Route-map to redistribute COOP to OSPF

Prefix list permit internal BD Subnet bound to L3 out

Le 32 is direct result of the config of apply HBR to subnet 192.168.2.254/24

/32 specific host route troubleshooting

EPM local EP learning server leaf (usual Behavior)

```
bdsol-aci32-leaf3# show system internal epm  endpoint ip 192.168.2.2

MAC : 0050.56a4.2c7c ::: Num IPs : 1
IP# 0 : 192.168.2.2 ::: IP# 0 flags :  ::: l3-sw-hit: No
Vlan id : 1 ::: Vlan vnid : 8474 ::: VRF name : RD-OSPF:RD
BD vnid : 16089027 ::: VRF vnid : 2621441
Phy If : 0x1a008000 ::: Tunnel If : 0
Interface : Ethernet1/9
Flags : 0x80004c04 ::: sclass : 49156 ::: Ref count : 5
EP Create Timestamp : 12/20/2018 04:03:10.000884
EP Update Timestamp : 12/20/2018 06:47:13.916645
EP Flags : local|IP|MAC|sclass|timer|
```

COOP DB in local spine

```
bdsol-aci32-spine1# show coop  internal  info repo ep key
16089027 00:50:56:A4:2C:7C

Repo Hdr Checksum : 20785
Repo Hdr record timestamp : 12 20 2018 06:23:14 751677189
Repo Hdr last pub timestamp : 12 20 2018 06:23:14 756493038
Repo Hdr last dampen timestamp : 01 01 1970 00:00:00 0
Repo Hdr dampen penalty : 0
Repo Hdr flags : IN_OBJ EXPORT ACTIVE
EP bd vnid : 16089027
EP mac :  00:50:56:A4:2C:7C
flags : 0x80
repo flags : 0x122
Vrf vnid : 2621441
Epg vnid : 0
EVPN Seq no : 0
Remote publish timestamp: 01 01 1970 00:00:00 0
Snapshot timestamp: 12 20 2018 06:23:14 751677189
Tunnel nh : 10.0.88.91
MAC Tunnel  : 10.0.88.91
IPv4 Tunnel : 10.0.88.91
IPv6 Tunnel : 10.0.88.91
ETEP Tunnel : 0.0.0.0
num of active ipv4 addresses : 1
num of anycast ipv4 addresses : 0
num of ipv4 addresses : 1
num of active ipv6 addresses : 0
num of anycast ipv6 addresses : 0
num of ipv6 addresses : 0
Primary Path:
Current published TEP : 10.0.88.91
Backup Path:
BackupTunnel nh : 0.0.0.0
Current Backup (publisher_id): 0.0.0.0
Anycast_flags : 0
Current citizen (publisher_id):
10.0.88.91
Previous citizen : 10.0.88.91
Prev to Previous citizen : 10.0.88.91
Synthetic Flags : 0x5
Synthetic Vrf : 39
Synthetic IP : 4.249.82.13
Tunnel EP entry: 0x2122e1f8
Backup Tunnel EP entry: (nil)
TX Status: COOP_TX_DONE
Leaf 0 Info :
IPv4 Repo Hdr Checksum : 0
IPv4 Repo Hdr record timestamp : 12 20
2018 06:23:14 751677189
IPv4 Repo Hdr last pub timestamp : 12 20
2018 06:23:14 756493038
IPv4 Repo Hdr last dampen timestamp : 01
01 1970 00:00:00 0
IPv4 Repo Hdr dampen penalty : 0
IPv4 Repo Hdr flags : IN_OBJ EXPORT
Real IPv4 EP : 192.168.2.2
```

TX Status	Definition
COOP_TX_NONE	Worker thread has not dispatched notification to Host-Route thread
COOP_TX_PENDING.	Worker thread has dispatched the notification to Host-Route thread but host-route thread has not processed the notification yet.
COOP_TX_IN_PROG	Host-Route thread is processing the notification
COOP_TX_DONE	Host-Route thread has processed the notification and has sent out Host-Route message to BLEAFs.

COOP local Pod EP in BL

In OBJ EXPORT ACTIVE
Is used to tell we need to leak it to urib

```
bdsol-aci32-leaf3# show coop  internal  info repo ep key
16089027 00:50:56:A4:2C:7C


MTS RX OK
Next repo refresh: 1330 seconds 515 ms
Repo Hdr Checksum : 0
Repo Hdr record timestamp : 12 20 2018 06:23:14 751677189
Repo Hdr last pub timestamp : 12 20 2018 06:23:14 756493038
Repo Hdr last dampen timestamp : 01 01 1970 00:00:00 0
Repo Hdr dampen penalty : 0
Repo Hdr flags : IN_OBJ EXPORT ACTIVE
EP bd vnid : 16089027
EP mac :  00:50:56:A4:2C:7C
flags : 0x80
repo flags : 0x122
Vrf vnid : 2621441
Epg vnid : 0
EVPN Seq no : 0
Remote publish timestamp: 01 01 1970 00:00:00 0
Snapshot timestamp: 01 01 1970 00:00:00 0
num of active ipv4 addresses : 1
num of ipv4 addresses : 1
num of active ipv6 addresses : 0
num of ipv6 addresses : 0
Current citizen (publisher_id): 10.0.88.91
Publisher Oracle (Oracle_id): 10.0.128.64
```

```
Leaf 0 Info :
IPv4 Repo Hdr Checksum : 0
IPv4 Repo Hdr record timestamp : 12 20 2018 06:23:14
751677189
IPv4 Repo Hdr last pub timestamp : 12 20 2018 06:23:14
756493038
IPv4 Repo Hdr last dampen timestamp : 01 01 1970 00:00:00 0
IPv4 Repo Hdr dampen penalty : 0
IPv4 Repo Hdr flags : IN_OBJ EXPORT
Real IPv4 EP : 192.168.2.2
Synthetic Flags IPv4 EP : 0
EVPN Seq no : 0
Remote publish timestamp: 01 01 1970 00:00:00 0
Current publisher_id: 0.0.0.0
BackupTunnel nh : 0.0.0.0
MAC Tunnel  : 0.0.0.0
IPv4 Tunnel : 0.0.0.0
IPv6 Tunnel : 0.0.0.0
Current Backup (publisher_id): 0.0.0.0
Synthetic Vrf IPv4 EP: 0
Synthetic IP IPV4 EP : 0.0.0.0
Tunnel EP entry: (nil)
```

COOP remote pod EP in BL (non local pod BL)

In OBJ REMOTE ACTIVE
Is used to tell we DO NOT need to leak it to urib
As the EP is on remote Pod

```
TS RX OK
Next repo refresh: 3238 seconds 529 ms
Repo Hdr Checksum : 0
Repo Hdr record timestamp : 01 01 1970 00:00:00 0
Repo Hdr last pub timestamp : 01 01 1970 00:00:00 0
Repo Hdr last dampen timestamp : 01 01 1970 00:00:00 0
Repo Hdr dampen penaty : 0
Repo Hdr flags : IN_OBJ REMOTE ACTIVE
EP bd vnid : 16089027
EP mac :  00:50:56:A4:28:5D
flags : 0x1080
repo flags : 0x182
Vrf vnid : 2621441
Epg vnid : 0
EVPN Seq no : 0
Remote publish timestamp: 12 20 2018 04:02:39 859086512
Snapshot timestamp: 01 01 1970 00:00:00 0
num of active ipv4 addresses : 1
num of ipv4 addresses : 1
num of active ipv6 addresses : 0
num of ipv6 addresses : 0
Current citizen (publisher_id): 10.0.128.64
Publisher Oracle (Oracle_id): 10.0.128.64
```

```
Leaf 0 Info :
IPv4 Repo Hdr Checksum : 0
IPv4 Repo Hdr record timestamp : 01 01 1970 00:00:00 0
IPv4 Repo Hdr last pub timestamp : 01 01 1970 00:00:00 0
IPv4 Repo Hdr last dampen timestamp : 01 01 1970 00:00:00 0
IPv4 Repo Hdr dampen penalty : 0
IPv4 Repo Hdr flags : IN_OBJ REMOTE
Real IPv4 EP : 192.168.2.1
Synthetic Flags IPv4 EP : 0
EVPN Seq no : 0
Remote publish timestamp: 12 20 2018 04:03:36 527745141
Current publisher_id: 0.0.0.0
BackupTunnel nh : 0.0.0.0
MAC Tunnel  : 0.0.0.0
IPv4 Tunnel : 0.0.0.0
IPv6 Tunnel : 0.0.0.0
Current Backup (publisher_id): 0.0.0.0
Synthetic Vrf IPv4 EP: 0
Synthetic IP IPV4 EP : 0.0.0.0
Tunnel EP entry: (nil)
```

Coop Ip-DB on leaf (new cli)

```
bdsol-aci32-leaf3# show coop internal  info  ip-db


IP address : 192.168.2.2
Vrf : 2621441
Flags : 0
EP bd vnid : 16089027
EP mac :  00:50:56:A4:2C:7C
Publisher Id : 10.0.88.91
Record timestamp : 12 20 2018 04:14:40 653649250
Publish timestamp : 12 20 2018 04:14:40 655134431
Remote publish timestamp: 01 01 1970 00:00:00 0

------------------------------
```

- **Displays the EP-IP added in the IP-DB database in border-leaf.**
- **Eps learned from remote pods should not get programmed in this DB.**
- **Eps learned from remote leaf should get programmed here.**
- **Eps learned from any leaf in the local pod get programmed in this db.**

COOP citizen log on BL

```
bdsol-aci32-leaf3# show coop  internal  trace-detail-uc | egrep "192.168.2.2" | egrep -v 254
408) 2018 Dec 20 13:20:32.723693 TID 01:coop_cz_ipv4_update_add_leaf:1827: Processing add leaf 192.168.2.2
428) 2018 Dec 20 13:20:32.723562 TID 01:coop_cz_ipv4_update_add_leaf:1827: Processing add leaf 192.168.2.2
448) 2018 Dec 20 13:20:32.723171 TID 01:coop_cz_ipv4_update_add_leaf:1827: Processing add leaf 192.168.2.2
468) 2018 Dec 20 13:20:32.719436 TID 01:coop_cz_ipv4_update_add_leaf:1827: Processing add leaf 192.168.2.2
538) 2018 Dec 20 13:20:32.715713 TID 01:coop_citizen_publish_ep:1049: ADD EP msg <16089027, 00:50:56:A4:2C:7C> #IPs: 1 <2621441,
192.168.2.2> with trans_id 4947 rec_ts: 1545286994:751677189 pub_ts: 1545
286994:756493038 tep_ip:10.0.88.91 Anycast service: FALSE
```

COOP rib-leak on BL

- As BD is host-route enabled, so spine will download all Eps under the BD to the border-leaf.
- These Eps may include private subnets.
- This route-map and prefix-lists are used by COOP citizen to decide what routes to leak to URIB

```
bdsol-aci32-leaf3# show coop internal host-route bridge-domain 16089027
Host-Based Routing BD Details:
bd-vnid:16089027, flags:0x1
host-route: Enabled
vrf[0]: RD-OSPF:RD, vnid:2621441 flags:0x1
policy af:IPv4 name:coop-ribleak-2621441 cfg:1 hdl:251567012
policy af:IPv6 name:coop-ribleak-2621441 cfg:1 hdl:251565556

bdsol-aci32-leaf3# show route-map coop-ribleak-2621441
route-map coop-ribleak-2621441, permit, sequence 1
  Match clauses:
    ip address prefix-lists: IPv4-coop-ribleak-2621441-16089027
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
route-map coop-ribleak-2621441, deny, sequence 20000
  Match clauses:
  Set clauses:
bdsol-aci32-leaf3#
bdsol-aci32-leaf3#
bdsol-aci32-leaf3# show ip prefix-list IPv4-coop-ribleak-2621441-16089027
ip prefix-list IPv4-coop-ribleak-2621441-16089027: 1 entries
   seq 1 permit 192.168.2.0/24 le 32
```

Route-map permits
BD subnet (who are public)
To be leaked from coop to
Rib with Tag for loop prevention

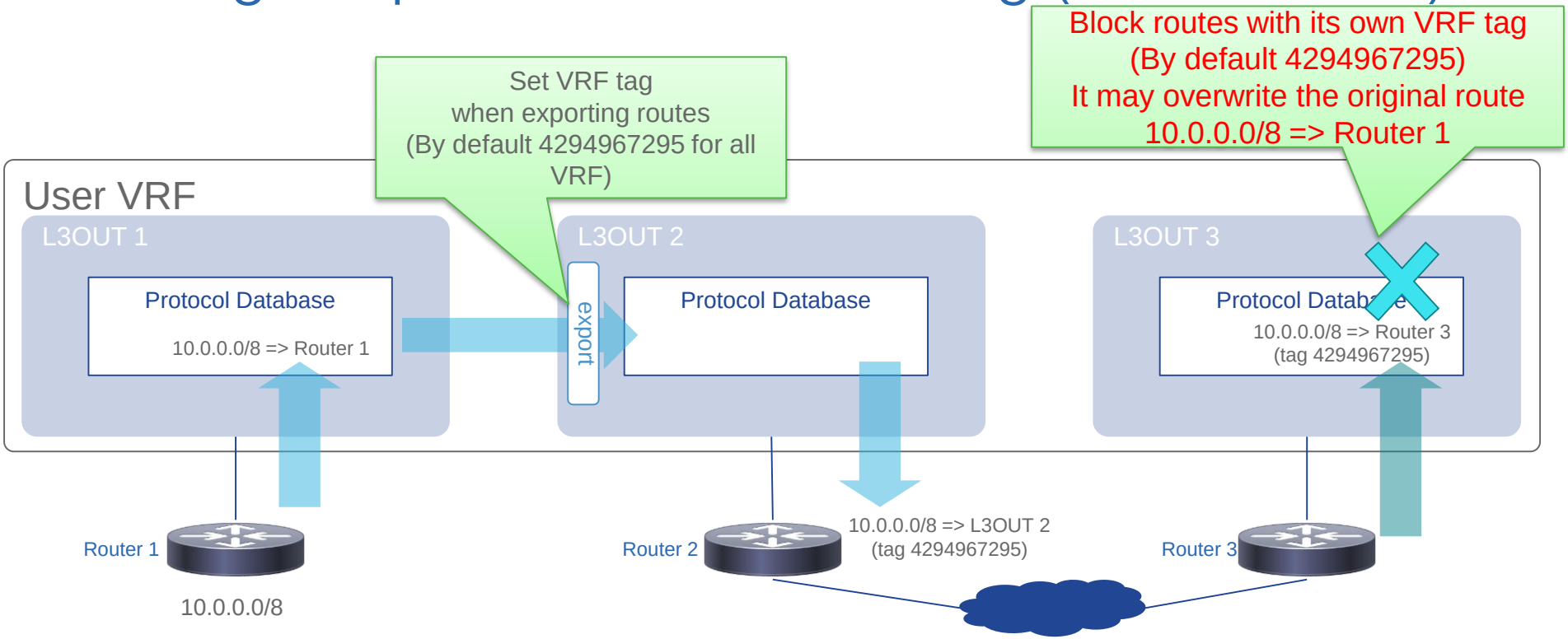
Resulting routing table on BL (+ ospf here)

```
bdsol-aci32-leaf3# show ip route 192.168.2.2 vrf RD-OSPF:RD

192.168.2.2/32, ubest/mbest: 1/0, pervasive, redist-only, pending ufdm
    *via Null0, [2/0], 00:49:52, coop, coop, tag 4294967295 (redist-only)
bdsol-aci32-leaf3# show ip ospf  database external 192.168.2.2 detail  vrf RD-OSPF:RD
        OSPF Router with ID (10.58.212.3) (Process ID default VRF RD-OSPF:RD)

                Type-5 AS External Link State
   LS age: 1541
   Options: 0x2 (No TOS-capability, No DC)
   LS Type: Type-5 AS-External
   Link State ID: 192.168.2.2 (Network address)
   Advertising Router: 10.58.212.3
   LS Seq Number: 0x80000007
   Checksum: 0xf44d
   Length: 36
   Network Mask: /32
         Metric Type: 2 (Larger than any link state path)
         TOS: 0
         Metric: 1
         Forward Address: 0.0.0.0
         External Route Tag: 4294967295
   LS age: 1550
   Options: 0x2 (No TOS-capability, No DC)
   LS Type: Type-5 AS-External
   Link State ID: 192.168.2.2 (Network address)
   Advertising Router: 10.58.212.4
   LS Seq Number: 0x80000007
   Checksum: 0xee52
   Length: 36
   Network Mask: /32
         Metric Type: 2 (Larger than any link state path)
         TOS: 0
         Metric: 1
         Forward Address: 0.0.0.0
         External Route Tag: 4294967295
```

Note the pending ufdm
(route is not programmed to ufib)

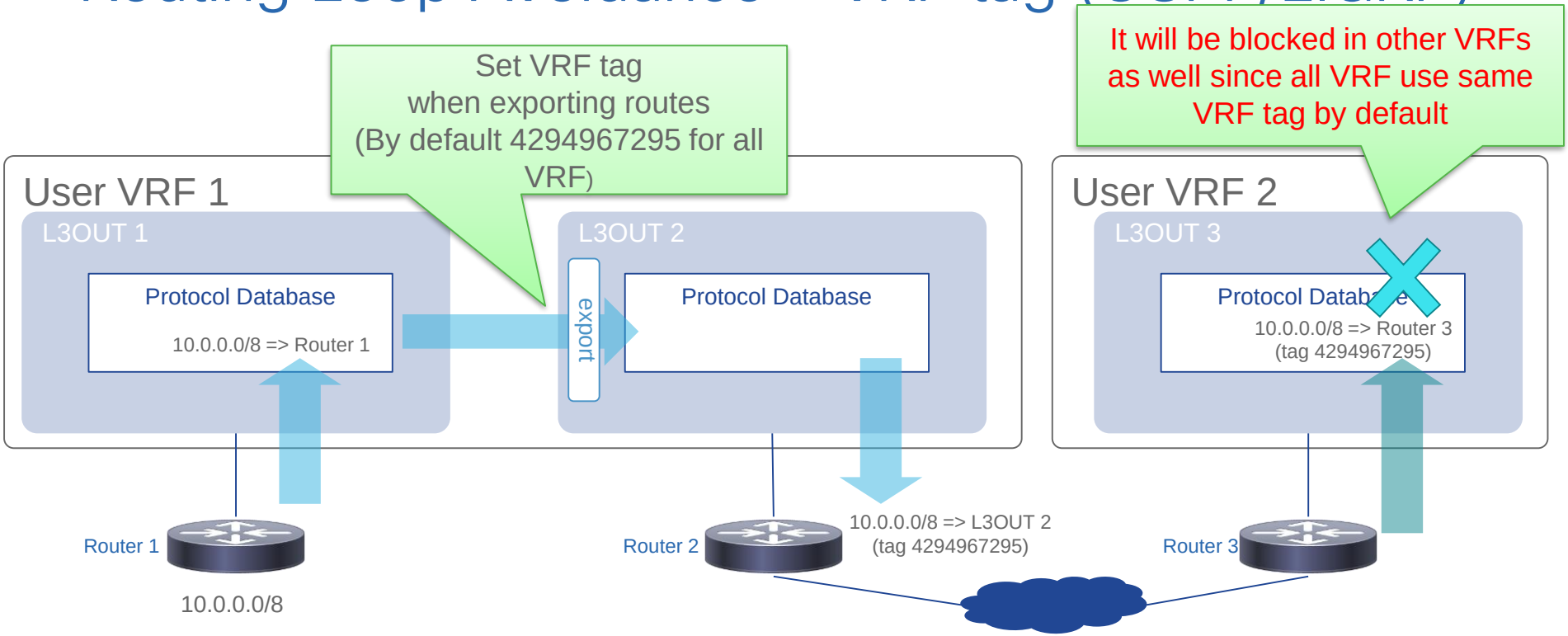
Routing loop Avoidance

Routing Loop Avoidance – VRF tag (OSPF/EIGRP)



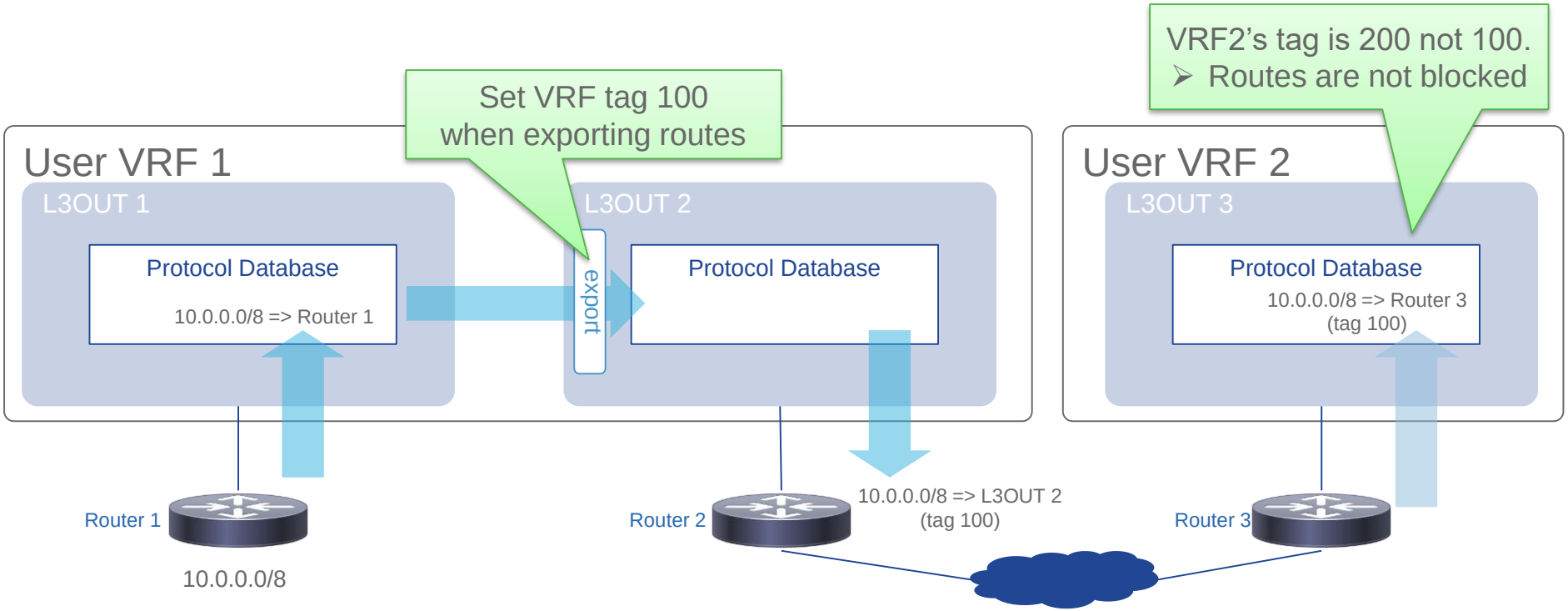
※ VRF tagging for exported routes and blocking routes with VRF tag are always enabled

Routing Loop Avoidance – VRF tag (OSPF/EIGRP)



※ VRF tagging for exported routes and blocking routes with VRF tag are always enabled

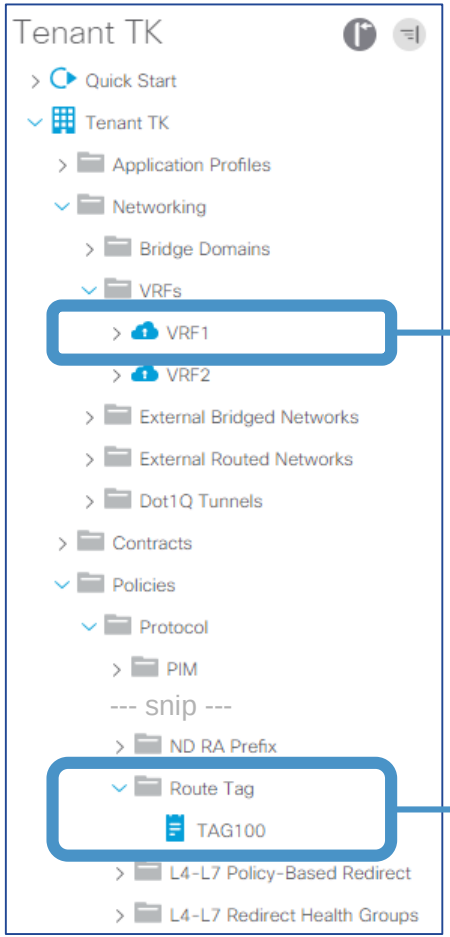
Routing Loop Avoidance – VRF tag (OSPF/EIGRP)

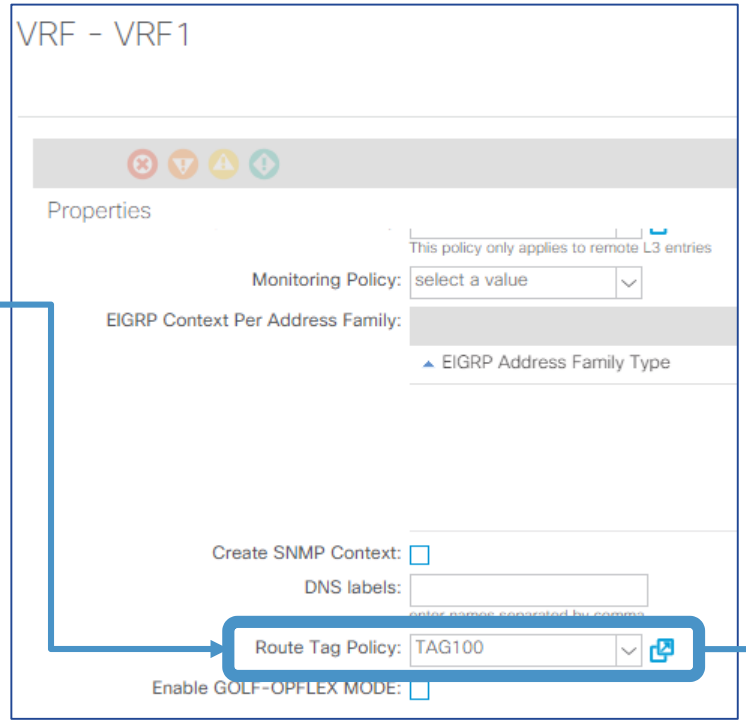


VRF 1 tag – 100
VRF 2 tag – 200

Please make sure to do this only when routes need to be advertised back to ACI

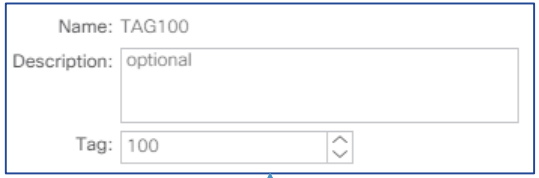
Routing Loop Avoidance – VRF tag (OSPF/EIGRP)





VRF tag can be configured per VRF

In this example VRF1's tag is 100



※ VRF tag is only for OSPF and EIGRP

Routing Loop Avoidance – VRF tag (OSPF/EIGRP)

Route Tag Policy: TAG100

IP Address: 10.0.0.0/8
address/mask

Scope:
- [x] Export Route Control Subnet
- [ ] Import Route Control Subnet
- [ ] External Subnets for the External EPG
- [ ] Shared Route Control Subnet
- [ ] Shared Security Import Subnet

```
leaf# show ip ospf vrf TK:VRF1 | egrep 'route-map|Redis'
 Table-map using route-map exp-ctx-2097152-deny-external-tag
 Redistributing External Routes from
   static route-map exp-ctx-st-2097152
   direct route-map exp-ctx-st-2097152
   eigrp route-map exp-ctx-proto-2097152
   bgp route-map exp-ctx-proto-2097152
```

Always there with VRF tag

note:

Import Route Control Subnet is added here after VRF tag deny rule if Import Route Control Enforcement is enabled.

xport routes with VRF tag

```
leaf# show route-map exp-ctx-proto-2097152
route-map exp-ctx-proto-2097152, permit, sequence 15802
  Match clauses:
    ip address prefix-lists: IPv4-proto49158-2097152-exc-ext-inferred-export-
dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 100
```

```
leaf# show ip prefix-list IPv4-proto49158-2097152-exc-ext-inferred-export-dst
ip prefix-list IPv4-proto49158-2097152-exc-ext-inferred-export-dst: 1 entries
   seq 1 permit 10.0.0.0/8
fab3-p1-leaf3#
```

Block routes with VRF tag

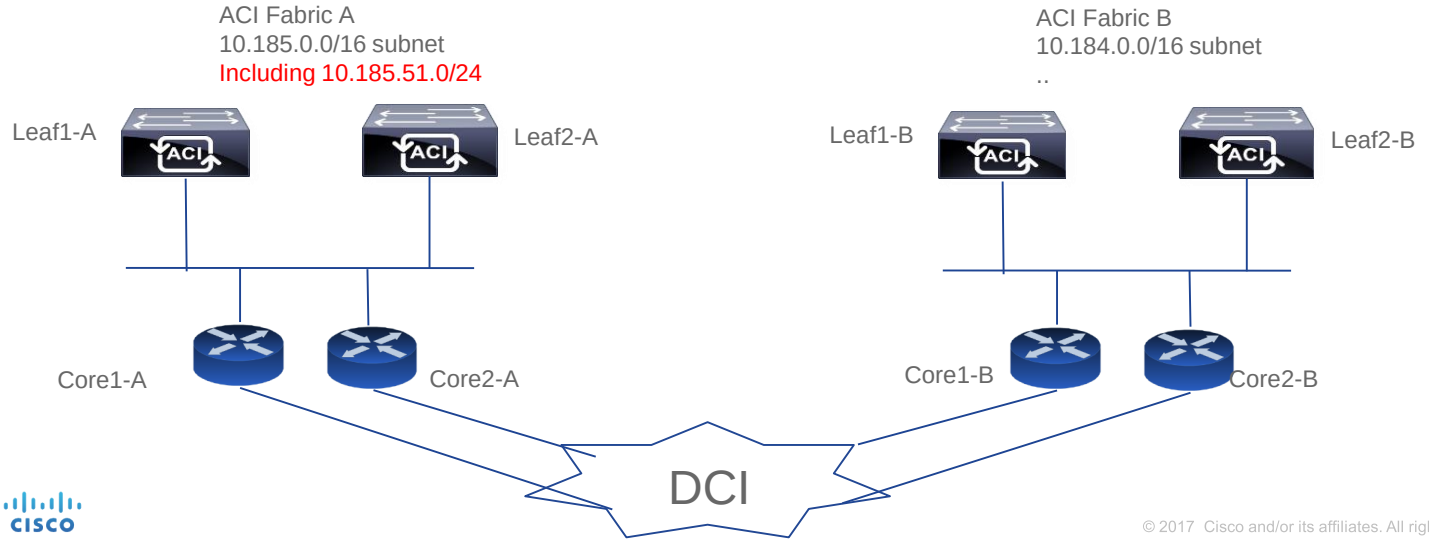
```
leaf# show route-map exp-ctx-2097152-deny-external-tag
route-map exp-ctx-2097152-deny-external-tag, deny, sequence 1
  Match clauses:
    tag: 100
  Set clauses:
route-map exp-ctx-2097152-deny-external-tag, permit, sequence
200
  Match clauses:
  Set clauses:
```

※ route-map and VRF tag are per VRF

Routing loop Case Study

Case Study - Topology

- Two different ACI fabric
- All leaves and core in OSPF Area 0
- BD subnet advertised by Fabric A in range 10.185/16 and by fabric B in range 10.184/16
- All subnet are LSA type 5 injected by ACI into OSPF domain

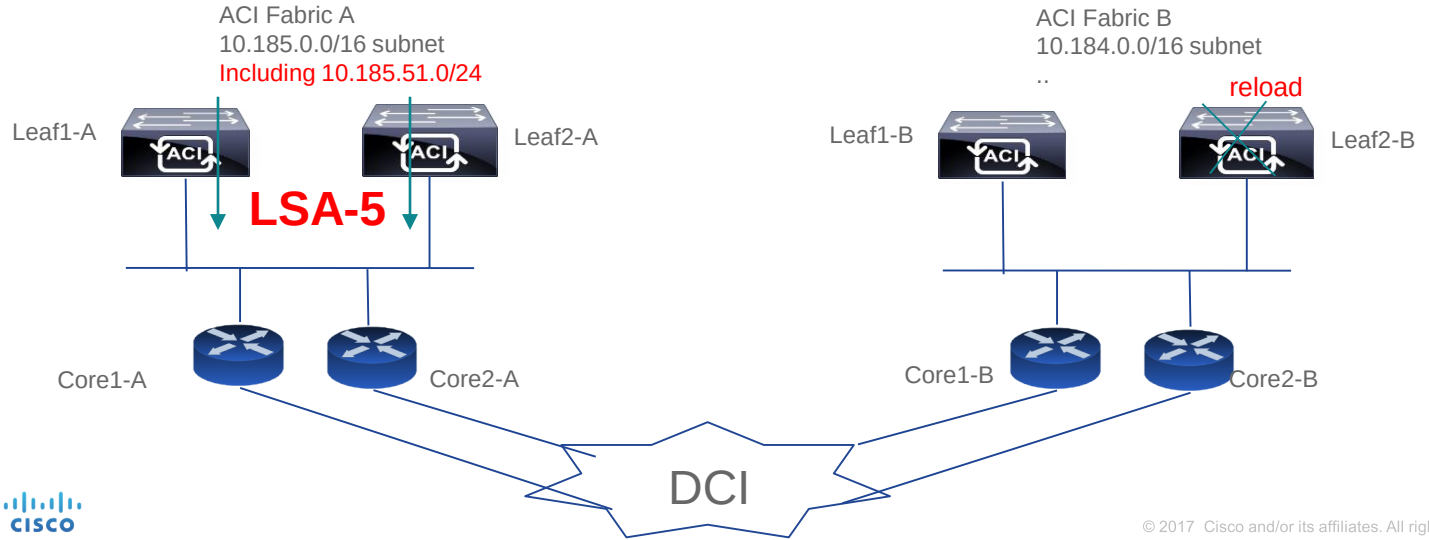


Problem description

- Reload of leaf1-B led to loss of routing for prefix 10.185.xxx in core1-A and core2-A !!!

Seen on coreA when leafB reload :

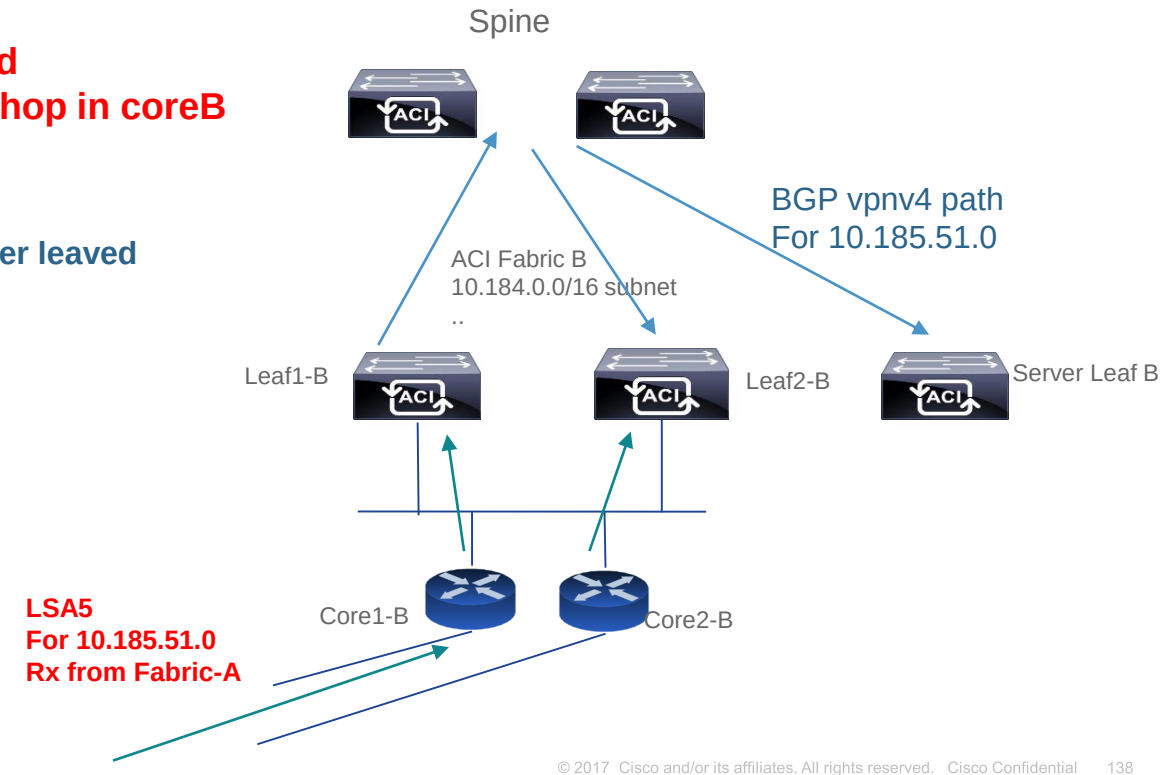
"ospf-1": 10.185.51.0/24, deleting nh 10.185.5.30%Vlan400, metric [110/20] route-type type-2 tag 0x00000000 flags 0x00000000



What happens in ACI fabric-B ?

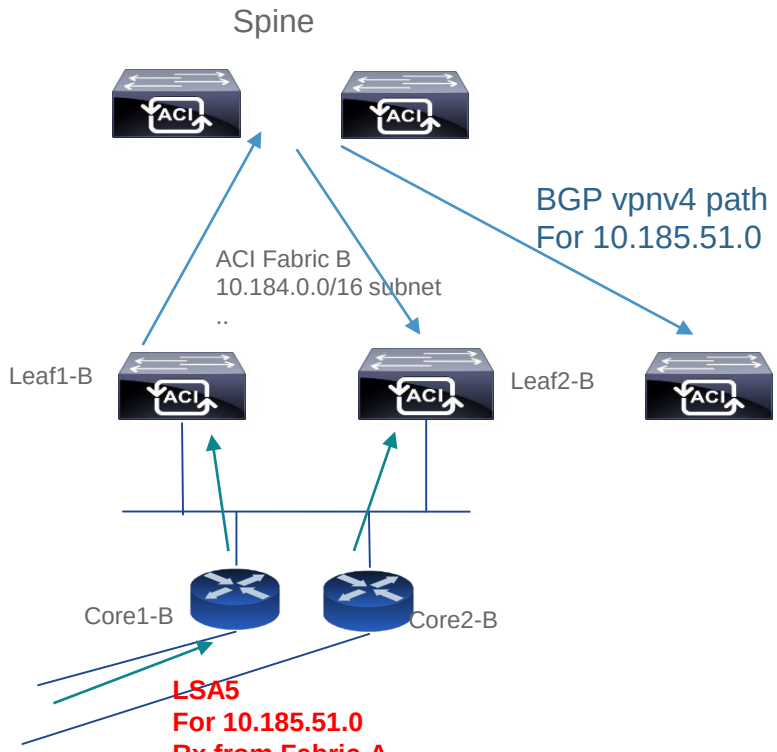
LSA5 is received from Fabric-A
Flooded in OSPF area and received
By both leaf in Fabric-B with Next-hop in coreB

Leaf-B redistribute the OSPF prefix to
BGP VPNv4 to send in ACI fabric to other leaved
Through spine RR



Result entry in Fabric-B

```
pod2-leaf2# show bgp vpnv4 unicast 10.185.51.0 vrf DC:DC
BGP routing table information for VRF overlay-1, address family VPNv4
Unicast
Route Distinguisher: 10.0.168.93:26    (VRF DC:DC)
BGP routing table entry for 10.185.51.0/24, version 61
Paths: (2 available, best #2)
Flags: (0x80c0002 0x000004) on xmit-list, is not in urib, exported, is in
objstore
  vpn: version 511, (0x100002) on xmit-list
Multipath: eBGP iBGP

  VPN AF advertised path-id 2
  Path type: internal 0xc0000018 0x40040 ref 1, path is valid, not best
reason: Weight
            Imported from 10.0.168.95:35:10.185.51.0/24
 AS-Path: NONE, path sourced internal to AS
!!!!! PATH FROM iBGP coming from neighbor leaf
    10.0.168.95 (metric 3) from 10.0.168.92 (10.0.168.92)
      Origin incomplete, MED 20, localpref 100, weight 0
      Received label 0
      Received path-id 1
      Extcommunity:
          RT:101:2097157
          VNID:2097157
          COST:pre-bestpath:162:110
      Originator: 10.0.168.95 Cluster list: 10.0.168.92

  Advertised path-id 1, VPN AF advertised path-id 1
  Path type: redist 0x408 0x40001 ref 0, path is valid, is best path
!!!!! PATH FROM OSPF
  AS-Path: NONE, path locally originated
    0.0.0.0 (metric 0) from 0.0.0.0 (10.0.168.93)
      Origin incomplete, MED 20, localpref 100, weight 32768
      Extcommunity:
          RT:101:2097157
          VNID:2097157
          COST:pre-bestpath:162:110
```



What happens when leaf reload ?

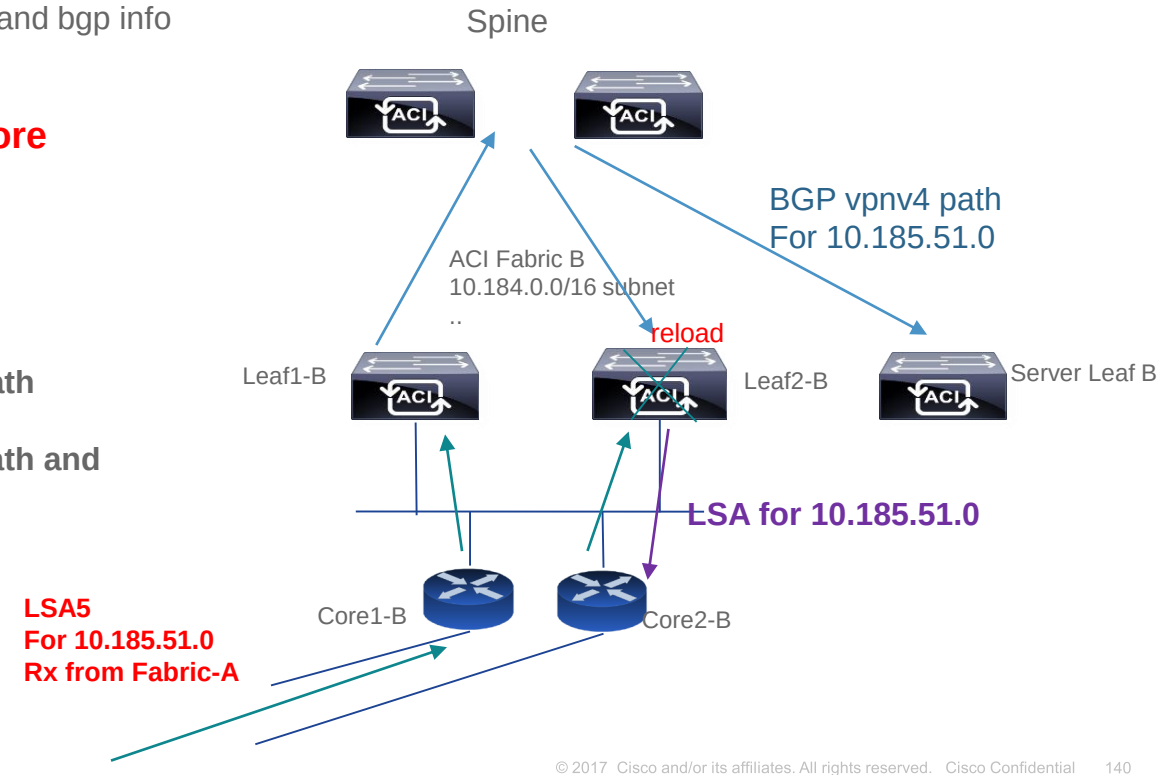
Leaf2-B got reloaded, hence lose all ospf and bgp info
What happens when it comes back up ?

It will get both OSPF path from core
And BGP path from leaf1-B

He may get OSPF before BGP or
BGP before OSPF

Assuming BGP path is Rx before OSPF path

Leaf 2-B may redistribute to OSPF BGP path and
Inject it in Area 0 (Based on route-map) !



Reloading leaf output

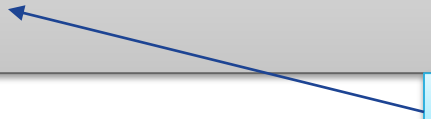
Route-map that determine What we redistribute in ospf area

We have a prefix-list that allow everything !!!

```
pod2-leaf2# show ip ospf  vrf DC:DC
..
   Area BACKBONE(0.0.0.0)
        Area has existed for 00:20:55
        Interfaces in this area: 2 Active interfaces: 2
        Passive interfaces: 1  Loopback interfaces: 1
        No authentication available
        SPF calculation has run 9 times
         Last SPF ran for 0.002716s
        Area ranges are
        Area-filter in 'exp-ctx-proto-2097157'
        Number of LSAs: 14, checksum sum 0x62e65

The above route-map is used to decide what we redistribute to OSPF :
pod2-leaf2# show  route-map exp-ctx-proto-2097157
route-map exp-ctx-proto-2097157, permit, sequence 19801
  Match clauses:
    ip address prefix-lists: IPv4-proto49154-2097157-agg-ext-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
pod2-leaf2# show ip prefix-list IPv4-proto49154-2097157-agg-ext-inferred-export-dst
ip prefix-list IPv4-proto49154-2097157-agg-ext-inferred-export-dst: 1 entries
   seq 1 permit 0.0.0.0/0 le 32
```

Resulting wrong routing entry in core

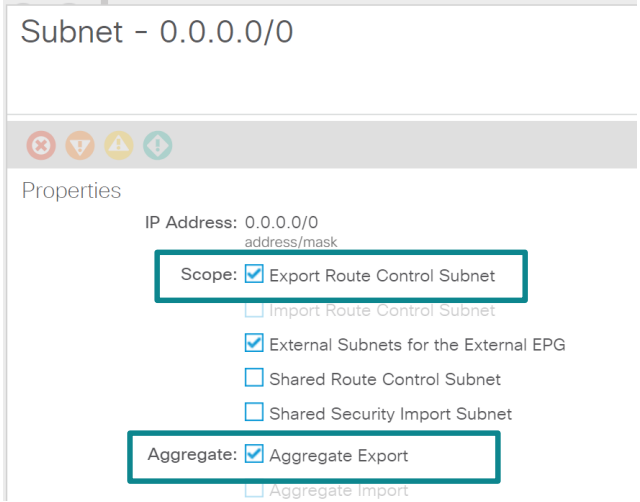
```
CORE# show ip route 10.185.51.0/24 vrf DC
IP Route Table for VRF "BT"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>
10.185.51.0/24, ubest/mbest: 1/0
    *via 10.184.3.4, Vlan883, [110/1], 00:00:01, ospf-1, type-2, tag 4294967295
bdsol-n6004-02#
```



Wrong next-hop and route tag from ACI

Problem correct itself after couple of seconds (when reloading leaf got correct LSA it stops send BGP path to OSPF)

Root case



- **Subnet flag in l3 out , export route control with aggregate export is set for al routes**
- External subnet for external EPG is used for what we will accept from that L3 out
- Export route control subnet is used to send transit routing to that l3 out
- They should normally not be set together

If 0.0.0.0/0 is external subnet
If we need transit routing, we need to set specific prefix from different L3 out as export route control

Export Route control – What to remember

- Aci do not have any mechanism such as SOO in MPLS VPN to prevent a route received from a site (a l3 out here) to send it back to the same site on a different leaf on same l3 out
- Export route control SHOULD exclude any subnet which may be receive on that L3 out !

MP-BGP review
iBGP in ACI detail

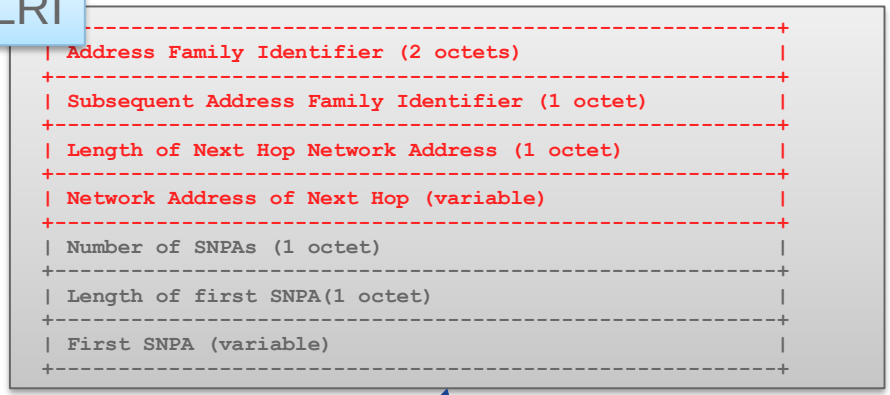
MP-BGP update packet format (RFC 4760)

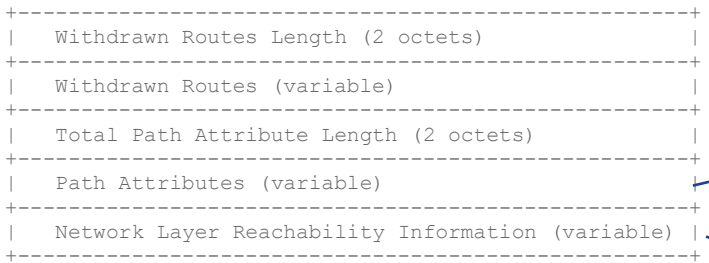
MP-BGP allows to send
Any type of protocol in the NLRI.
MP_REACH_NLRI attribute
Specifies next-hop and which address family (AFI/SAFI)

Extended community carries the Route-target information

Route Distinguisher are part of the NLRI prefix

NLRI





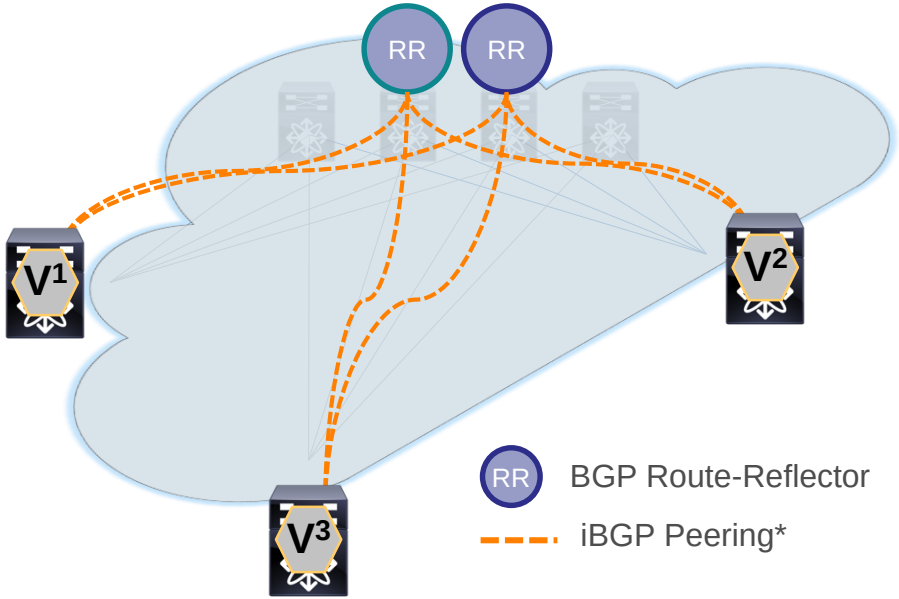
Attribute can be :
ORIGIN, AS_PATH, NEXT_HOP,MED ,
LOCAL_PREF, ATOMIC_AGGREGATE,
AGGREGATOR,COMMUNITY,
EXTENDED_COMMUNITY (contains Route-target amont other things ,
MP_REACH_NLRI

NLRI for VPNv4 for example is extended to contains :
- Label stack
- Route Distinguisher
- IPv4 address and masks,

Multiprotocol BGP (MP-BGP) Primer

For Your Reference

- Multiprotocol BGP (MP-BGP)
- Extension to Border Gateway Protocol (BGP) - RFC 4760
- VPN Address-Family:
  - Allows different types of address families (e.g. VPNv4, VPNv6, L2VPN EVPN (RFC 7432), MVPN)
  - Information transported across single BGP peering



*eBGP supported without BGP Route-Reflector

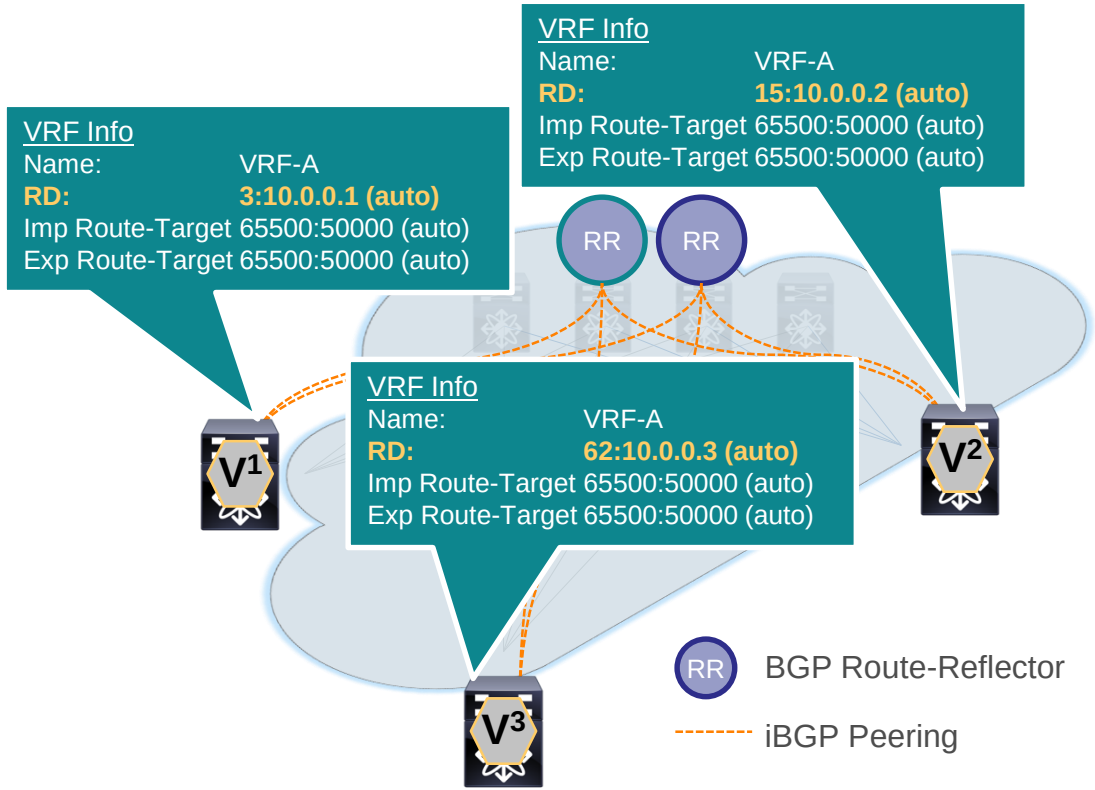
IETF

https://tools.ietf.org/html/draft-ietf-bess-evpn-overlay-04

Multiprotocol BGP (MP-BGP) Primer

For Your Reference

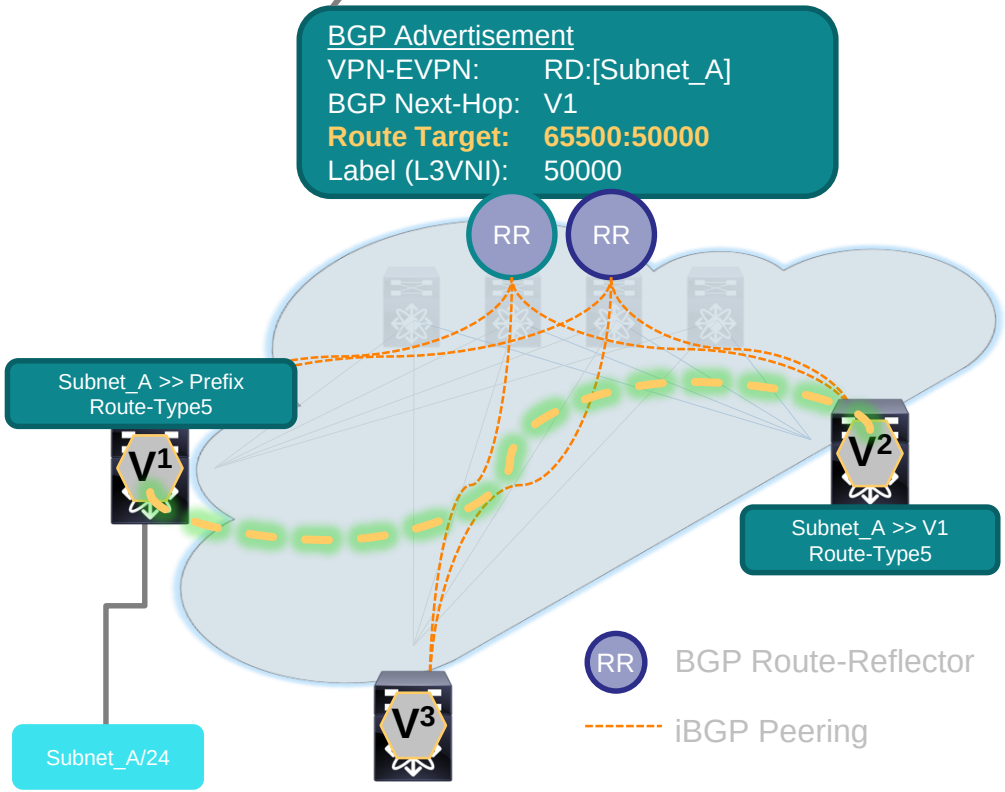
- VPN segmentation for tenant routing (Multi-Tenancy)
  - Route Distinguisher (RD)
  - 8-byte field of VRF parameters
  - value to make VPN prefix unique:
    - RD + VPN prefix



Multiprotocol BGP (MP-BGP) Primer

For Your Reference

- VPN Segmentation for tenant routing (Multi-Tenancy)
- Selective distribute VPN routes - Route Target (RT)
  - 8-byte field of VRF parameter
  - unique value to define the import/export rules for VPN prefix

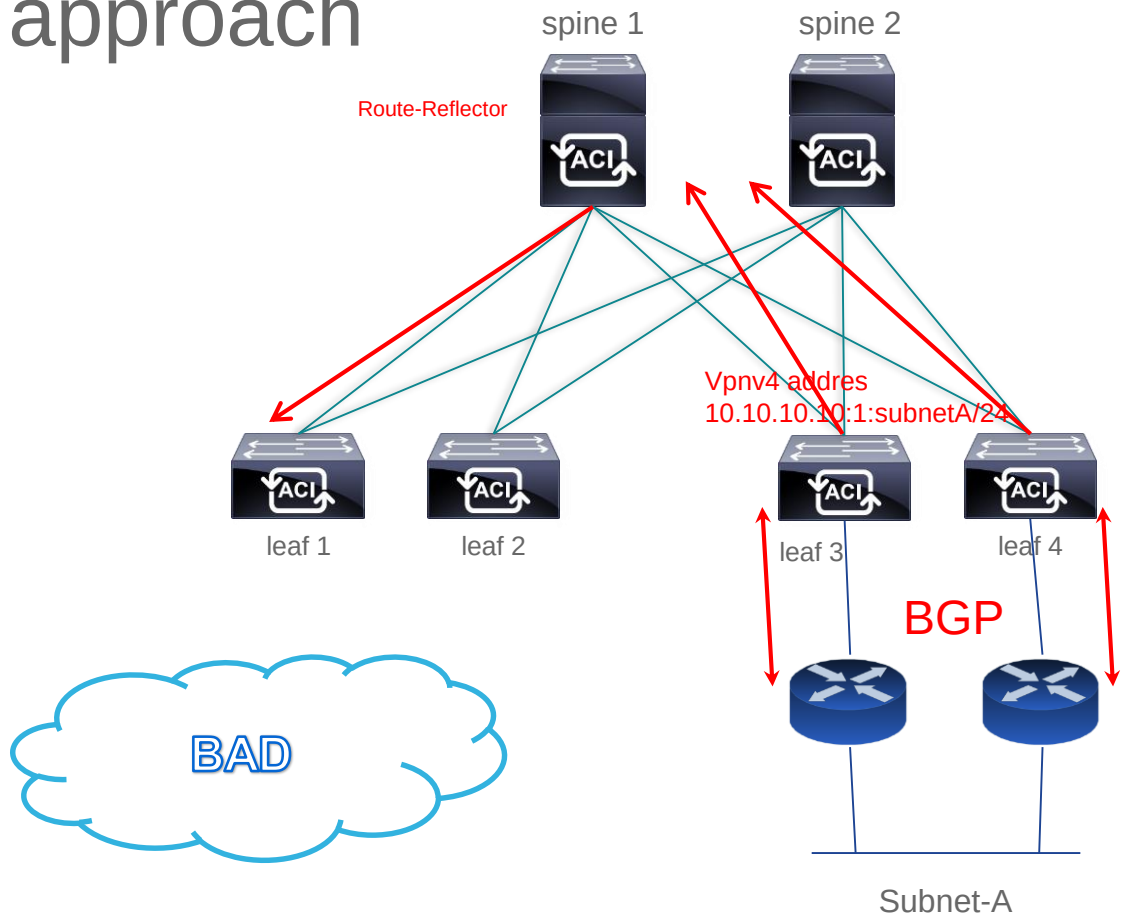


Route Distinguisher in ACI

- In regular vpnv4 network we can either use same RD on every node for the same vrf or use a unique RD per node in the same VRF.

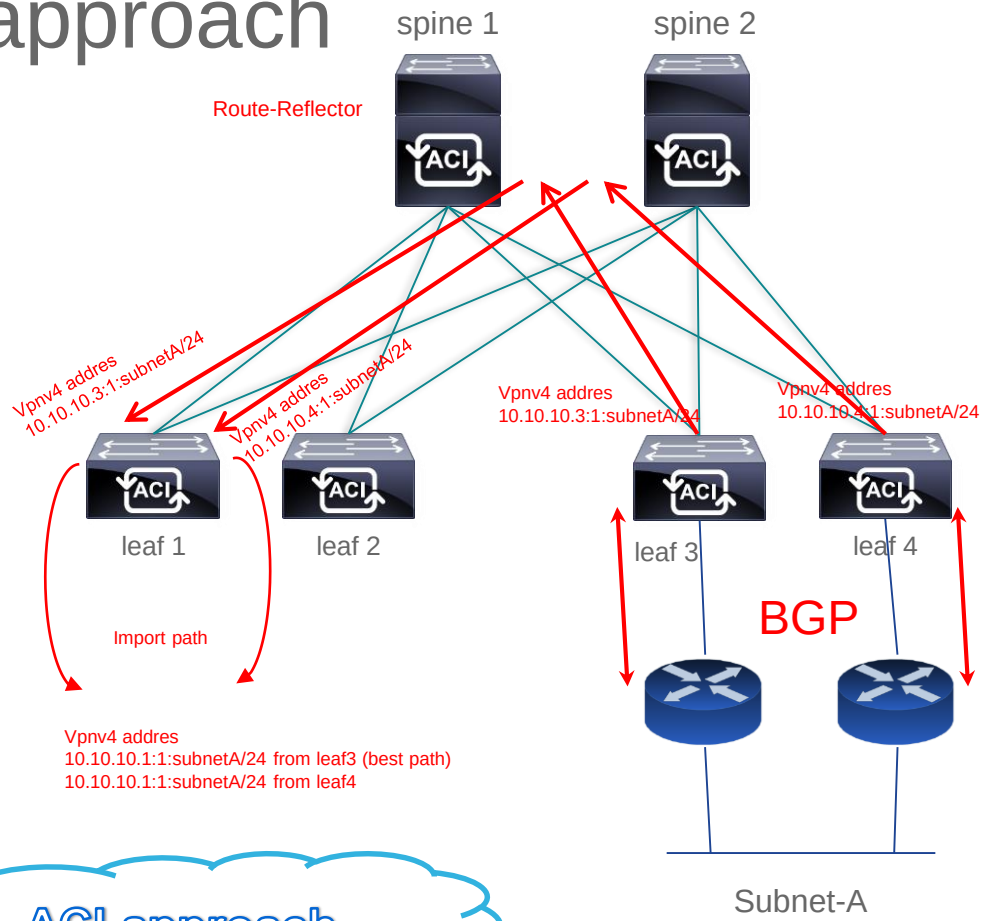
Same RD per node approach

- We suppose all leaf in vrf uses same RD (say 10.10.10.10:1)
- Both Leaf 3 and 4 do send
- Vpnv4 10.10.10.10:1:subnetA/24 to Route Reflector
- Route reflector gets twice same exact vpnv4 prefix and apply best path, it only reflect one Path to Leaf 1 (say from leaf3)
- Leaf 1 gets vpnv4 address with NH leaf3 TEP.
- If leaf3 goes down no backup till we receive new prefix from RR



Different RD per node approach

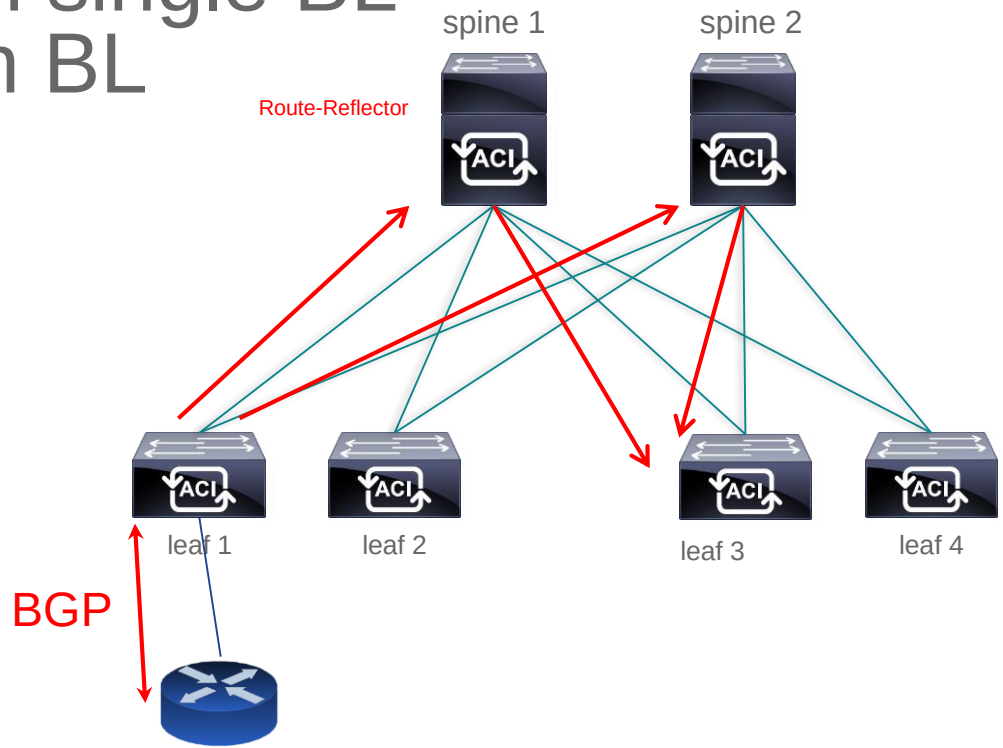
- We suppose all leaf in vrf uses same RD (say 10.10.10.leafID:1)
- Leaf 3 do send Vpnv4 10.10.10.3:1:subnetA/24 to Route Reflector
- Leaf 4 do send Vpnv4 10.10.10.4:1:subnetA/24 to Route Reflector
- Route reflector gets two different vpnv4 prefix and reflect both to leaf 1
- Leaf 1 gets two vpnv4 address with one from leaf3, one from leaf4. It does import both to its own RD (10.10.10.1:1:subnetA/24)
- Then leaf 1 applies best path selection and only install one in RIB (say leaf 3)
- If leaf3 goes down we can select directly back path to leaf4



ACI approach

Example 1 – BGP with single BL Prefix received on non BL

- Leaf 1 has eBGP connection and gets prefix 172.16.1.0/24 in vrf L3:L3.Let's look on leaf 3
- Path send to both spine , each spine reflect it.




```
pod2-leaf3# show bgp vpnv4 unicast 172.16.1.0 vrf overlay-1  | egrep "Route Dist"
Route Distinguisher: 10.0.168.91:4    (VRF L3:L3)
Route Distinguisher: 10.0.168.95:10
```

```
pod2-leaf3# acidiag fnvread | egrep "leaf1|leaf3"
     101       pod2-leaf1      SAL1820SMHV     10.0.168.95/32    leaf       1          active   0
     103       pod2-leaf3      SAL1818RUHM     10.0.168.91/32    leaf       1          active   0
```

Example 1 - Reading BGP – Rx path

```
pod2-leaf3# show bgp vpnv4 unicast 172.16.1.0 vrf overlay-1
BGP routing table information for VRF overlay-1, address family VPNv4
Unicast
Route Distinguisher: 10.0.168.95:10
BGP routing table entry for 172.16.1.0/24, version 943
Paths: (2 available, best #1)
Flags: (0x000002) on xmit-list, is not in urib
Multipath: eBGP iBGP

  Advertised path-id 1
  Path type: internal, path is valid, is best path
  AS-Path: 100 99 , path sourced external to AS
    10.0.168.95 (metric 3) from 10.0.168.92 (10.0.168.92)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 0
      Extcommunity:
          RT:101:2785280
          VNID:2785280
      Originator: 10.0.168.95 Cluster list: 10.0.168.92

  Path type: internal, path is valid, not best reason: Neighbor Address
  AS-Path: 100 99 , path sourced external to AS
    10.0.168.95 (metric 3) from 10.0.168.94 (10.0.168.94)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 0
      Extcommunity:
          RT:101:2785280
          VNID:2785280
      Originator: 10.0.168.95 Cluster list: 10.0.168.94

  Path-id 1 not advertised to any peer
```

This is the prefix we receive .
It is vpnv4 prefix in RD 10.0.168.95:10 (leaf1 RD) for 172.16.1.0/24

We receive that path twice (one from each Route-Reflector)

```
pod2-leaf3# acidiag fnvread | egrep "spine"
 201     pod2-spine1     10.0.168.92/32   spine     1          active    0
 202     pod2-spine2     10.0.168.94/32   spine     1          active    0
```

In each we have same route-target : 101:2785280

BGP got two path identical (same RD) hence it runs best path and choose the first

Example 1 - Reading BGP (cont.) imported path

```
BGP routing table information for VRF overlay-1, address family VPNv4
Unicast
Route Distinguisher: 10.0.168.91:4    (VRF L3:L3)
BGP routing table entry for 172.16.1.0/24, version 341
Paths: (1 available, best #1)
Flags: (0x08001a) on xmit-list, is in urib, is best urib route
  vpn: version 944, (0x100002) on xmit-list
Multipath: eBGP iBGP

  Advertised path-id 1
  Path type: internal, path is valid, is best path
             Imported from 10.0.168.95:10:172.16.1.0/24
  AS-Path: 100 99 , path sourced external to AS
    10.0.168.95 (metric 3) from 10.0.168.92 (10.0.168.92)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 0
      Extcommunity:
          RT:101:2785280
          VNID:2785280
      Originator: 10.0.168.95 Cluster list: 10.0.168.92

  VRF advertise information:
  Path-id 1 not advertised to any peer

  VPN AF advertise information:
```

This is the prefix we imported in our own RD 10.0.168.91:4 (here locally we know it is vrf L3:L3) .
We see it is imported prefix from 10.0.168.95:10:172.16.1.0/24 (leaf1-RD:subnetA/24)

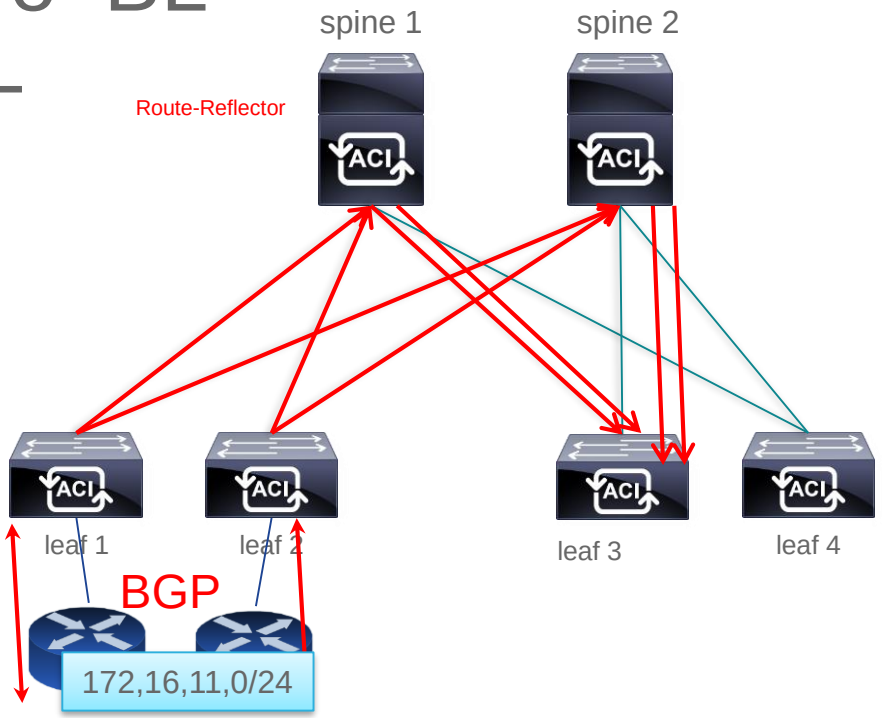
Note that only the best path from previous slide is imported (from Spine with ip 10.0.168.92)

We can import it because the Route-Target do match what we allow for import in the vrf

```
pod2-leaf3# show bgp process vrf L3:L3 | egrep -A 1
"RT"
    Export RT list:
        101:2785280
    Import RT list:
        101:2785280

RT is AS nr + ctx vnid
```

Example 2 – BGP with two BL Prefix received on non BL

- Leaf 1 and leaf 2 has eBGP connection and gets prefix 172.16.11.0/24 in vrf LB:LB.Let's look on leaf 3
- Path send to both spine from both leaf , each spine reflect it.




```
pod2-leaf3# show bgp vpnv4 unicast 172.16.11.0 vrf
overlay-1  | egrep "Route Dist"
Route Distinguisher: 10.0.168.91:11    (VRF LB:LB)
Route Distinguisher: 10.0.168.93:21
Route Distinguisher: 10.0.168.95:31
```

```
pod2-leaf3# acidiag fnvread | egrep "leaf1|leaf3"
     101       pod2-leaf1      SAL1820SMHV     10.0.168.95/32    leaf       1          active   0
     102       pod2-leaf2      SAL1816QVBC     10.0.168.93/32    leaf       1          active   0
     103       pod2-leaf3      SAL1818RUHM     10.0.168.91/32    leaf       1          active   0
```

Example 2 – BGP path on leaf 3

Path received with RD of leaf 1

```
BGP routing table information for VRF overlay-1, address
family VPNv4 Unicast
Route Distinguisher: 10.0.168.95:31
BGP routing table entry for 172.16.11.0/24, version 1197
Paths: (2 available, best #2)
Flags: (0x000002) on xmit-list, is not in urib
Multipath: eBGP iBGP

  Path type: internal, path is valid, not best reason:
Neighbor Address
  AS-Path: 99 , path sourced external to AS
    10.0.168.95 (metric 3) from 10.0.168.94 (10.0.168.94)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 0
      Extcommunity:
          RT:101:2588672
          VNID:2588672
      Originator: 10.0.168.95 Cluster list: 10.0.168.94

  Advertised path-id 1
  Path type: internal, path is valid, is best path
  AS-Path: 99 , path sourced external to AS
    10.0.168.95 (metric 3) from 10.0.168.92 (10.0.168.92)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 0
      Extcommunity:
          RT:101:2588672
          VNID:2588672
      Originator: 10.0.168.95 Cluster list: 10.0.168.92

  Path-id 1 not advertised to any peer
```

In each RD as we have Two path with same vpnv4 address(one from each RR), hence we run Best path selection for each RD.

Path received with RD of leaf 2

```
BGP routing table information for VRF overlay-1, address
family VPNv4 Unicast
Route Distinguisher: 10.0.168.93:21
BGP routing table entry for 172.16.11.0/24, version 1018
Paths: (2 available, best #2)
Flags: (0x000002) on xmit-list, is not in urib
Multipath: eBGP iBGP

Path type: internal, path is valid, not best reason:
ighbor Address
AS-Path: 99 , path sourced external to AS
  10.0.168.93 (metric 3) from 10.0.168.94 (10.0.168.94)
    Origin IGP, MED not set, localpref 100, weight 0
    Received label 0
    Extcommunity:
        RT:101:2588672
        VNID:2588672
    Originator: 10.0.168.93 Cluster list: 10.0.168.94

  Advertised path-id 1
  Path type: internal, path is valid, is best path
  AS-Path: 99 , path sourced external to AS
    10.0.168.93 (metric 3) from 10.0.168.92 (10.0.168.92)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 0
      Extcommunity:
          RT:101:2588672
          VNID:2588672
      Originator: 10.0.168.93 Cluster list: 10.0.168.92

  Path-id 1 not advertised to any peer
```

Example 2 – BGP path on leaf 3 (cont.)

```
pod2-leaf3# show bgp vpnv4 unicast 172.16.11.0 vrf overlay-1
BGP routing table information for VRF overlay-1, address family VPNv4 Unicast
Route Distinguisher: 10.0.168.91:11    (VRF LB:LB)
BGP routing table entry for 172.16.11.0/24, version 61
Paths: (2 available, best #2)
Flags: (0x08001a) on xmit-list, is in urib, is best urib route
  vpn: version 1196, (0x100002) on xmit-list
Multipath: eBGP iBGP

  Path type: internal, path is valid, not best reason: Router Id, multipath
             Imported from 10.0.168.95:31:172.16.11.0/24
  AS-Path: 99 , path sourced external to AS
    10.0.168.95 (metric 3) from 10.0.168.92 (10.0.168.92)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 0
      Extcommunity:
          RT:101:2588672
          VNID:2588672
      Originator: 10.0.168.95 Cluster list: 10.0.168.92

  Advertised path-id 1
  Path type: internal, path is valid, is best path
             Imported from 10.0.168.93:21:172.16.11.0/24
  AS-Path: 99 , path sourced external to AS
    10.0.168.93 (metric 3) from 10.0.168.92 (10.0.168.92)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 0
      Extcommunity:
          RT:101:2588672
          VNID:2588672
      Originator: 10.0.168.93 Cluster list: 10.0.168.9

  VRF advertise information:
  Path-id 1 not advertised to any peer
```

Path imported with RD of leaf 3

Imported from leaf 1 RD

We again have two path in RD 10.0.168.91:11 for the same prefix so we run best path again See the **Multipath** keyword. ACI does iBGP multipath by default so we will install both path in Rib

Imported from leaf 2 RD

```
pod2-leaf3# show ip route vrf LB:LB 172.16.11.0

172.16.11.0/24, ubest/mbest: 2/0
    *via 10.0.168.93%overlay-1, [200/0], 16:47:30, bgp-101, internal, tag 99 (mpls-vpn)
    *via 10.0.168.95%overlay-1, [200/0], 00:42:13, bgp-101, internal, tag 99 (mpls-vpn)
```

RD and Route-Target in ACI

- RD format :
  - Leaf-PTEP:id – id is a number identifying a vrf (before 4.2)
  - Node-ID:vrf_vnid – after 4.2
- Route-Target format
  - BGP_ASN:vrf_vnid

Route control in ACI detail

Agenda

- Route-control in ACI intro
- Route-map logic in ACI for BGP and OSPF/EIGRP
- Combinable vs non combinable route-map
- Where to apply route-map and what effect does it have ?
- New in 4.2 : BGP per per peer route-map
- Case study : ingress and egress traffic steering for BGP and OSPF

ACI route-Control intro

Reminder on route-map

- Route-map has the following structure :

```
route-map <name-of-RM>
   sequence <nr-1> [permit | deny]
     Match YYYY
     Set ZZZ
   sequence <nr-2> [permit | deny]
     Match YYYY
     Set ZZZ
   …
```

Processing route-map:

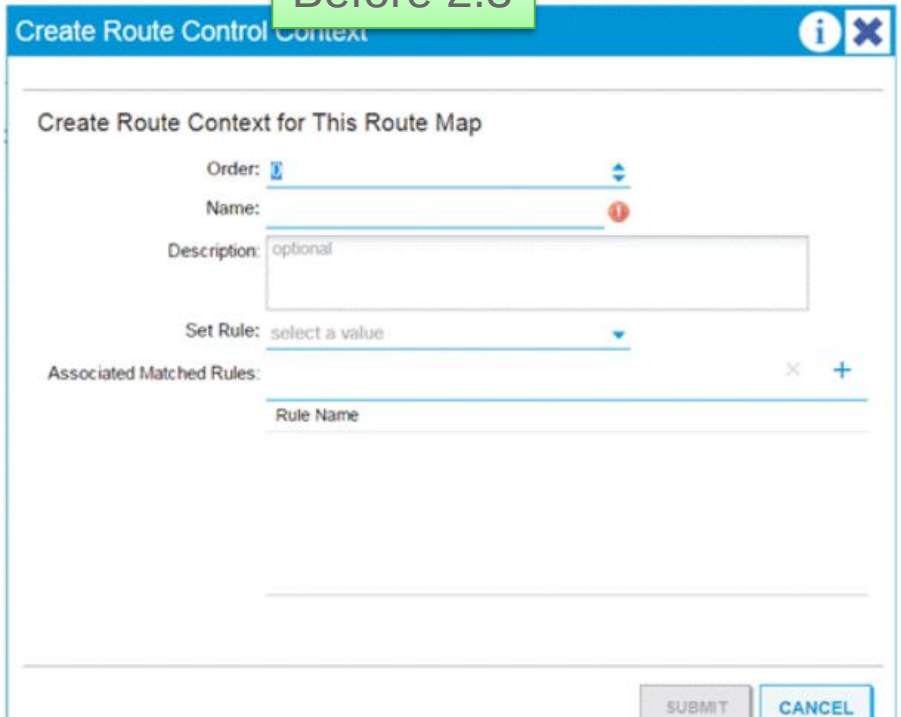
- We read sequence sequentially
- For each sequence, we evaluate against match criteria
  - If no match, go to next sequence
  - If match :
    - If seq is deny – do not allow the route (set action useless in deny sequence)
    - If seq is permit – apply set action if any (or allow if no set action)

    in both case STOP PROCESSING route-map
  - If reaching last seq and no match in any sequence implicit deny (do not allow)

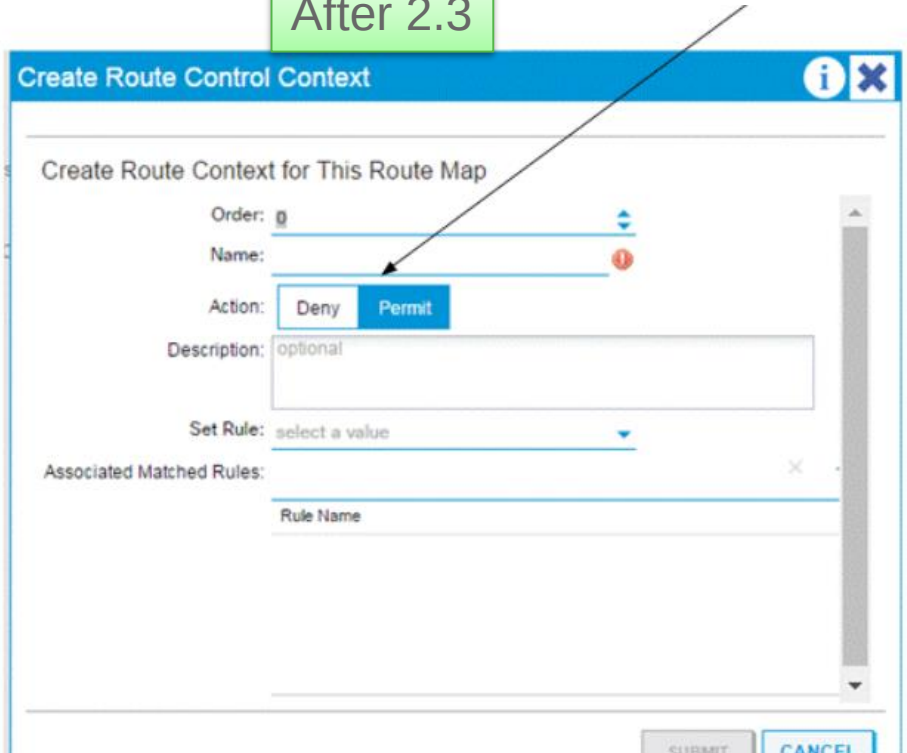
Deny sequence were introduced only in 2.3

ACI UI called : route-map sequence as route control context

Before 2.3



After 2.3



Route-map in ACI

- Every route-control in ACI is done through route-map
- Route-map are in APIC and send to leaf through the object model
- However Route-map or route-map sequence may be :
  - By default in code, such as loop prevention sequence
  - Implicit by various config (BD subnet advertised, External EPG prefix, …)
  - Manually user configured route-map applied somewhere (BD, ext EPG, redistribution,…)
  - Manually user configured route-map applied by default (default-export, default-import)
  - Implicit and manual configuration may be combined or not

Tips – route-map name

- Route map name usually indicated what they are and where they apply. If user configured, they will contain route-map object name :
  - exp-ctx-st-3112960 → static/direct route-map in VRF vnid 3112960
  - exp-l3out-BGP1-peer-2523139 → route-map in L3 out BGP1 in VRF 2523139 vnid
  - exp-ctx-proto-3112960 → redist of protocol (Ospf, eigrp, bgp,..) in vrf 3112960
- Route-map name started by exp are explicit due to some configuration in tenant
- Route-map name starting by imp are implicit build by ACI (example isis-ospf redistribution in multipod)

Tips – prefix-list name

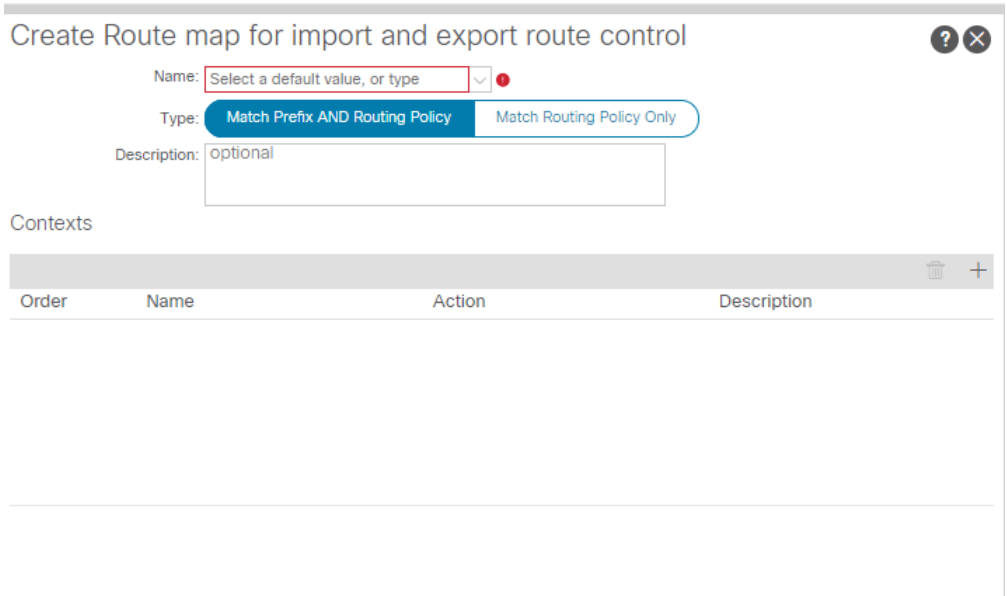
- Prefix-list name also indicate what they are:
  - external or internal prefix
  - Applied to a pcTag
  - Vrf vnid
  - May contain object name
- Example :
  - IPv4-st32772-3112960-exc-ext-inferred-export-dst → applied to pcTag 32772 in vrf 3112960 and for external (transit prefix) in export direction
  - → `apic1# moquery -c l3extInstP -f 'l3.extInstP.scope =="3112960"' | egrep -B 20 "pcTag.*32772" | egrep dn`

```
dn                  : uni/tn-RD-MC/out-MC/instP-mc-epg
```

Route Map
Combinable Vs non combinable

Route-map creation

When you create a route-map the First thing to choose is the type :
Match Prefix AND routing Policy
or
Match Routing Policy only



All route-map used so far in this presentation always use the default options Match Prefix and Routing Policy

Different between Match Prefix and Routing <> Match routing only

- Match Prefix and Routing Policy: Pervasive subnets (fvSubnet) and external subnets (l3extSubnet) are combined with a route profile and merged into a single route map (or route map entry). Match Prefix and Routing Policy is the default value.
- In combinable mode – Match sequence in usual route-map are replaced by subnet underneath the scope where the route-map is applied

- Match Routing Policy Only: The route profile is the only source of information to generate a route map, and it will overwrite other policy attributes.
- In Non combinable, YOU MUST specify exact match of what you want to match in the route-map definition, this overwrite potentially BD settings or l3 out subnet setting
- Closer to IOS/NXOS

Route Control

Where can we apply route control (tested in 4.2)

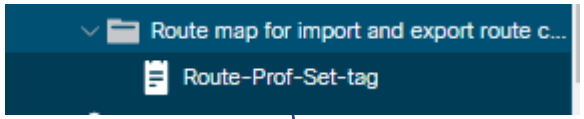
Route-map creation

- First you need to create a route-map under the L3 out you want to influence (here we do it in bgp l3 out)

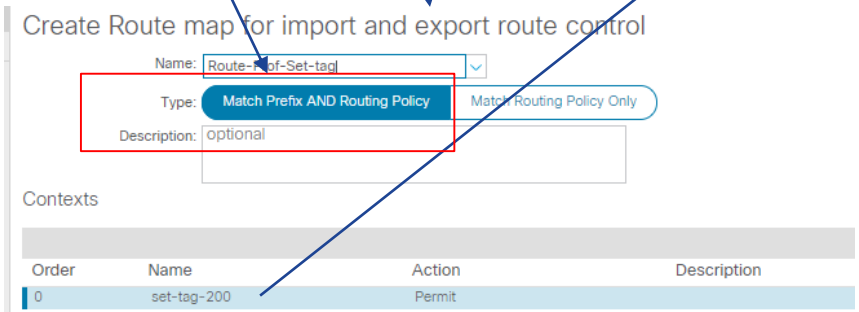
- Most example in this section are done on a BGP L3 out
- However most of it would apply to ospf/eigrp, except that two routes map are used one for static/direct and one for protocol

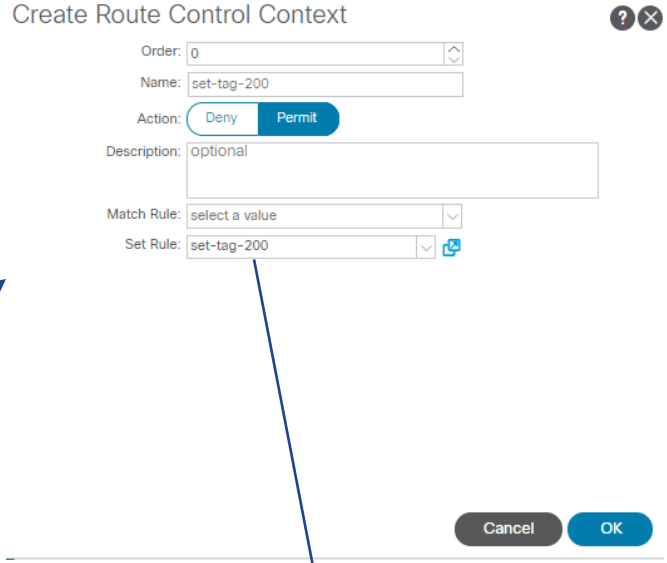
Create Route-map

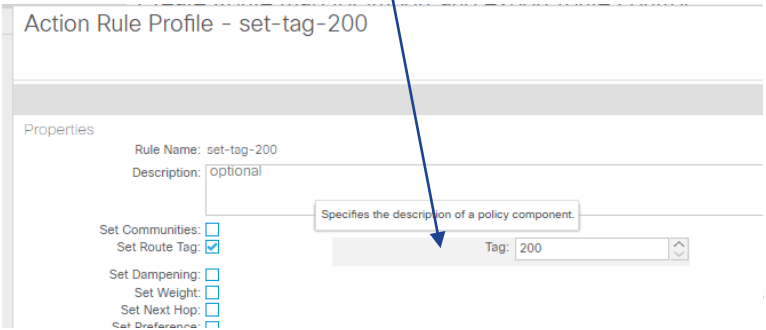
Create a route-map under the l3 out you want To change



Route-map set as combinable (more on that later)



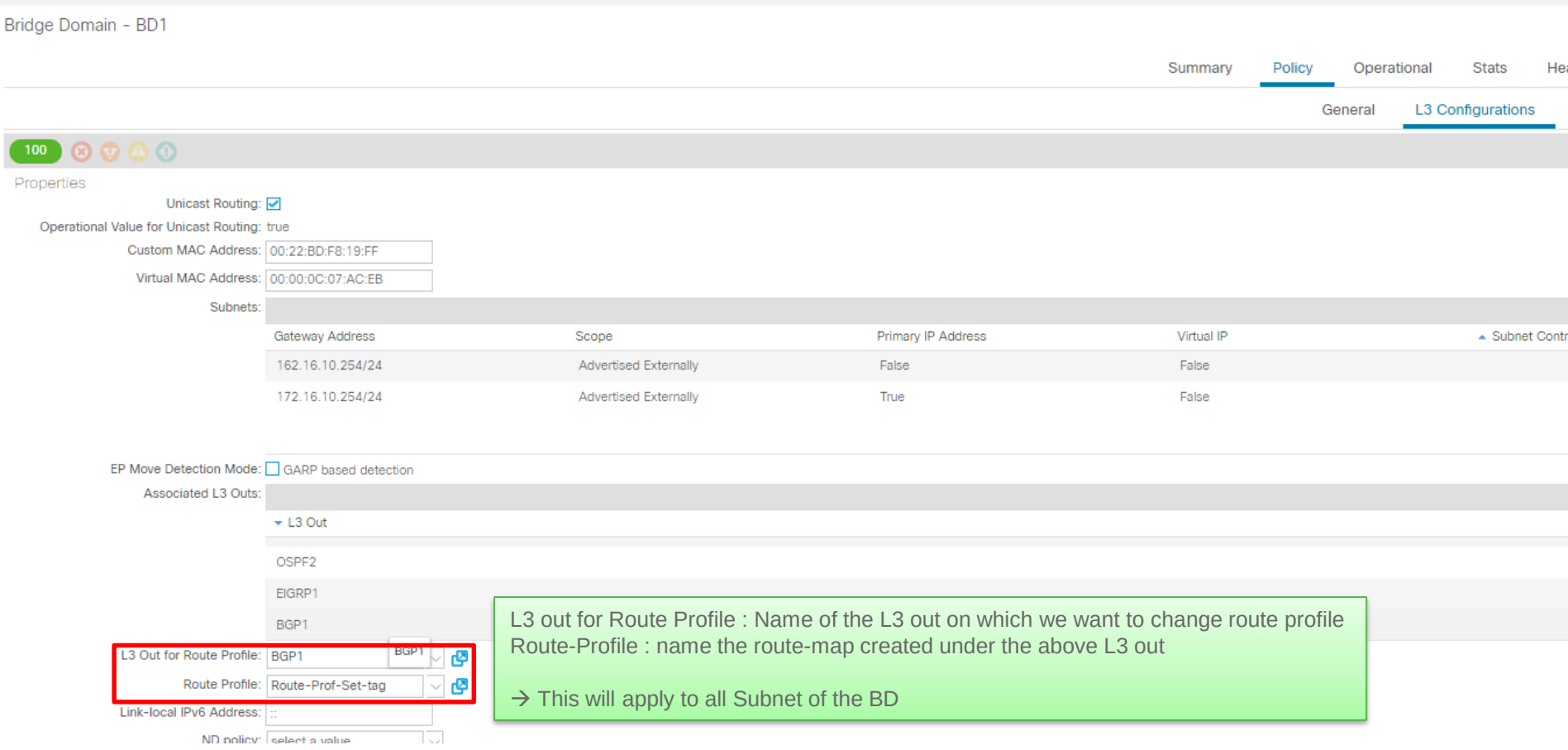




1. Route control at the BD level

- A route-profile at the Bridge Domain level is typically used to apply a policy to all subnets defined under a specific BD.
  - To configure this go to 'L3 Configurations' under the Bridge Domain,
  - select the L3out that will apply the policy when advertising the Subnet,
  - and then select the route-profile that is configured under that l3out.

Apply route-map to BD



Resulting route-map

```
bdsol-aci32-leaf1# show route-map  exp-l3out-BGP1-peer-2654211
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 8201
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-int-out-Route-Prof-Set-tag2set-tag-2000pfx-only-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 200
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15801
  Match clauses:
    tag: 4294967292
  Set clauses:
    tag 0
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15802
  Match clauses:
    tag: 4294967291
  Set clauses:
    tag 4294967295
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15803
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-ext-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15804
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 0
```

New sequence added before all others matching all subnet under BD with set tag 200

```
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer16387-2654211-exc-int-out-Route-Prof-Set-tag2set-tag-2000pfx-only-dst
ip prefix-list IPv4-peer16387-2654211-exc-int-out-Route-Prof-Set-tag2set-tag-2000pfx-only-dst: 2 entries
   seq 1 permit 172.16.10.254/24
   seq 2 permit 162.16.10.254/24
```

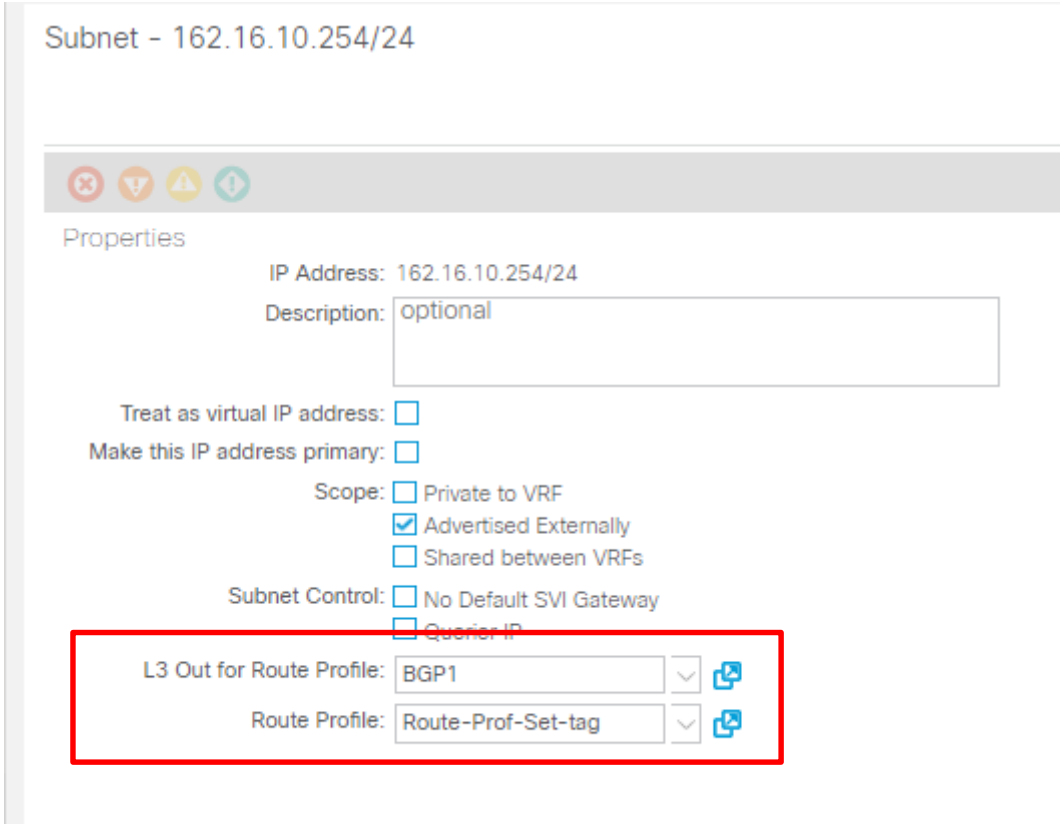
Existing sequence (without the route control under BD) – never used here as seq 8201 has same prefix list and is before

```
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer16387-2654211-exc-int-inferred-export-dst
ip prefix-list IPv4-peer16387-2654211-exc-int-inferred-export-dst: 2 entries
   seq 1 permit 172.16.10.254/24
   seq 2 permit 162.16.10.254/24
```

2. Route control at the BD subnet

- The route-profile can directly be associated to the BD Subnet.
- One of the only use-cases for doing this would be when there is more than one subnet configured under the BD and policy should be applied to these as they are advertised out more than one l3out. (currently only one l3out for route-profile can be associated at BD level)

Appy route-map to Subnet



Resulting route-map

```
bdsol-aci32-leaf1# show route-map  exp-l3out-BGP1-peer-2654211
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 8201
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-int-out-Route-Prof-Set-tag2set-tag-2000pfx-only-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 200
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15801
  Match clauses:
    tag: 4294967292
  Set clauses:
    tag 0
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15802
  Match clauses:
    tag: 4294967291
  Set clauses:
    tag 4294967295
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15803
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-ext-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15804
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 0
```

New sequence added before all others matching only one of the subet of the BD with set tag 200

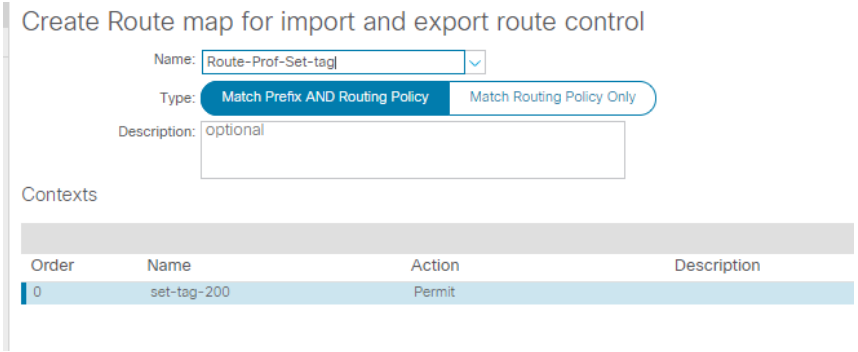
```
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer16387-2654211-exc-int-out-Route-Prof-Set-tag1set-tag-2000pfx-only-dst
ip prefix-list IPv4-peer16387-2654211-exc-int-out-Route-Prof-Set-tag1set-tag-2000pfx-only-dst: 1 entries
   seq 1 permit 162.16.10.254/24
```

Existing sequence (without the route control under BD) – HERE it is used as 172.16.10.0/24 is NOT Covered by first sequence

```
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer16387-2654211-exc-int-inferred-export-dst
ip prefix-list IPv4-peer16387-2654211-exc-int-inferred-export-dst: 2 entries
   seq 1 permit 172.16.10.254/24
   seq 2 permit 162.16.10.254/24
```

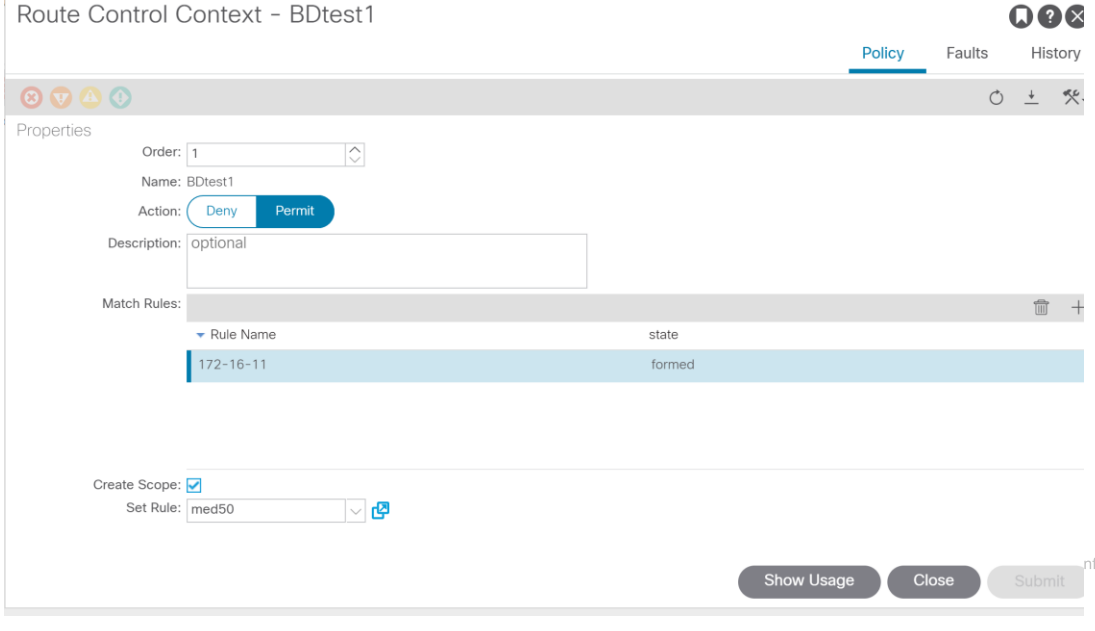
Note Combinable Route-map

- Note if the route map MUST BE set as combinable (Match Prefix and Routing policy) for the previous example
- If the route-map is set a match routing policy-only, we will not merge it with route-map based on prefix and the config will not be applied anywhere (no effect, no tagging to 200 of any routes)



1b – route-map NON Combinable at BD level

- Here we add a match statement in a non combinable route-map



1b- resulting route-map

```
bdsol-aci32-leaf1# show route-map  exp-l3out-BGP1-peer-2654211
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 8601
  Match clauses:
    ip address prefix-lists: IPv4-peer10950-2654211-exc-int-out-Export-BD2BDtest11172-16-11-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
    metric 50
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15801
  Match clauses:
    tag: 4294967292
  Set clauses:
    tag 0
route-map exp-l3out-BGP1-peer-26
  Match clauses:
    tag: 4294967291
  Set clauses:
    tag 4294967295
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15803
  Match clauses:
    ip address prefix-lists: IPv4-peer10950-2654211-exc-ext-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15804
  Match clauses:
    ip address prefix-lists: IPv4-peer10950-2654211-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 0
route-map exp-l3out-BGP1-peer-2654211, deny, sequenc
  Match clauses:
    route-type: direct
  Set clauses:
```

Prefix-list coming from
Non combinable RM with match

```
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer10950-2654211-exc-int-out-Export-BD2BDtest11172-16-11-dst
ip prefix-list IPv4-peer10950-2654211-exc-int-out-Export-BD2BDtest11172-16-11-dst: 1 entries
   seq 1 permit 172.16.11.0/24 le 32
```

Prefix-list coming from BD subnet (existing
Even without route-map)

```
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer10950-2654211-exc-int-inferred-export-
dst
ip prefix-list IPv4-peer10950-2654211-exc-int-inferred-export-dst: 3 entries
   seq 1 permit 172.25.1.254/24
   seq 2 permit 172.16.10.254/24
   seq 3 permit 162.16.10.254/24
```

3. Route control at the external l3 out epg level

- A Route-Profile can also be applied directly to an external epg level.
- This is intended for applying policy to transit prefixes but can also be used to apply policy to internal prefixes.
- The only caveat being that the internal prefixes (if matched) will receive the default vrf tag. If those subnets are supposed to be advertised back into ACI in a different VRF then make sure to change the default tag for that vrf so that the prefixes are accepted and installed in the routing-table.

External EPG Instance Profile - bgp1

100

Properties

QoS Class: Unspecified

Target DSCP: Unspecified

Configuration Status: applied

Configuration Issues:

Preferred Group Member: Exclude | Include

Subnets:

IP Address	Scope	Name	Aggregate
0.0.0.0/0	Export Route Control Subnet External Subnets for the External EPG		Aggregate Export
10.52.0.0/24	Export Route Control Subnet		
10.53.0.0/24	Export Route Control Subnet		

Page 1 Of 1 Objects Per Page: 15

L3Out Contract Masters:

L3Out Contract Master

No items have been found.
Select Actions to create a new item.

Route Control Profile:

Name	Direction
Route-Prof-Set-tag	Route Export Policy

Resulting route-map (here route-map set action sets med to 50)

```
bdsol-aci32-leaf1# show route-map  exp-l3out-BGP1-peer-2654211
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 8201
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-ext-out-Route-Prof-Set-tag2set-tag-2000pfx-only-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
    metric 50
    community none
    extcommunity
..
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15804
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 0
route-map exp-l3out-BGP1-peer-2654211, deny, sequence 16000
  Match clauses:
    route-type: direct
  Set clauses:
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 16201
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-agg-ext-out-Route-Prof-Set-tag2set-tag-2000pfx-only-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
    metric 50
    community none
    extcommunity
```

New sequence added before all others matching only all specific subnet of that l3 out epg with metric 50 (tag is always defulat loop tag)

```
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer16387-2654211-exc-ext-out-Route-Prof-Set-tag2set-tag-2000pfx-only-dst
ip prefix-list IPv4-peer16387-2654211-exc-ext-out-Route-Prof-Set-tag2set-tag-2000pfx-only-dst: 2 entries
   seq 1 permit 10.52.0.0/24
   seq 2 permit 10.53.0.0/24
```

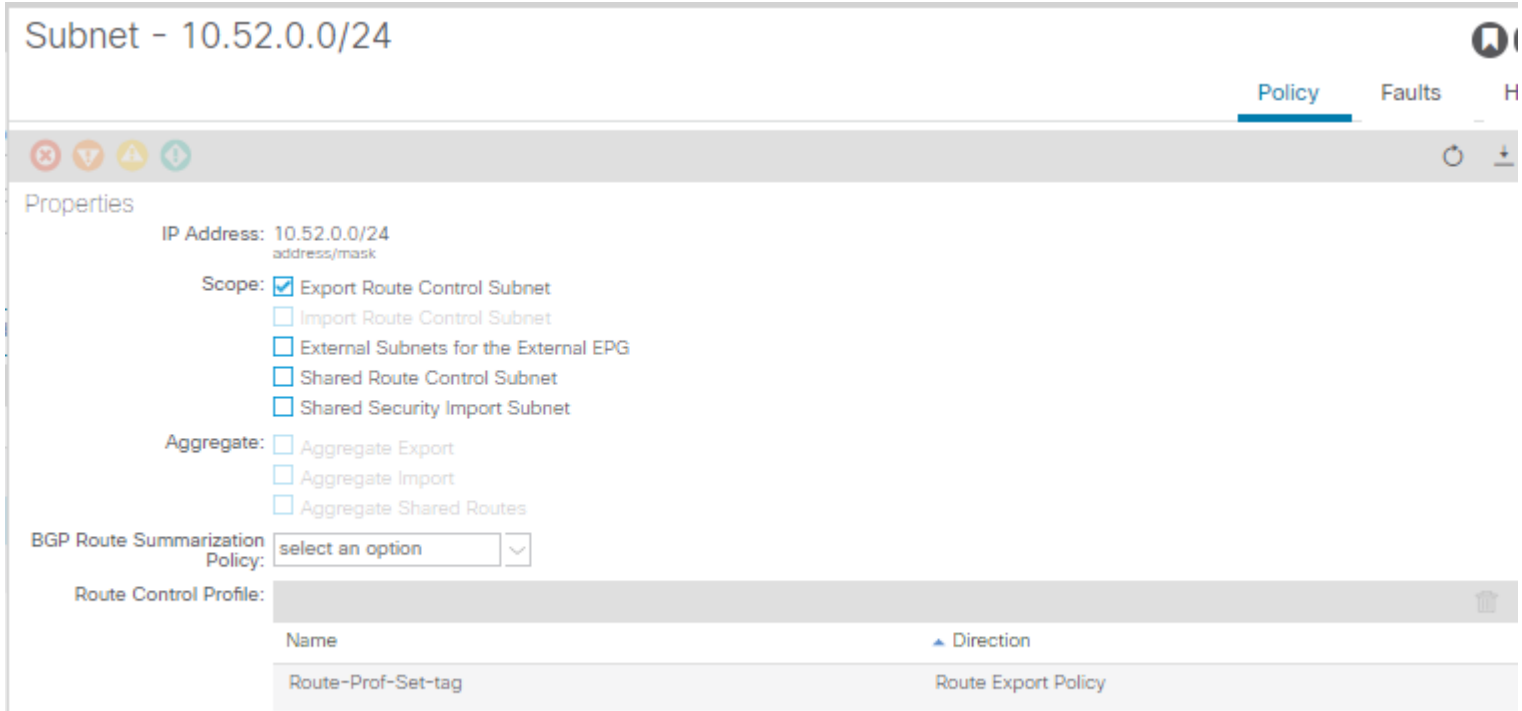
Aggregate default also set to med 50 but in last sequence (as 0.0.0.0/0 is in same epg with export route-control

```
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer16387-2654211-agg-ext-out-Route-Prof-Set-tag2set-tag-2000pfx-only-dst
ip prefix-list IPv4-peer16387-2654211-agg-ext-out-Route-Prof-Set-tag2set-tag-2000pfx-only-dst: 1 entries
   seq 1 permit 0.0.0.0/0 le 32
```

4. Route control at the external l3 out subnet level

- A Route-Profile can also be applied directly to an external epg subnet level.
- Same as scenario 3, but only for some subnet of the l3 out epg

GUI – only 10.52.0.0/24 is using route-map



Resulting route-map (here route-map set action sets med to 50)

```
bdsol-aci32-leaf1# show route-map  exp-l3out-BGP1-peer-2654211
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 8201
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-ext-out-Route-Prof-Set-tag2set-tag-2000pfx-only-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
    metric 50
    community none
    extcommunity
..
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15803
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-ext-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295

route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15804
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 0
route-map exp-l3out-BGP1-peer-2654211, deny, sequence 16000
  Match clauses:
    route-type: direct
  Set clauses:
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 19801
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-agg-ext-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
```

Same as in previous example, but here prefix-list only contains 10.52.0.0/24 not 10.53.0.0/24

```
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer16387-2654211-exc-ext-out-Route-Prof-Set-tag2set-tag-2000pfx-only-dst
ip prefix-list IPv4-peer16387-2654211-exc-ext-out-Route-Prof-Set-tag2set-tag-2000pfx-only-dst: 2 entries
   seq 1 permit 10.52.0.0/24
```

Sequence for specific prefix such as 10.53.0.0/24 NOT bound to the route-map

```
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer16387-2654211-exc-ext-inferred-export-dst
ip prefix-list IPv4-peer16387-2654211-exc-ext-inferred-export-dst: 1 entries
   seq 2 permit 10.53.0.0/24
```

Aggregate default is NOT set to med 50 here.

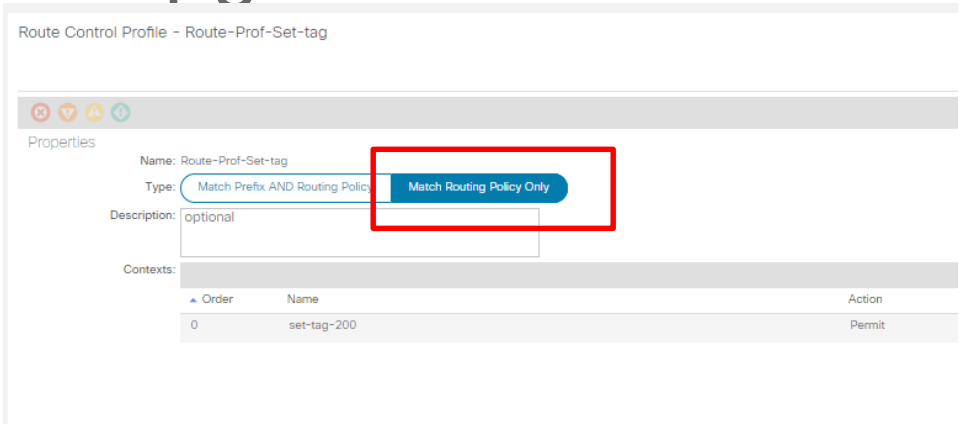
```
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer16387-2654211-agg-ext-inferred-export-dst
ip prefix-list IPv4-peer16387-2654211-agg-ext-inferred-export-dst: 1 entries
   seq 1 permit 0.0.0.0/0 le 32
```

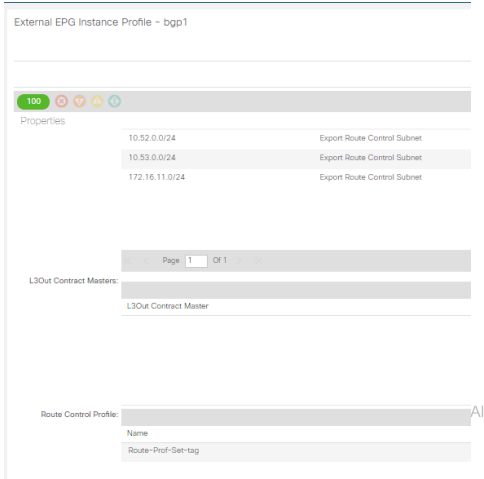

Example 3b or 4b - Non-Combinable Route-map applied to l3 epg or l3 epg subnet

- If we change the route map to not combinable,
  - Whether it is applied at l3 epg or l3 subnet do not matter as we do not combine with prefix, we only match the match rule.
    in the route-map used here there was no match rule only a set as the match was implicitely done by the combinable mode matching on prefix.

    So my previous route-map as non combinable is useless ☺

Note : non combinable route-map applied to Ext Epg or ext EPG subnet seems to completely delete the default prefix-list coming from export/import route control subnet





Resulting route-map (here route-map set action sets med to 50)

```
bdsol-aci32-leaf1# show route-map  exp-l3out-BGP1-peer-2654211
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15801
  Match clauses:
    tag: 4294967292
  Set clauses:
    tag 0
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15802
  Match clauses:
    tag: 4294967291
  Set clauses:
    tag 4294967295
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15804
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 0
route-map exp-l3out-BGP1-peer-2654211, deny, sequence 16000
  Match clauses:
    route-type: direct
  Set clauses:
```

Not set MED anywhere . Export route-control flag is actually overcome
But the route-map applied to l3 out epg

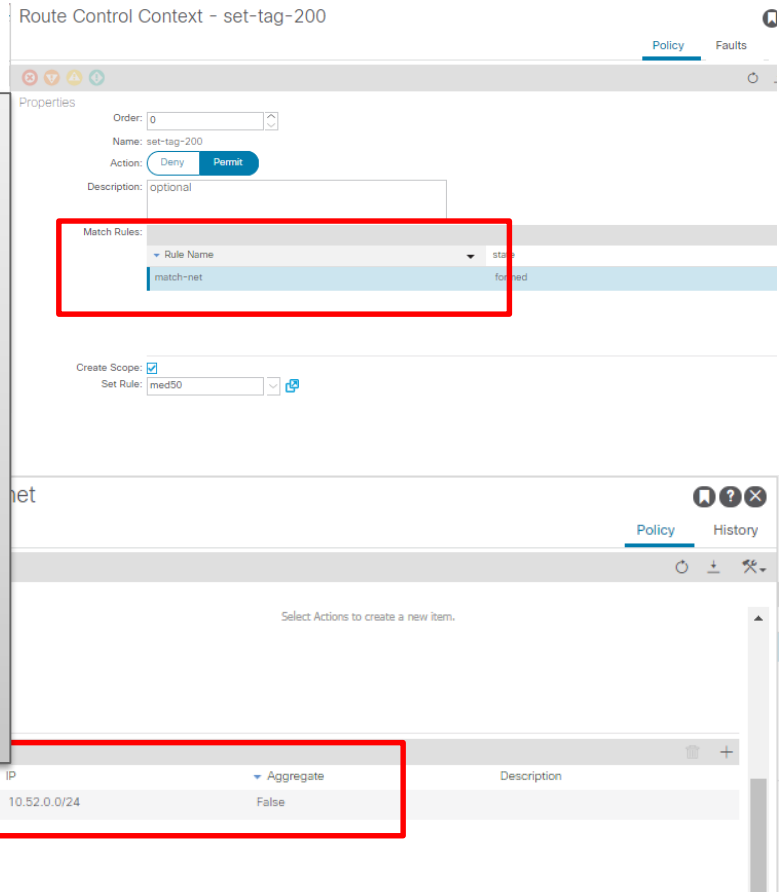
Only prefix is for internal prefix (int-inferred-export

Example 3c-4c : let add specific match rule in the route-map

```
bdsol-aci32-leaf1# show route-map  exp-l3out-BGP1-peer-2654211
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 8201
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-ext-out-Route-Prof-Set-
tag2set-tag-2000match-net-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
    metric 50
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15801
  Match clauses:
    tag: 4294967292
  Set clauses:
    tag 0
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15802
  Match clauses:
    tag: 4294967291
  Set clauses:
    tag 4294967295
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15804
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 0
```

Prefix list added by the match rule

```
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer16387-2654211-exc-ext-out-Route-
Prof-Set-tag2set-tag-2000match-net-dst
ip prefix-list IPv4-peer16387-2654211-exc-ext-out-Route-Prof-Set-tag2set-tag-
2000match-net-dst: 1 entries
   seq 1 permit 10.52.0.0/24
```



Note on 3c-4c

- If you use route-map combinable and you have both
  - Subnet and/or l3 epg with route profile
  - And MATCH rule in route-map
- We will combine in prefix list both prefix with export route-control and prefix matches in route-map

```
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer16387-2654211-exc-ext-out-Route-Prof-Set-tag2set-tag-
2000match-net-dst
ip prefix-list IPv4-peer16387-2654211-exc-ext-out-Route-Prof-Set-tag2set-tag-2000match-net-dst: 3 entries
   seq 1 permit 10.52.0.0/24      -- match in route-map
   seq 2 permit 10.53.0.0/24      -- export route control and route profile attached to epg or subnet
```

Recommendation

- For transit routing, it is recommended to take one of those approach :
  - Route-map combinable and NO MATCH statement – match done per export route control flag only

  Or

  - Route-map not combinable and ALL MATCH statement done in route-map (no subnet with exp route-control flag). This is more flexible and allow with multiple route-map sequence to
    - Match exactly what we want in each sequence with different set exact
    - To mix deny and permit sequence to prevent some subnet out
      - Example :

        Seq 1 : deny 172.16.10.0/24 and 172.16.11.0/24

        Seq 2 : permit 172.16.0.0/16 and set MED 50

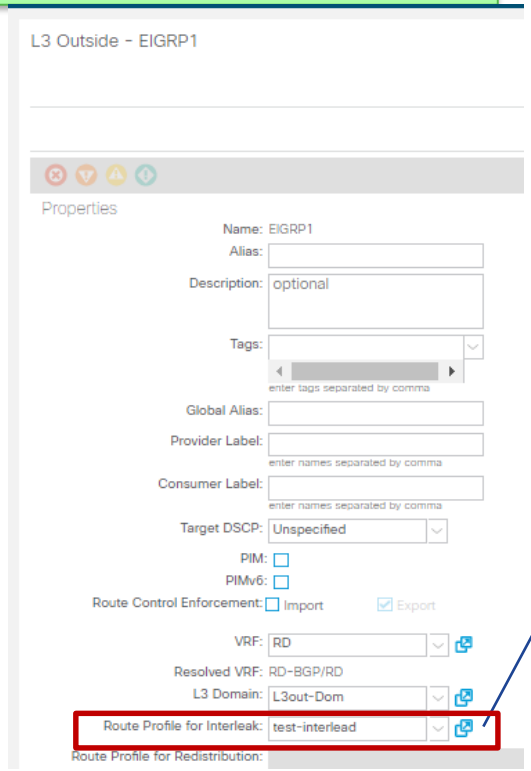
        Seq 3 : permit 0.0.0.0/0 no set action

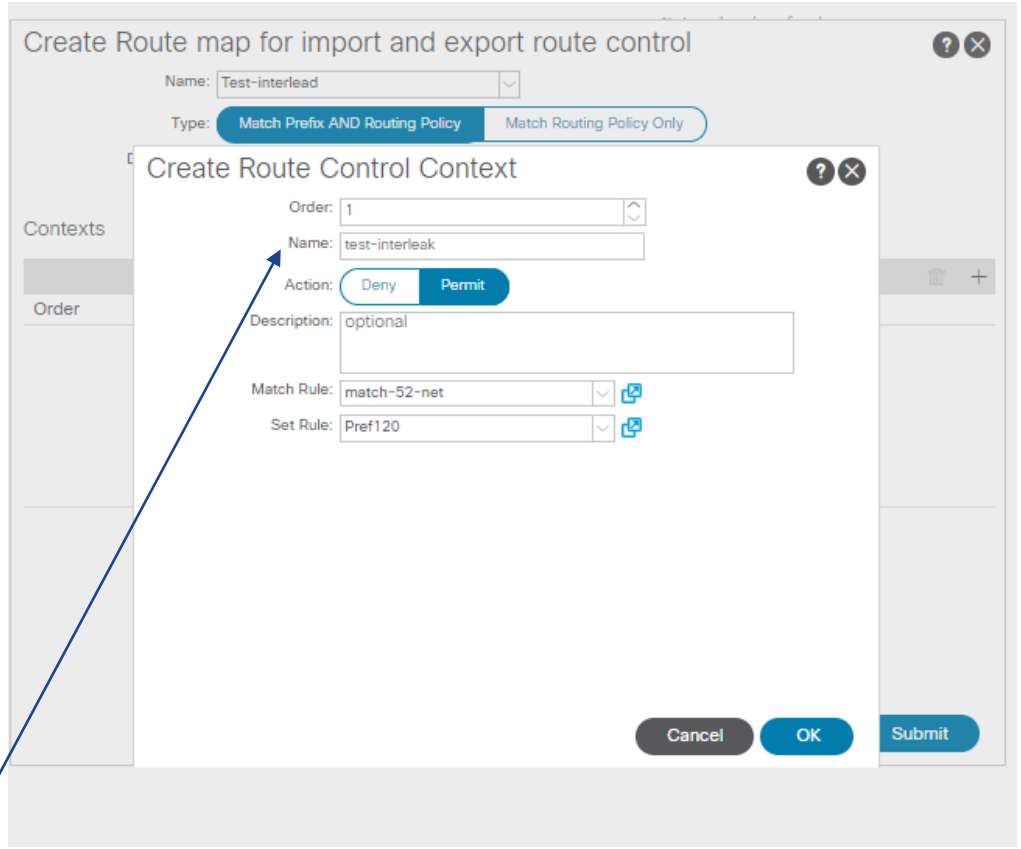
5. Route control at the L3 out level as interleak policy

- The "Route Profile for Interleak" is intended specifically to set policy when redistributing prefixes from some external protocol into BGP.
- The route-profile is then applied on the source External protocol (non-bgp) as a "Route Profile for Interleak" policy.
- This is useful for setting BGP attributes when a prefix is redistributed into the internal fabric bgp process or it can also be used to set bgp attributes when advertising transit prefixes from a non-bgp l3out to a bgp l3out.
- So this is set for incoming route coming in one L3 out (BGP , EIGRP, OSPF,..) to set attribute when they are injected in iBGP

Example

We create a new route-map in an interleak of eigrp l3 out matching network 10.52.0.0/24 and settting local pref to 120





Resulting route-map

```
bdsol-aci32-leaf1# show bgp process vrf RD-BGP:RD
..
 Redistribution
        direct, route-map permit-all
        static, route-map imp-ctx-bgp-st-interleak-2654211
        coop, route-map exp-ctx-st-2654211
        eigrp, route-map imp-ctx-proto-interleak-2654211

bdsol-aci32-leaf1# show route-map imp-ctx-proto-interleak-2654211
route-map imp-ctx-proto-interleak-2654211, permit, sequence 201
  Match clauses:
    ip address prefix-lists: IPv4-st2654211-ext-in-test-interlead0test0match-52-net-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    local-preference 120

bdsol-aci32-leaf1# show ip prefix-list IPv4-st2654211-ext-in-test-interlead0test0match-52-net-dst
ip prefix-list IPv4-st2654211-ext-in-test-interlead0test0match-52-net-dst: 1 entries
   seq 1 permit 10.52.0.0/24
```

This will mark the bgp path for 10.52.0.0 to local pref 120 in iBGP and Eventual other L3 out BGP peer

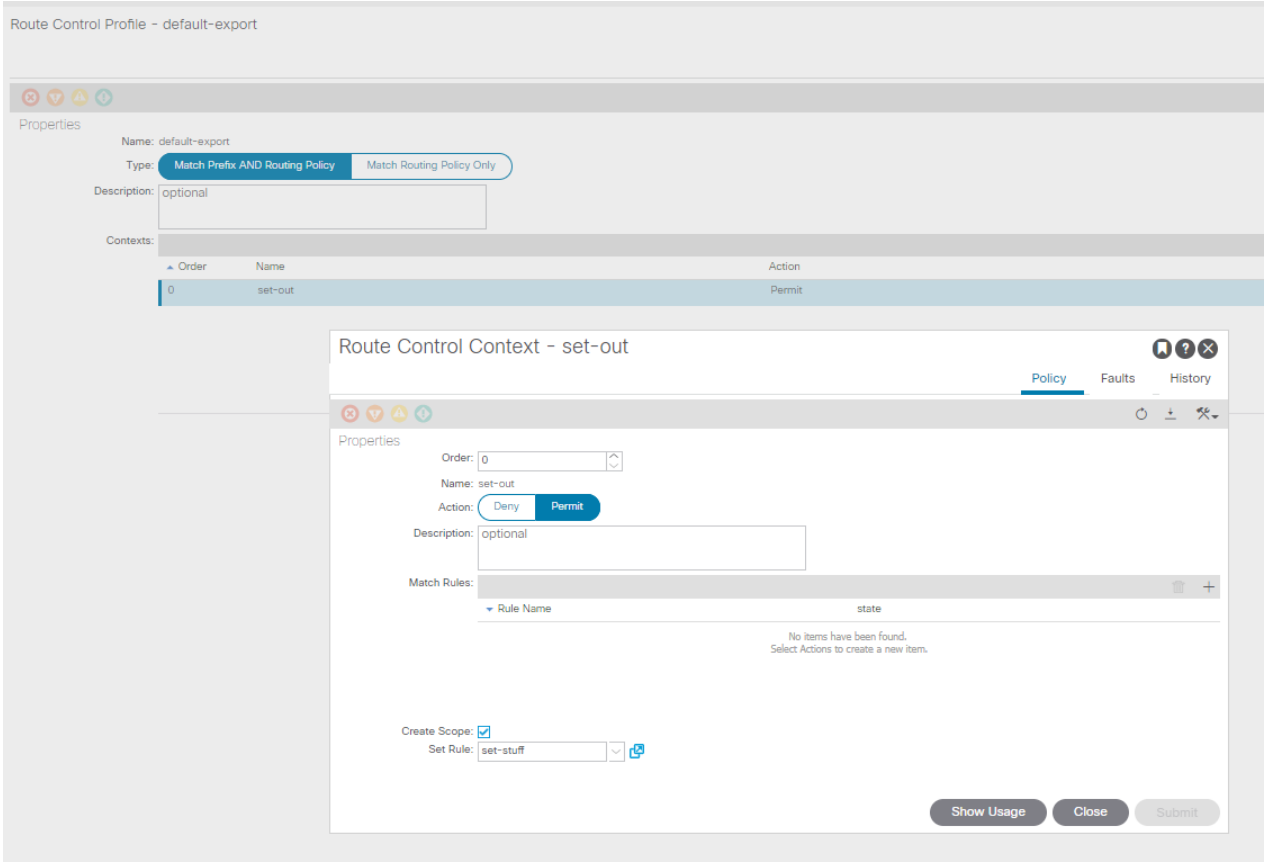
6. Route control as default export/import

- There are two different default route-profiles that can be configured at the l3out level.
  - 'default-import' and 'default-export' route-profiles.
- These do not have to be applied anywhere.
- As long as they exist they will affect matched routes that are being advertised out (or in) that l3out.
- The configuration is identical to any other route-profile creation except that the name must be specified as 'default-export' or 'default-import

Default export route-map combinable

We had a seq
In default-export
With no match rule
And set med 142

As it is combinable
It will apply to all prefix
(internal BD and transit)
No need to apply that
Route-map anywhere



Resulting route-map (here route-map set action sets med to 142)

```
bdsol-aci32-leaf1# show route-map  exp-l3out-BGP1-peer-2654211
 route-map exp-l3out-BGP1-peer-2654211, permit, sequence 8201
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-ext-out-default-export2se
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
    metric 142
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 8202
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-int-out-default-export2set-out0pfx-only-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 500
    metric 142
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15801
  Match clauses:
    tag: 4294967292
  Set clauses:
    tag 0
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15802
  Match clauses:
    tag: 4294967291
  Set clauses:
    tag 4294967295
route-map exp-l3out-BGP1-peer-2654211, deny, sequence 16000
  Match clauses:
    route-type: direct
  Set clauses:
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 16201
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-agg-ext-out-default-export4set-out0pfx-only-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
    metric 142
```

Seq for specific transit export route control subnet

```
bdsol-aci32-leaf1# show ip prefix-list  IPv4-peer16387-2654211-exc-ext-out-default-
export2set-out0pfx-only-dst
ip prefix-list IPv4-peer16387-2654211-exc-ext-out-default-export2set-out0pfx-only-ds
3 entries
   seq 1 permit 10.52.0.0/24
   seq 2 permit 10.53.0.0/24
   seq 3 permit 172.16.11.0/24
```

Sequence containing all BD subnet (internal)

```
show ip prefix-list IPv4-peer16387-2654211-exc-int-out-default-export2set-out0pfx-only-dst
ip prefix-list IPv4-peer16387-2654211-exc-int-out-default-export2set-out0pfx-only-dst: 2
entries
   seq 1 permit 172.16.10.254/24
   seq 2 permit 162.16.10.254/24
```

Seq for agg export

```
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer16387-2654211-agg-ext-out-
default-export4set-out0pfx-only-dst
ip prefix-list IPv4-peer16387-2654211-agg-ext-out-default-export4set-out0pfx-
only-dst: 1 entries
   seq 1 permit 0.0.0.0/0 le 32
```


Example 6b- We change the default-export route-map to non combinable

As, there was no match rule, we do not allow anything out. If we want to allow sth out, we need to create specific match rule (could be 0.0.0.0/0 aggregate)

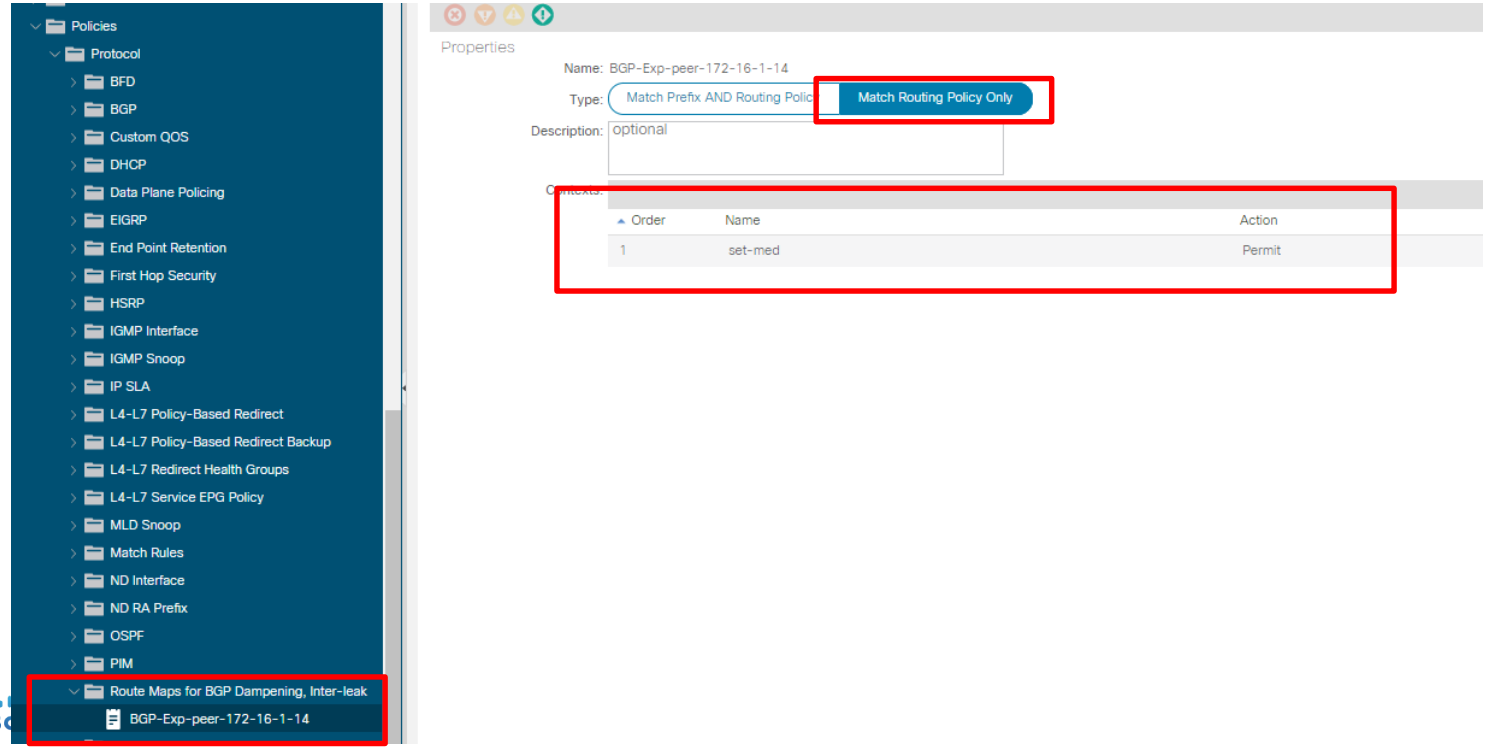
```
bdsol-aci32-leaf1# show route-map  exp-l3out-BGP1-peer-2654211
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15801
  Match clauses:
    tag: 4294967292
  Set clauses:
    tag 0
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15802
  Match clauses:
    tag: 4294967291
  Set clauses:
    tag 4294967295
route-map exp-l3out-BGP1-peer-2654211, deny, sequence 16000
  Match clauses:
    route-type: direct
  Set clauses:
```

Route Control Per BGP peer (new in 4.2)

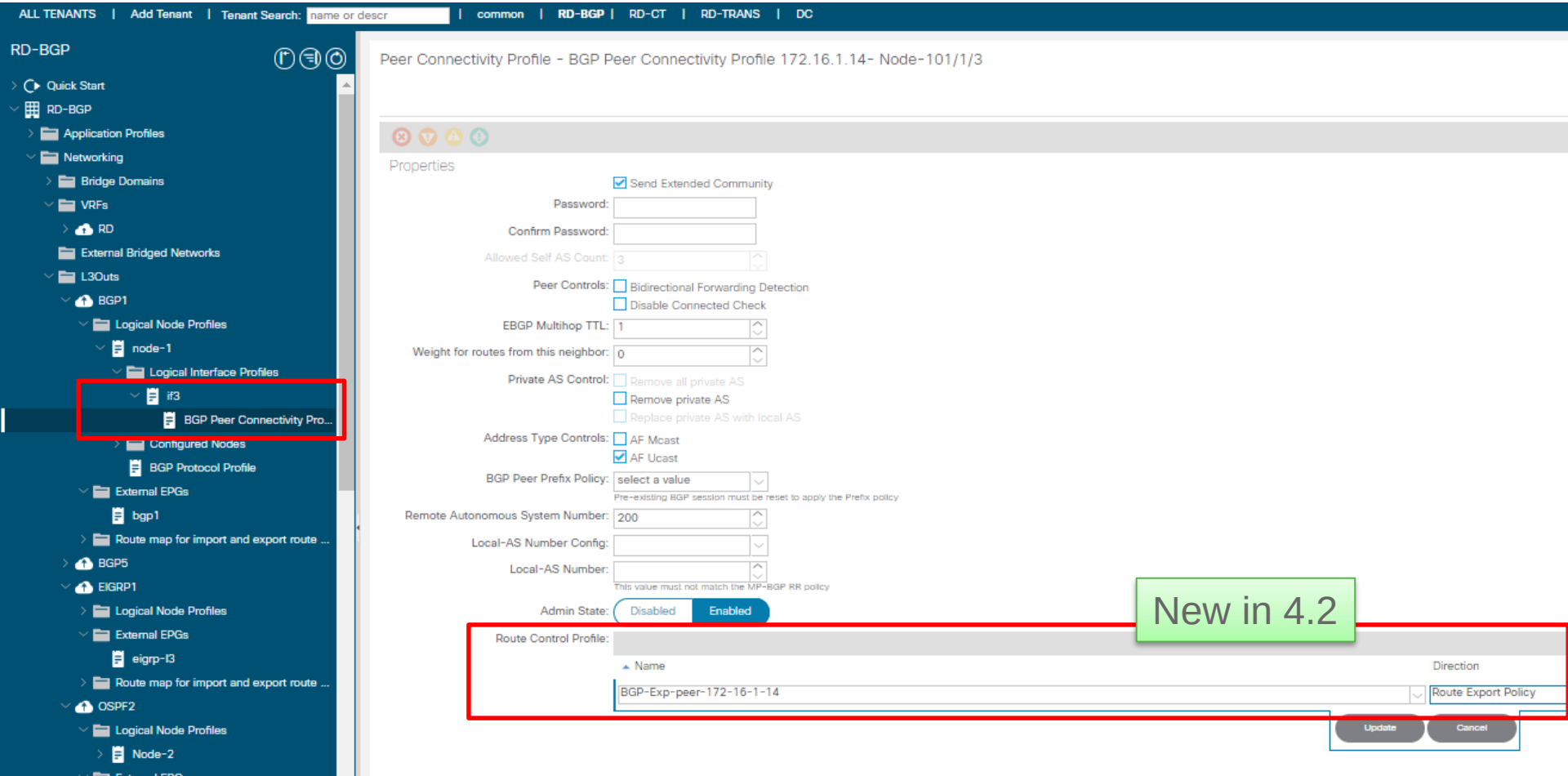
Overview

- For Cisco APIC releases before Release 4.2(1), you configure these policies at the L3Out level, under the L3Out profile (l3extInstP) or through the L3Out subnet under the L3Out (l3extSubnet), so those policies apply to protocols configured for all nodes or paths included in the L3Out. With this configuration, there could be multiple node profiles configured in the L3Out, and each could have multiple nodes or paths with the BGP neighbor specified. Because of this, there is no way to apply individual policies to each protocol entity.
- Beginning with Cisco APIC Release 4.2(1), the route control per BGP peer feature is introduced to begin to address this situation, where more granularity in route export and import control is needed.

1/ Create a route-map with match route policy only



2/ apply it to BGP peer



Before : outbound route-map is per L3 out (BGP1) and per vrf (2654211)

```
bdsol-aci32-leaf1# show ip bgp neighbors 172.16.1.14 vrf RD-BGP:RD| egrep route-map
  Inbound route-map configured is permit-all, handle obtained
  Outbound route-map configured is exp-l3out-BGP1-peer-2654211, handle obtained
```

After only the route-map we configure is there. If we have other neighbor you can use Different route-map

```
bdsol-aci32-leaf1# show ip bgp neighbors 172.16.1.14 vrf RD-BGP:RD| egrep route-map
  Inbound route-map configured is permit-all, handle obtained
  Outbound route-map configured is RD-BGP-BGP-Exp-peer-172-16-1-14-BGP1-out, handle obtained
```

Old route-map detail

- Old route-map (global for the l3 out) contains
  - BD where the L3 out is attached
  - Subnet with export route-control flag
  - …

```
bdsol-aci32-leaf1# show route-map exp-l3out-BGP1-peer-2654211
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15803
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-ext-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15804
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 0
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer16387-2654211-exc-ext-inferred-export-dst
ip prefix-list IPv4-peer16387-2654211-exc-ext-inferred-export-dst: 1 entries
   seq 1 permit 10.52.0.0/24
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer16387-2654211-exc-int-inferred-export-dst
ip prefix-list IPv4-peer16387-2654211-exc-int-inferred-export-dst: 1 entries
   seq 1 permit 172.16.10.254/24
bdsol-aci32-leaf1#
```

New route-map detail

- Note that the new per neighbor route-map takes precedence to previous global config, it only contains match rule you set in the route-map, it does not contain BD subnet where we bound this L3 out for example

```
bdsol-aci32-leaf1# show route-map RD-BGP-BGP-Exp-peer-172-16-1-14-BGP1-out
route-map RD-BGP-BGP-Exp-peer-172-16-1-14-BGP1-out, permit, sequence 601
  Match clauses:
    ip address prefix-lists: IPv4-RD-BGP-BGP-Exp-peer-172-16-1-14-BGP1-out-ext-0set-med1172-16-11-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
route-map RD-BGP-BGP-Exp-peer-172-16-1-14-BGP1-out, deny, sequence 16000
  Match clauses:
    route-type: direct
  Set clauses:
bdsol-aci32-leaf1# show ip prefix-list IPv4-RD-BGP-BGP-Exp-peer-172-16-1-14-BGP1-out-ext-0set-med1172-16-
11-dst
ip prefix-list IPv4-RD-BGP-BGP-Exp-peer-172-16-1-14-BGP1-out-ext-0set-med1172-16-11-dst: 1 entries
   seq 1 permit 172.16.11.0/24 le 32
```

BGP path selection

1. Prefer the path with the highest **WEIGHT**. (local to router – cisco specific)
2. Prefer the path with the highest **LOCAL_PREFERENCE**. (value of 100 by default if no Local pref set. Local to AS)
3. Prefer the path that was **locally** originated via a network or aggregate BGP subcommand or through redistribution from an IGP.
4. Prefer the path with the shortest **AS_PATH**.
5. Prefer the path with the lowest **origin type**. (IGP is lower than Exterior Gateway Protocol (EGP), and EGP is lower than INCOMPLETE)
6. Prefer the path with the lowest multi-exit discriminator – MED . (Paths received with no MED are assigned a MED of 0
7. Prefer **eBGP over iBGP** paths.
8. Prefer the path with the lowest **IGP metric** to the BGP next hop. Continue, even if bestpath is already selected.
9. Determine if multiple paths require installation in the routing table for **BGP Multipath.** Continue, if bestpath is not yet selected.
10. When both paths are external, prefer the path that was received first (**the oldest one**).
11. Prefer the route that comes from the BGP router with the lowest **router ID**.
12. Prefer the path that comes from the lowest neighbor address. (This address is the IP address that is used in the BGP **neighbor** configuration)

Criteria we would use in larger bgp network, ext router Would pick up one or the other depending on igp metric

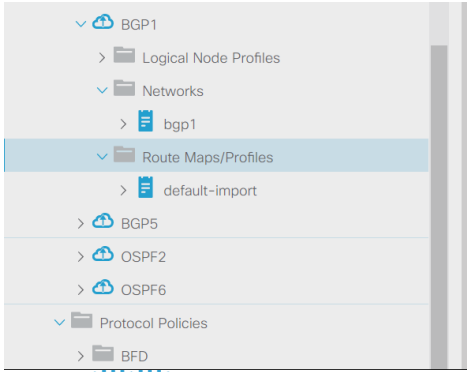
Criteria used here

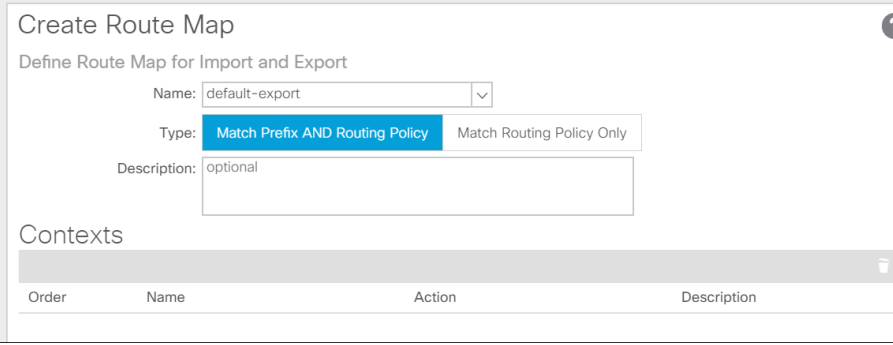
BGP influencing outbound path

- Here we can set MED out on prefix send from leaf 1. MED will be set to 50. The default being 0 we will prefer path from Leaf 5 (lower MED)
- This can be done in 2 way : using BD config (L3 out for Route Profile and route profile under L3 config of BD) . Currently broken in 3.0 due to CSCvg23213
- Or by applying the default route-export route-map under the L3 out for BGP1 (leaf 1 BGP) . We will do that here.
- Note : we could as well have made the as-path longer

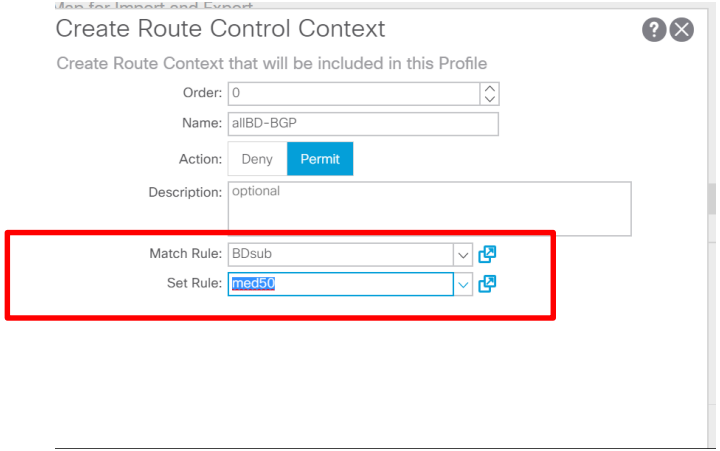
Step 1 – Configure default route-export on L3 out BGP 1

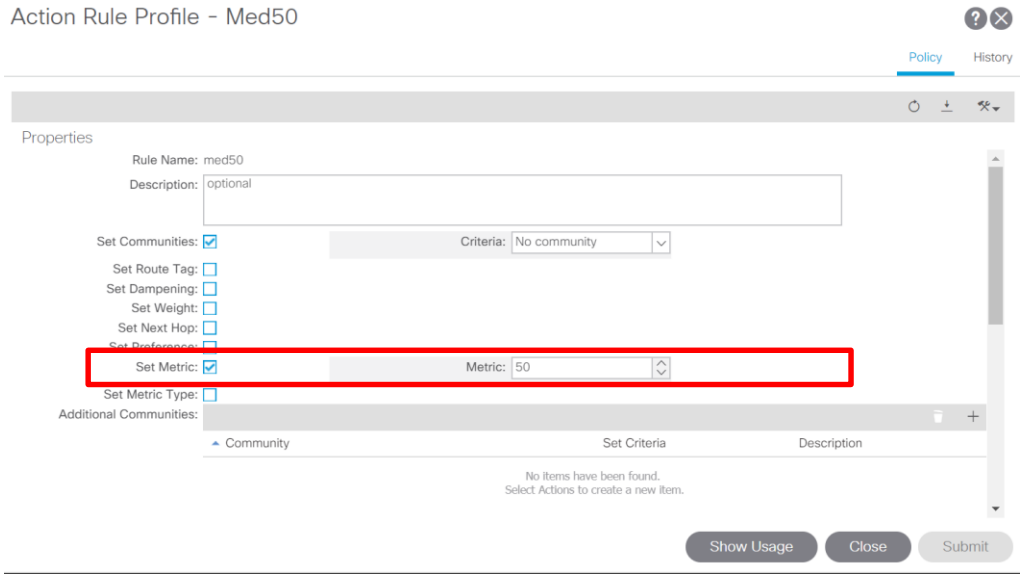
Right click on Route Maps/Profile under the L3 out we want to tune
And create new route-map, default-export will be in the drop down by default.
Then add a Context to it





Step 2 – Create the context to set MED to 50





Route-map on BL before and after

```
bdsol-aci32-leaf1# show ip bgp neighbors 172.16.1.14  vrf RD-BGP:RD | egrep route
  BGP version 4, remote router ID 172.16.1.25
  Inbound route-map configured is imp-l3out-BGP1-peer-2654211, handle obtained
  Outbound route-map configured is exp-l3out-BGP1-peer-2654211, handle obtained
```

```
bdsol-aci32-leaf1# show route-map exp-l3out-BGP1-peer-2654211
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 15801
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
route-map exp-l3out-BGP1-peer-2654211, deny, sequence 16000
  Match clauses:
    route-type: direct
  Set clauses:
```

Before

```
bdsol-aci32-leaf1# show route-map exp-l3out-BGP1-peer-2654211
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 8201
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-ext-out-default-export2allBD-BGP0BDsub-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    tag 4294967295
    metric 50
    community none
    extcommunity 4byteas-generic none
route-map exp-l3out-BGP1-peer-2654211, permit, sequence 8202
  Match clauses:
    ip address prefix-lists: IPv4-peer16387-2654211-exc-int-out-default-export2allBD-BGP0BDsub-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:
    metric 50
    community none
    extcommunity 4byteas-generic none
route-map exp-l3out-BGP1-peer-2654211, deny, sequence 16000
  Match clauses:
    route-type: direct
  Set clauses:
bdsol-aci32-leaf1# show ip prefix-list IPv4-peer16387-2654211-exc-ext-out-default-export2allBD-BGP0BDsub-dst
ip prefix-list IPv4-peer16387-2654211-exc-ext-out-default-export2allBD-BGP0BDsub-dst: 1 entries
   seq 1 permit 172.16.10.0/24 le 32
bdsol-aci32-leaf1#
```

After