Implement BGP EVPN Protected Overlay Segmentation on Catalyst 9000 Series Switches

Contents

Introduction

This document describes how to implement BGP EVPN VXLAN Protected Overlay Segmentation on Catalyst 9000 Series Switches.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- BGP EVPN VxLAN concepts
- BGP EVPN Unicast Troubleshooting
- BGP EVPN VxLAN routing policy

Components Used

The information in this document is based on these software and hardware versions:

- Catalyst 9300
- Catalyst 9400
- Catalyst 9500
- Catalyst 9600
- Cisco IOS® XE 17.12.1 and later versions

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

High Level Feauture Description

The protected segment feature is a security measure that prevents ports from forwarding traffic to each other, even if they are on the same VLAN and same switch

- This feature is similar to 'switchport protected' or private Vlans, but for EVPN fabrics.
- This design forces all traffic to the CGW where it can be inspected by a Firewall before being sent to its final destination.
- Traffic flows are controlled, deterministic, and easy to inspect using a centralized security appliance.

Document Details

This document is part 2 or 3 inter-related documents:

- **Document 1:** Implement BGP EVPN Routing Policy on Catalyst 9000 Series Switches covers how to control the BGP BUM traffic in the Overlay, and must be configured first
- **Document 2:** This document. Building upon the Overlay design and policy of document 1, this document describes the implementation of the 'protected' keyword
- **Document 3:**Implement BGP EVPN DHCP Layer 2 Relay on Catalyst 9000 Series Switches covers how DHCP relay works on an L2 only VTEP

Caution: You must implement the configuration in document 1 prior to implementing protected segment configurations.

Protected Segment Types

Totally isolated

- Allows only North to South communication, and
- The gateway is advertised into the fabric with the 'default-gateway advertise' CLI

Mostly Isolated

- Allows North to South communication (in this use case East / West traffic flows are allowed based on firewall traffic policies)
- Allows East to West communication (based on firewall traffic policies)
- The gateway is external to the fabric & the SVI is not advertised using the 'default-gateway advertise' CLI

Switch Behavior

- Hosts cannot communicate with each other directly even if they are connected to the same switch (ARP request not sent to other ports on same switch when hosts are in the same VRF/Vlan/Segment)
- No BUM traffic between L2 VTEPs (IMET prefixes filtered using the routing policy configuration)
- All packets from the hosts are relayed to Border Leaf to be forwarded. (This means for Host 1 to communicate to host 2 on same leaf, traffic is hair pinned up to the CGW)

Route Type 2 Handling

- Access Leafs advertise local RT2 with E-Tree Extended Community and Leaf flag set.
- Access Leafs do not install any remote RT2 received with E-Tree Extended Community and Leaf flag set in data plane.
- Access Leafs do not install each others RT2 in data plane.
- Access Leafs and Border Leaf (CGW) install each others RT2 in data plane.
- No configuration change required on Access Leaf or Border Leaf.

Design Summary

- For broadcast (BUM) the RT3 topology is hub and spoke in order to force broadcast traffic such as ARP up to the GCW.
- To account for host mobility the RT2 are full mesh at the BGP control plane (when a host moves from one VTEP to another the Seq number is incremented in the RT2)
- The data plane selectively installs MAC addresses.
    - A leaf installs only local MACs & RT2 which contain the DEF GW attribute
    - The CGW does not have the protected KW and installs all local MAC & remote RT2 in its data plane.

Terminology

VRF	Virtual Routing Forwarding	Defines a layer 3 routing domain that be separated from other VRF and global IPv4/IPv6 routing domain
AF	Address Family	Defines which type prefixes and routing info BGP handles
AS	Autonomous System	A set of Internet routable IP prefixes that belong to a network or a collection of networks that are all managed, controlled and supervised by a single entity or organization
EVPN	Ethernet Virtual Private Network	Extension that allows BGP to transport Layer 2 MAC and Layer 3 IP information is EVPN and uses Multi-Protocol Border Gateway Protocol (MP-BGP) as the protocol to distribute reachability information that pertains to the VXLAN overlay network.

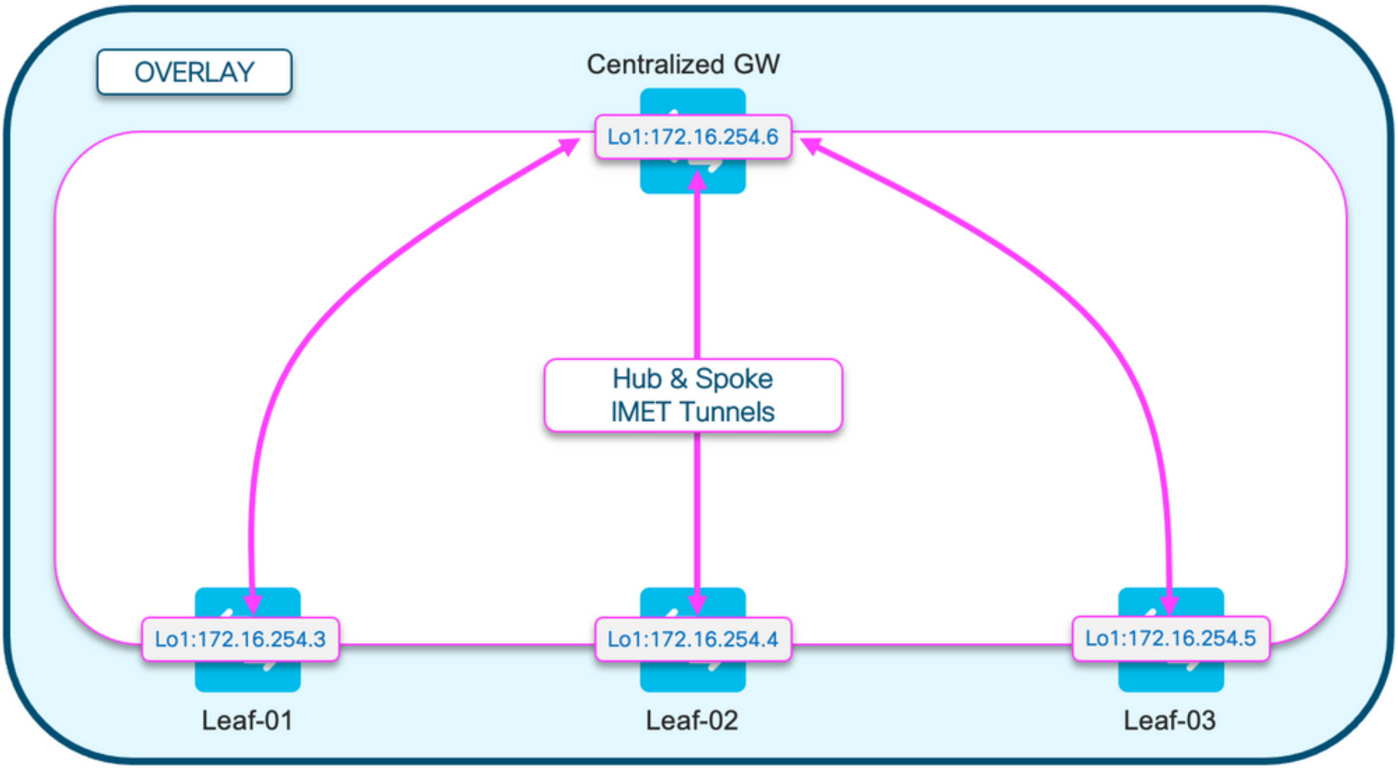
VXLAN	Virtual Extensible LAN (Local Area Network)	VXLAN is designed to overcome the inherent limitations of VLANs and STP. It is a proposed IETF standard [RFC 7348] to provide the same Ethernet Layer 2 network services as VLANs do, but with greater flexibility. Functionally, it is a MAC-in-UDP encapsulation protocol that runs as a virtual overlay on a Layer 3 underlay network.
CGW	Centralized Gateway	And implementation of EVPN where the gateway SVI are not on each leaf. Instead, all routing is done by a specific leaf using asymmetric IRB (Integrated Routing and Bridging)
DEF GW	Default Gateway	A BGP extended community attribute added to the MAC/IP prefix via the command "default-gateway advertise enable" under the 'l2vpn evpn' configuration section.
IMET (RT3)	Inclusive Multicast Ethernet Tag (Route)	Also called BGP type-3 route. This route type is used in EVPN to deliver BUM (broadcast / unknown unicast / multicast) traffic between VTEPs.
RT2	Route Type 2	BGP MAC or MAC/IP prefix that represents a host MAC or Gateway MAC-IP
EVPN Mgr	EVPN Manager	Central management component for various other components (example: learns from SISF and signals to L2RIB)
SISF	Switch Integrated Security Feature	An agnostic host tracking table that is used by EVPN to learn what local hosts are present on a Leaf
L2RIB	Layer 2 Routing Information Base	In intermediate component for managing interactions between BGP, EVPN Mgr, L2FIB
FED	Forwarding Engine Driver	Programs the ASIC (hardware) layer
MATM	Mac Address Table Manager	IOS MATM: software table which installs only local addresses and FED MATM: hardware table which installs local and remote addresses learned from control plane, and is part of the hardware forwarding plane

Flow Diagrams

Route-Type 2 (RT2) Diagram

This diagram shows the full mesh design of the type 2 MAC/MAC-IP host prefixes.

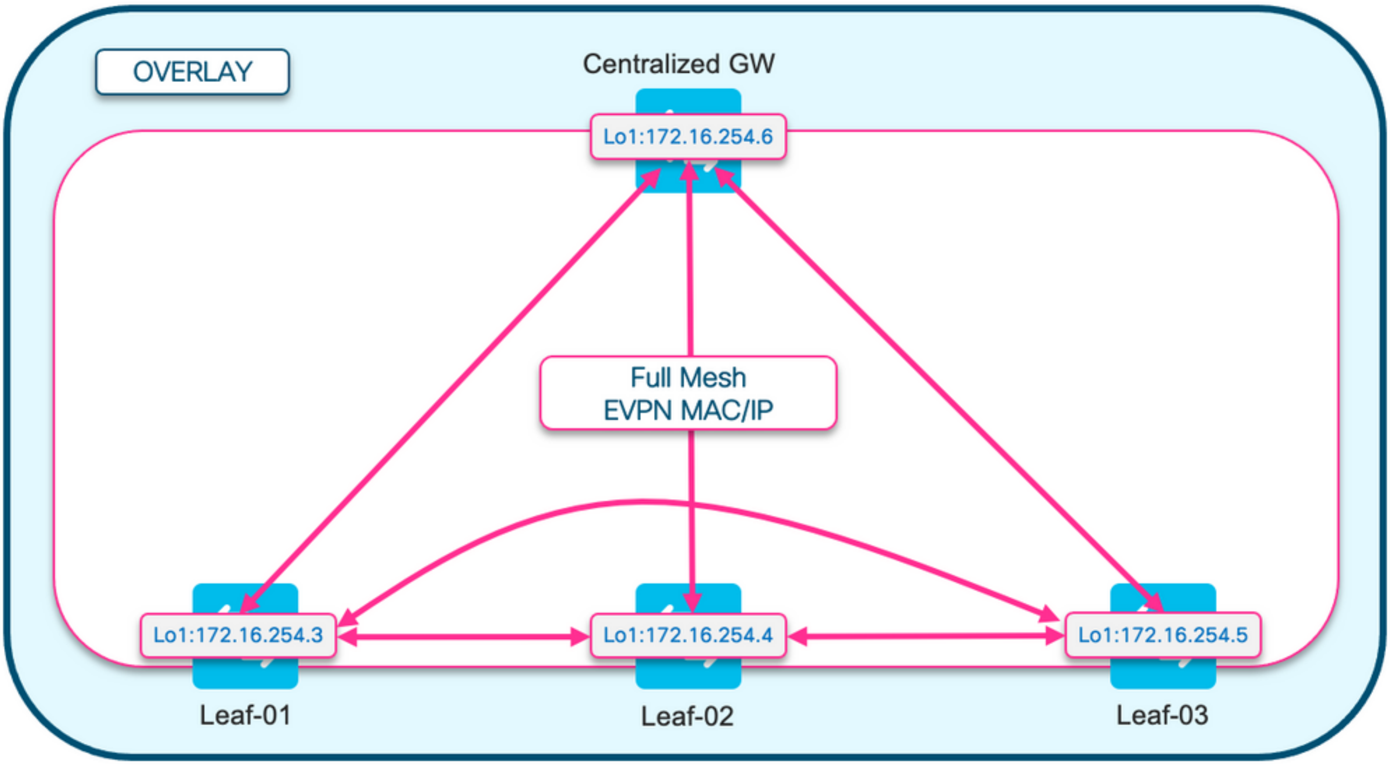
Note: Full mesh is required to support mobility and roaming



Route-Type 3 (RT3) Diagram

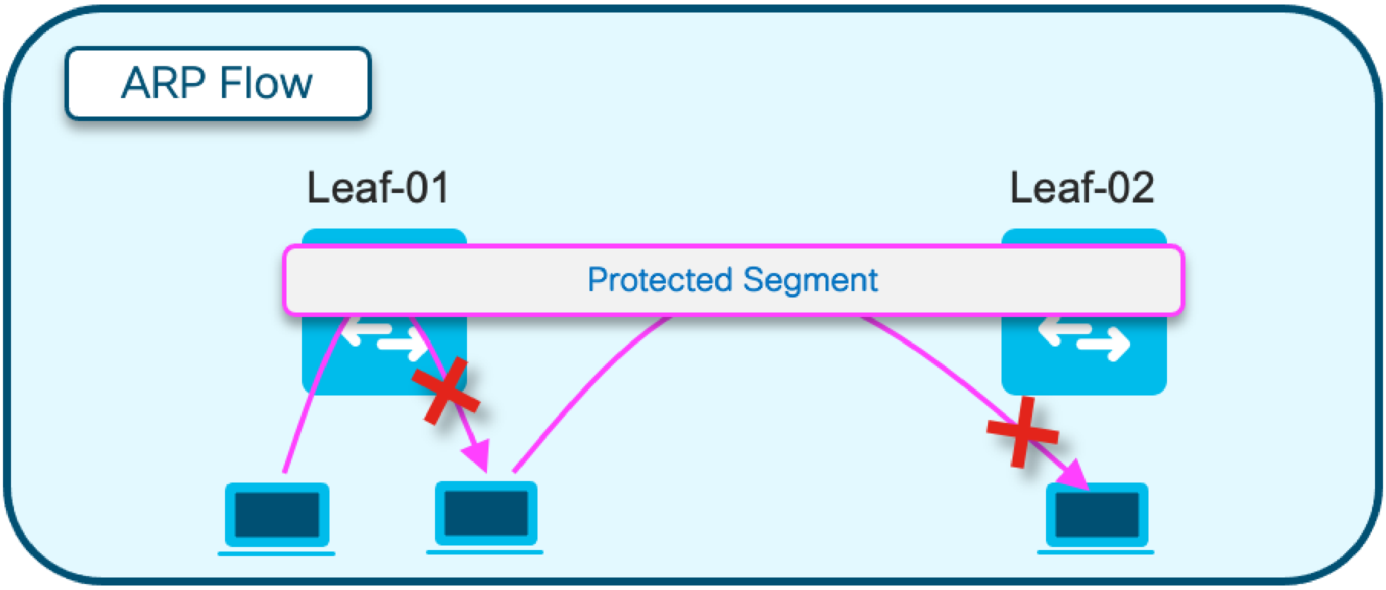
This diagram shows the hub and spoke design of the broadcast IMET (RT3) tunnels

Note: Hub and spoke broadcast is required to prevent leafs with the same segment from sending broadcast to each other directly.



Address Resolution (ARP) Diagram

This diagram demonstrates that ARP is not allowed to reach any host in the same EPVN segment. When host ARPs for another host, only the CGW gets this ARP and replies

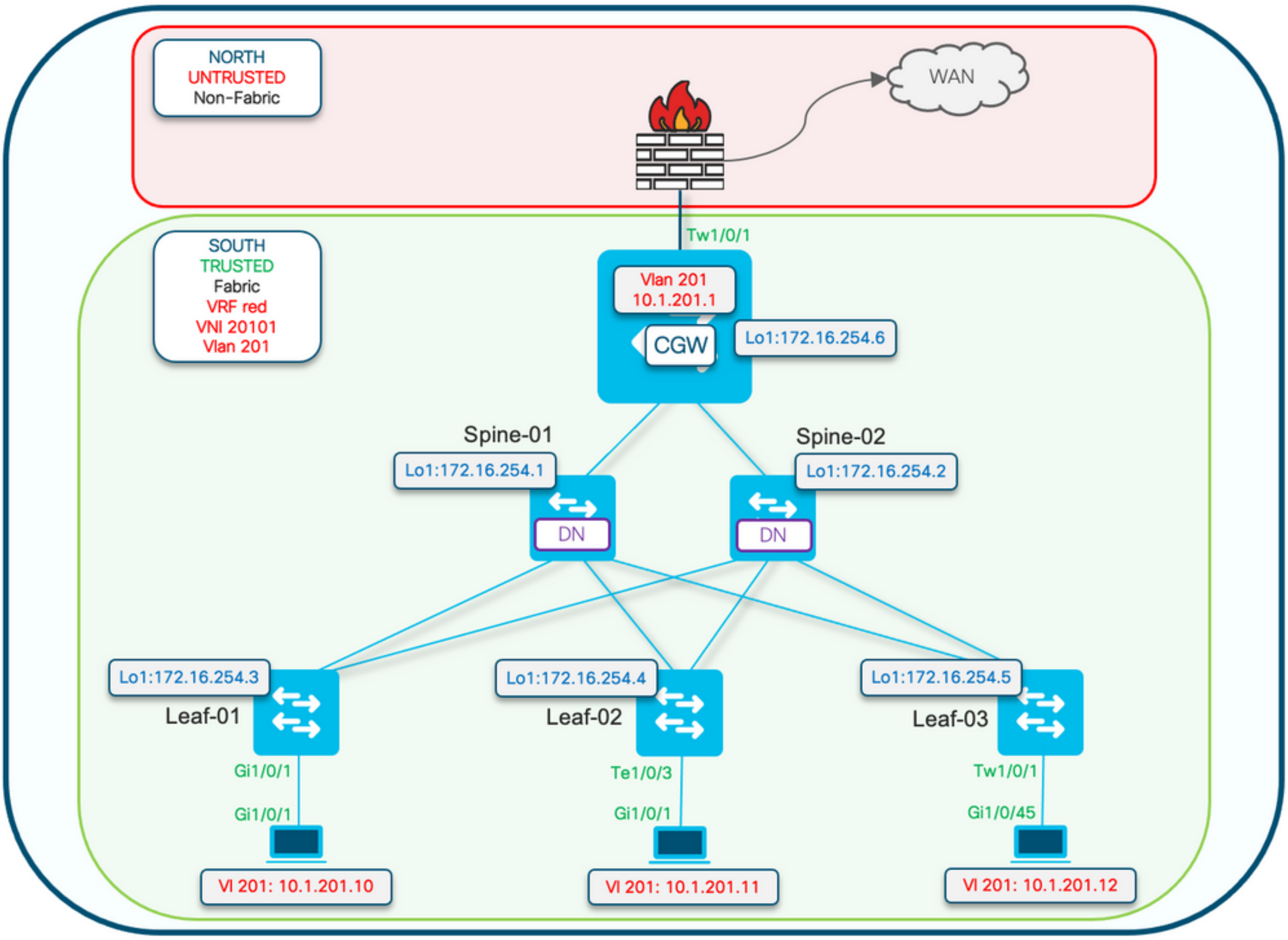


Note: This ARP behavior change is instantiated by the use of the 'protected' keyword.

Example: **member evpn-instance 202 vni 20201 protected**

Configure (Totally Isolated)

Network Diagram



Protected configuration keyword is applied on the Leaf switches. The CGW is a promiscuous device and installs all mac addresses.

Note: The routing policy community list & route-map configuration which controls the import/export of IMET prefixes is shown in Implement BGP EVPN Routing Policy on Catalyst 9000 Series Switches. Only protected segment differences are shown in this document.

Leaf-01 (Base EVPN Config)

<#root>

```
Leaf-01#

show run | sec l2vpn
l2vpn evpn

 replication-type static

 flooding-suppression address-resolution disable  <-- Disables ARP caching so ARP is always sent up to t

 router-id Loopback1
l2vpn evpn

instance 201
```

```
 vlan-based
 encapsulation vxlan

replication-type ingress                            <-- Sets segment to use Unicast replication of BUM traf

 multicast advertise enable
```

<#root>

```
Leaf01#

show run | sec vlan config

vlan configuration 201
 member evpn-instance 201 vni 20101

protected  <-- protected keyword added
```

CGW (Base Config)

<#root>

```
CGW#

show running-config | beg l2vpn evpn instance 201

l2vpn evpn instance 201 vlan-based
 encapsulation vxlan
 replication-type ingress

 default-gateway advertise enable    <-- adds the BGP attribute EVPN DEF GW:0:0 to the MAC/IP prefix

 multicast advertise enable
```

<#root>

```
CGW#

show running-config | sec vlan config

vlan configuration 201
 member evpn-instance 201 vni 20101
```

<#root>

```
CGW#

show run int nve 1

Building configuration...

Current configuration : 313 bytes
!
interface nve1
 no ip address
 source-interface Loopback1
 host-reachability protocol bgp

 member vni 20101 ingress-replication local-routing  <-- 'ingress-replication' (Unicast all BUM traffic)
```

<#root>

```
CGW#

show run interface vlan 201

Building configuration...

Current configuration : 231 bytes
!
interface Vlan201


mac-address 0000.beef.cafe          <-- MAC is static in this example for viewing simplicity. This is no

 vrf forwarding red                  <-- SVI is in VRF red

 ip address 10.1.201.1 255.255.255.0
 no ip redirects

 ip local-proxy-arp                 <-- Sets CGW to Proxy reply even for local subnet ARP requests

 ip pim sparse-mode

 ip route-cache same-interface      <-- This is auto added when local-proxy-arp is configured. However,

 ip igmp version 3
 no autostate
```

Note: At the CGW there is no BGP policy applied. The CGW is allowed to receive and send all prefix types (RT2, RT5 / RT3).

Verify (Totally Isolated)

EVI Details

<#root>

```
Leaf01#
```

```
sh l2vpn evpn evi 201 detail
```

```
EVPN instance:       201 (VLAN Based)
  RD:                172.16.254.3:201 (auto)
  Import-RTs:        65001:201
  Export-RTs:        65001:201
  Per-EVI Label:     none
  State:             Established
  Replication Type:  Ingress
  Encapsulation:     vxlan
  IP Local Learn:    Enabled (global)
  Adv. Def. Gateway: Disabled (global)
  Re-originate RT5:  Disabled
  Adv. Multicast:    Enabled
  AR Flood Suppress: Disabled (global)


Vlan:               201
    Protected:        True (local access p2p blocked)  <-- Vlan 201 is in protected mode

<...snip...>
```

Local RT2 Generation (Local Host to RT2)

Verify the component dependency chain from local host learning to RT2 generation:

- **SISF** (While the Leaf does not have an SVI, SISF still gleans the host info via ARP frame from the host)
- **EVPN Mgr**
- **L2RIB**
- **BGP**

Tip: If a previous component is not properly programmed the whole dependency chain breaks (example: SISF does not have en entry then BGP cannot create an RT2).

SISF

Verify SISF has the host learned in DB (Host info learned from DHCP or ARP)

- SISF learns MAC entries from IOS-MATM learning then sends up to EVPN Mgr (must be MAC-REACHABLE with policy "evpn-sisf-policy").
- SISF gleans an IP/MAC binding on a local VTEP and using EVPN manager that information is expected to be programmed as a /32 route via BGP to other leafs.

Note: In this scenario the host has a static IP, so SISF uses ARP to glean the host details. In the Mostly Isolated section DHCP and DHCP snooping is shown.

<#root>

```
Leaf01#

show device-tracking database vlanid 201

vlanDB has 1 entries for vlan 201, 1 dynamic
Codes: L - Local, S - Static, ND - Neighbor Discovery, ARP - Address Resolution Protocol, DH4 - IPv4 DH
Preflevel flags (prlvl):
0001:MAC and LLA match     0002:Orig trunk            0004:Orig access
0008:Orig trusted trunk    0010:Orig trusted access   0020:DHCP assigned
0040:Cga authenticated     0080:Cert authenticated    0100:Statically assigned


    Network Layer Address                    Link Layer Address     Interface  vlan       prlvl      ag

ARP

 10.1.201.10

0006.f601.cd43
```

```
Gi1/0/1

    201        0005       3mn        REACHABLE  86 s

<-- Gleaned from local host ARP Request
```

EVPN Manager

EVPN Mgr learns Local MAC and installs into L2RIB. EVPN Mgr also learns the Remote MAC from L2RIB, but entry is used only for processing MAC mobility

Confirm EVPN Mgr is updated with the SISF entry

<#root>

```
Leaf01#

show l2vpn evpn mac evi 201

MAC Address    EVI   VLAN  ESI                      Ether Tag  Next Hop(s)
-------------- ----- ----- ------------------------ ---------- ---------------

0006.f601.cd43 201   201

0000.0000.0000.0000.0000 0

         Gi1/0/1:201    <-- MAC in VLan 201 local interface Gi1/0/1:service instance 201

<...snip...>
```

L2RIB

- L2RIB learns local MAC from EVPN Mgr and sends to BGP and L2FIB.
- L2RIB is also responsible for learning remote MACs from BGP to update EVPN Mgr and L2FIB.
- L2RIB needs both Local and remote for other components to be properly updated.
- L2RIB component sits between local and remote MAC learning depending on which direction / component needs to be updated

Verify L2RIB is updated with the local MAC from EVPN Mgr

<#root>

```
Leaf01#

show l2route evpn mac topology 201      <-- View the overall topology for this segment

  EVI       ETag

Prod

    Mac Address                                        Next Hop(s) Seq Number
----- ---------- ----- -------------- ---------------------------------------------------- ----------
  201          0
```

```
BGP

 0000.beef.cafe                              V:20101 172.16.254.6         0

<-- produced by BGP who updated L2RIB (remote learn)

  201         0

L2VPN

 0006.f601.cd43                                       Gi1/0/1:201         0

<-- produced by EVPN Mgr who updated L2RIB (local learn)



Leaf01#

show l2route evpn mac mac-address 0006.f601.cd43 detail

EVPN Instance:             201

Ethernet Tag:              0

Producer Name:             L2VPN                <-- Produced by local

MAC Address:               0006.f601.cd43       <-- Host MAC Address

Num of MAC IP Route(s):    1
Sequence Number:           0
ESI:                       0000.0000.0000.0000.0000
Flags:                     B()

Next Hop(s):               Gi1/0/1:201 (E-LEAF) <-- Port:Instance and info about the Role (Leaf)
```

BGP

Verify BGP is updated by L2RIB

<#root>

```
Leaf01#

show bgp l2vpn evpn route-type 2 0 0006.f601.cd43 *

BGP routing table entry for [2][172.16.254.3:201][0][48][0006F601CD43][0][*]/20, version 268232
Paths: (1 available, best #1,

table evi_201

)

<-- In the totally isolated evi context

  Advertised to update-groups:
     2
  Refresh Epoch 1
  Local


 0.0.0.0 (via default) from 0.0.0.0

 (172.16.255.3)

<-- from 0.0.0.0 indicates local
```

```
      Origin incomplete, localpref 100, weight 32768, valid, sourced,
```

local

```
, best
```

<-- also indicates local

```
      EVPN ESI: 00000000000000000000, Label1 20101
      Extended Community: RT:65001:201 ENCAP:8
```

EVPN E-Tree:flag:1

```
,label:0
```

<-- EVPN e-Tree attribute with Leaf flag = 1 (added to indicate this is a host address)

```
      Local irb vxlan vtep:
        vrf:not found, l3-vni:0
        local router mac:0000.0000.0000
        core-irb interface:(not found)
```

vtep-ip:172.16.254.3   <-- Local VTEP Loopback

```
      rx pathid: 0, tx pathid: 0x0
      Updated on Sep 14 2023 20:16:17 UTC
```

Remote RT2 Learning (Default Gateway RT2)

BGP

Verify BGP has learned the CGW RT2 prefix

<#root>

```
Leaf01#
```

show bgp l2vpn evpn route-type 2 0 0000.beef.cafe 10.1.201.1

```
BGP routing table entry for [2][172.16.254.3:201][0][48][0000BEEFCAFE][32][10.1.201.1]/24, version 1141
Paths: (1 available, best #1,
```

table evi_201

```
)
```

<-- EVI context is 201

```
  Flag: 0x100
  Not advertised to any peer
  Refresh Epoch 2
  Local, imported path from [2][172.16.254.6:201][0][48][0000BEEFCAFE][32][10.1.201.1]/24 (global)
    172.16.254.6 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000,
```

Label1 20101   <-- Correct segment identifier

```
      Extended Community: RT:65001:201 ENCAP:8
```

```
EVPN DEF GW:0:0    <-- Default gateway attribute is added via the 'default gateway advertise CLI'

      Originator: 172.16.255.6, Cluster list: 172.16.255.1
      rx pathid: 0, tx pathid: 0x0
      Updated on Sep 1 2023 15:27:45 UTC
```

L2RIB

Verify BGP updated L2RIB

- L2RIB learns local MAC from EVPN Mgr and sends to BGP and L2FIB. L2RIB is also responsible for learning remote MACs from BGP to update EVPN Mgr and L2FIB.
- L2RIB needs both Local and remote for other components to be properly updated.
- L2RIB component sits between local and remote MAC learning depending on which direction & component needs to be updated.

<#root>

```
Leaf01#

show l2route evpn default-gateway host-ip 10.1.201.1

  EVI       ETag  Prod    Mac Address                                Host IP
----- ---------- ----- -------------- --------------------------------------- -------------------------


201

          0

BGP



0000.beef.cafe



10.1.201.1

                              V:20101 172.16.254.6

<-- L2RIB has the MAC-IP of the Gateway programmed
```

L2FIB

Verify in L2FIB

- Component responsible for updating FED with the MACs to program in hardware.
- Remote MAC entries installed by L2FIB into FED-MATM are NOT punted to IOS-MATM. (IOS-MATM shows only local MACs, whereas FED-MATM displays both local and remote MAC).
- L2FIB output only shows remote MACs (It is not responsible for programming local MACs).

<#root>

```
Leaf01#
```

```
show l2fib bridge-domain 201 address unicast 0000.beef.cafe

MAC Address                   :

0000.beef.cafe                                            <-- CGW MAC

Reference Count               : 1
Epoch                         : 0

Producer                      : BGP                                                       <-- Learned fro

Flags                         : Static
Adjacency                     :

VXLAN_UC

  PL:2973(1) T:VXLAN_UC [MAC]20101:

172.16.254.6  <-- CGW Loopback IP

PD Adjacency                  : VXLAN_UC  PL:2973(1) T:VXLAN_UC [MAC]20101:172.16.254.6
Packets                       : 6979
Bytes                         : 0
```

FED

Verify in FED MATM

- At the hardware level of the Leafs configured with the 'protected keyword' you should only see the CGW default gateway MAC and the local host MACs.
- The switch looks at the RT2 prefix for the DEF GW attribute in order to determine what remote MAC is eligible to install.

<#root>

```
Leaf01#

show platform software fed switch active matm macTable vlan 201

VLAN   MAC

Type

  Seq#    EC_Bi   Flags  machandle           siHandle            riHandle            diHandle

Con

----------------------------------------------------------------------------------------------------------

201    0000.beef.cafe



0x5000001

       0      0     64  0x7a199d182498      0x7a199d183578

0x71e059173e08

      0x0                        0       82

VTEP 172.16.254.6
```

```
 adj_id 9

No
<-- Only remote MAC installed in Fed is the Default Gateway (0x5000001 type) Conn = No (meaning not dire

201    0006.f601.cd01

0x1

    2458      0      0  0x7a199d1a2248      0x7a199d19eef8      0x0                 0x7a199c6f7cd8


201    0006.f601.cd43        0x1    8131      0      0  0x7a199d195a98      0x7a199d19eef8      0x0

<-- Two local MAC addresses (0x1 type) Conn = Yes (directly connected)



Total Mac number of addresses:: 5
Summary:
Total number of secure addresses:: 0
Total number of drop addresses:: 0
Total number of lisp local addresses:: 0
Total number of lisp remote addresses:: 3
*a_time=aging_time(secs)  *e_time=total_elapsed_time(secs)
Type:

MAT_DYNAMIC_ADDR           0x1

  MAT_STATIC_ADDR            0x2  MAT_CPU_ADDR               0x4  MAT_DISCARD_ADDR           0x8
MAT_ALL_VLANS              0x10  MAT_NO_FORWARD             0x20  MAT_IPMULT_ADDR            0x40  MAT_RES
MAT_DO_NOT_AGE            0x100  MAT_SECURE_ADDR           0x200  MAT_NO_PORT               0x400  MAT_DRO
MAT_DUP_ADDR             0x1000  MAT_NULL_DESTINATION     0x2000  MAT_DOT1X_ADDR           0x4000  MAT_ROU
MAT_WIRELESS_ADDR       0x10000  MAT_SECURE_CFG_ADDR     0x20000  MAT_OPQ_DATA_PRESENT    0x40000  MAT_WIR
MAT_DLR_ADDR           0x100000  MAT_MRP_ADDR           0x200000  MAT_MSRP_ADDR          0x400000  MAT_LIS

MAT_LISP_REMOTE_ADDR 0x1000000

  MAT_VPLS_ADDR        0x2000000

MAT_LISP_GW_ADDR     0x4000000

<-- the addition of these values = 0x5000001

MAT_LISP_REMOTE_ADDR 0x1000000
MAT_LISP_GW_ADDR 0x4000000
MAT_DYNAMIC_ADDR 0x1
```

Data Plane Adjacency

As a final step after confirming FED entry you can resolve the rewrite index (RI)

<#root>

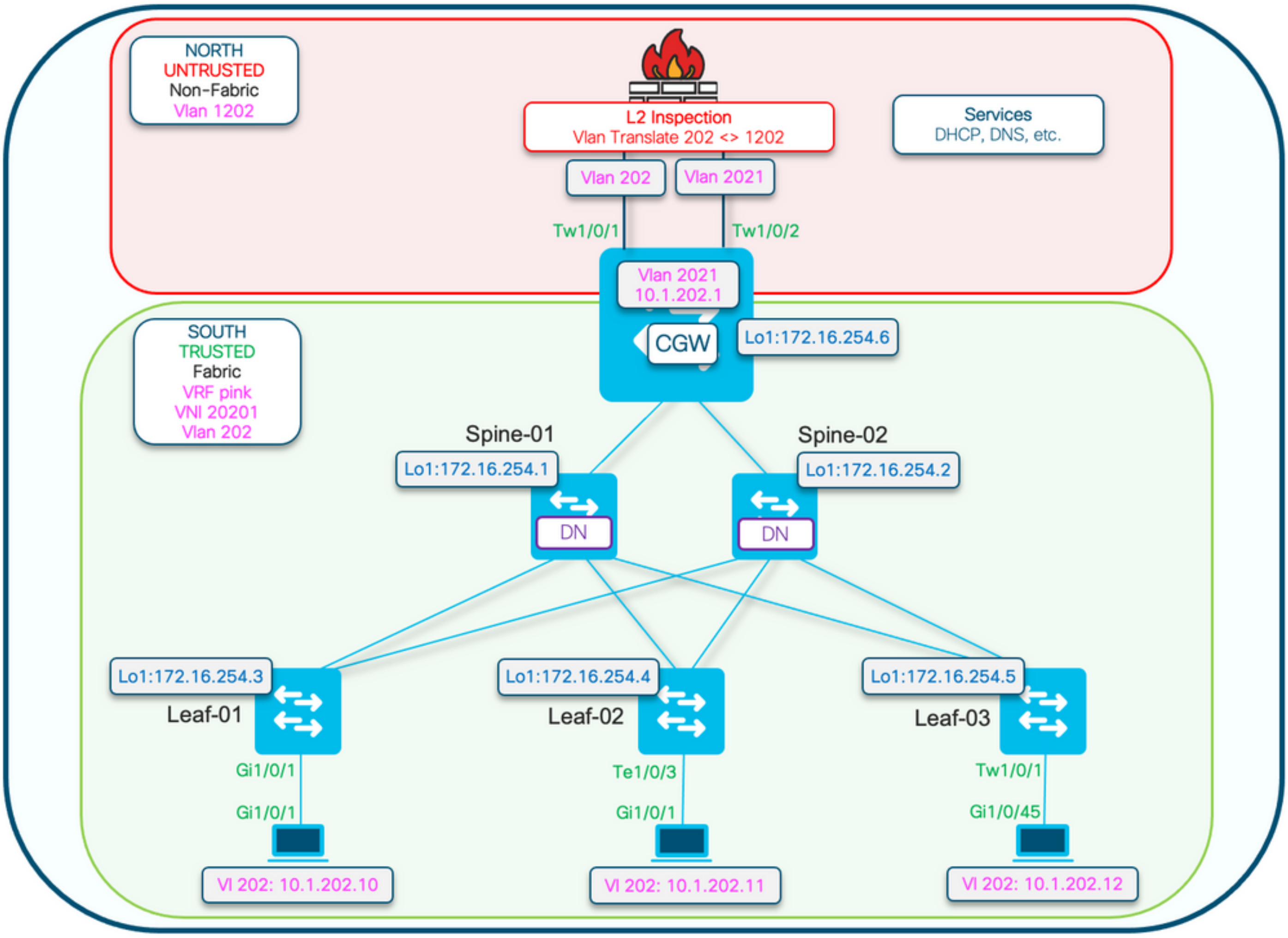
```
Leaf01#

sh platform hardware fed switch active fwd-asic abstraction print-resource-handle 0x71e059173e08 0
<-- 0x71e059173e08 is taken from previous FED command riHandle for the CGW MAC

Handle:0x71e059173e08 Res-Type:ASIC_RSC_RI Res-Switch-Num:255 Asic-Num:255 Feature-ID:AL_FID_L2_WIRELES
priv_ri/priv_si Handle: 0x71e05917b8d8Hardware Indices/Handles: index0:0x38  mtu_index/l3u_ri_index0:0x
Features sharing this resource:58 (1)]
```

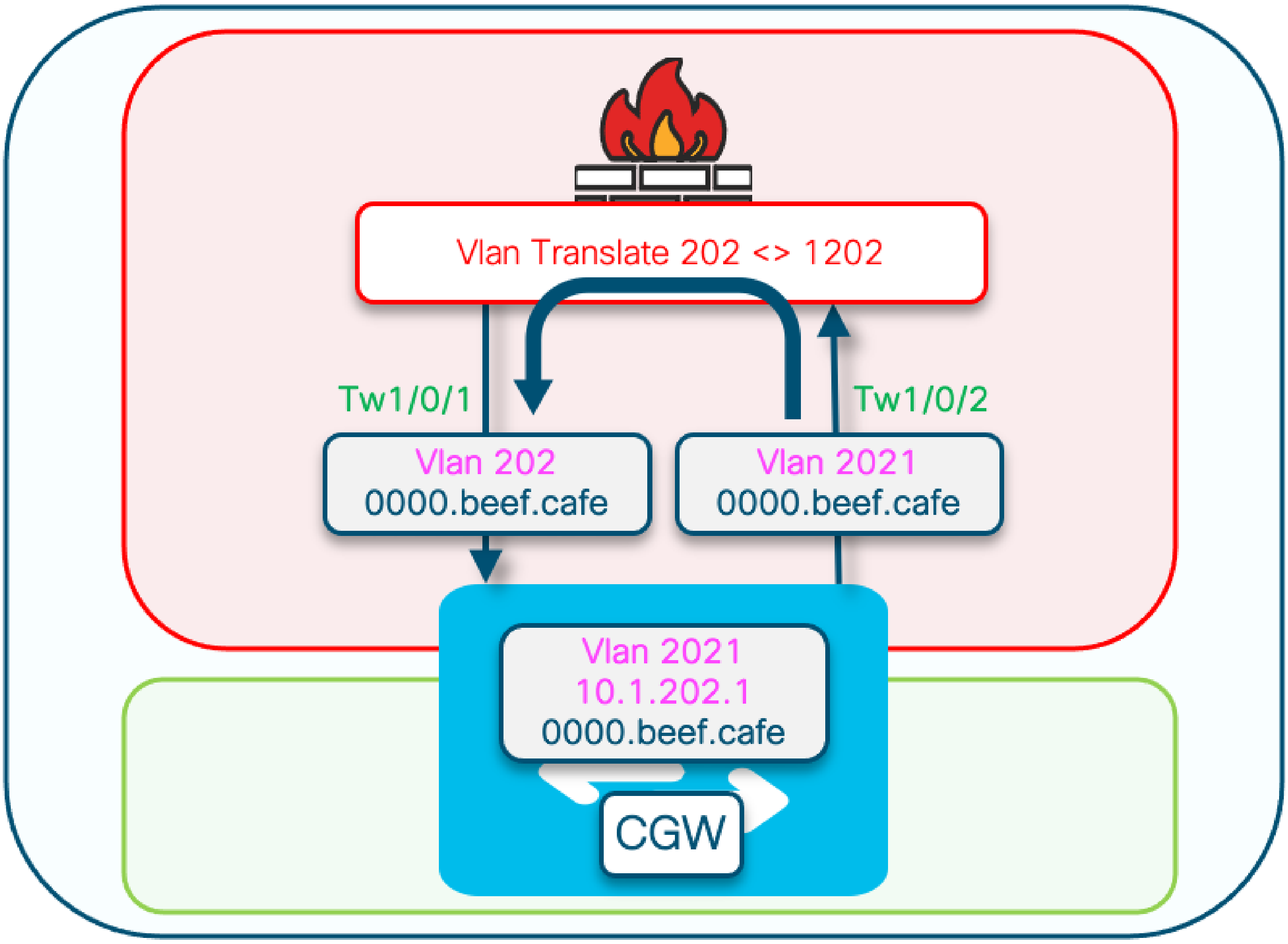
```
Brief Resource Information (ASIC_INSTANCE# 0)
----------------------------------------
ASIC#:0 RI:56 Rewrite_type:AL_RRM_REWRITE_LVX_IPV4_L2_PAYLOAD_ENCAP_EPG(116) Mapped_rii:LVX_L3_ENCAP_L2_

 Src IP:     172.16.254.3      <-- source tunnel IP
 Dst IP:     172.16.254.6      <-- dest tunnel IP

 iVxlan dstMac:     0x9db:0x00:0x00
 iVxlan srcMac:     0x00:0x00:0x00
 IPv4 TTL:     0
 iid present:     0

 lisp iid:     20101            <-- Segment 20101

 lisp flags:     0

 dst Port:     4789             <-- VxLAN

 update only l3if:     0
 is Sgt:     0
 is TTL Prop:     0
 L3if LE:     53 (0)
 Port LE:     281 (0)
 Vlan LE:     8 (0)
```

Note: You can also use 'show platform software fed switch active matm macTable vlan 201 detail' which chains this command with the FED command into one result

Configure (Partially Isolated)

Network Diagram


NORTH
UNTRUSTED
Non-Fabric
Vlan 1202
L2 Inspection
Vlan Translate 202 <> 1202
Services
DHCP, DNS, etc.
Vlan 202
Vlan 2021
Tw1/0/1
Tw1/0/2
Vlan 2021
10.1.202.1
CGW
Lo1:172.16.254.6
SOUTH
TRUSTED
Fabric
VRF pink
VNI 20201
Vlan 202
Spine-01
Lo1:172.16.254.1
DN
Spine-02
Lo1:172.16.254.2
DN
Lo1:172.16.254.3
Leaf-01
Lo1:172.16.254.4
Leaf-02
Lo1:172.16.254.5
Leaf-03
Gi1/0/1
Gi1/0/1
Te1/0/3
Gi1/0/1
Tw1/0/1
Gi1/0/45
Vl 202: 10.1.202.10
Vl 202: 10.1.202.11
Vl 202: 10.1.202.12



Vlan Translate 202 <> 1202
Tw1/0/1
Tw1/0/2
Vlan 202
0000.beef.cafe
Vlan 2021
0000.beef.cafe
Vlan 2021
10.1.202.1
0000.beef.cafe
CGW

Note: This section only covers differences from Totally Isolated Segments.

- Routing-policy to mark the GCW gateway MAC IP with the DEF GW attribute
- Custom Device tracking policy required to prevent MAC flaps
- Static device-tracking binding for the GW MAC IP

Leaf-01 (Base EVPN Config)

<#root>

```
Leaf-01#

show run | sec l2vpn
l2vpn evpn

 replication-type static

 flooding-suppression address-resolution disable  <-- Disables ARP caching so ARP is always sent up to t

 router-id Loopback1
l2vpn evpn
```

```
instance 202

 vlan-based
 encapsulation vxlan


replication-type ingress

 multicast advertise enable
```

<#root>

```
Leaf01#

show run | sec vlan config

vlan configuration 202
 member evpn-instance 202 vni 20201

protected <-- protected keyword added
```

CGW (Base Config)

Set the replication mode under the nve

<#root>

```
CGW#

show run int nve 1

Building configuration...

Current configuration : 313 bytes
!
interface nve1
 no ip address
 source-interface Loopback1
 host-reachability protocol bgp

 member vni 20201 ingress-replication local-routing <-- 'ingress-replication' (Unicast all BUM traffic)

end
```

Configure the external gateway SVI

<#root>

```
CGW#

show run interface vlan 2021

Building configuration...

Current configuration : 231 bytes
!
```

```
interface Vlan2021

mac-address 0000.beef.cafe         <-- MAC is static in this example for viewing simplicity. This is no

 vrf forwarding pink                 <-- SVI is in VRF pink

 ip address 10.1.202.1 255.255.255.0
 no ip redirects

 ip local-proxy-arp                <-- Sets CGW to Proxy reply even for local subnet ARP requests

 ip pim sparse-mode

 ip route-cache same-interface     <-- This is auto added when local-proxy-arp is configured. However,

 ip igmp version 3
 no autostate
end
```

Create a policy with gleaning disabled

<#root>

```
device-tracking policy dt-no-glean

 <-- Configure device tracking policy to prevent MAC-IP flapping

 security-level glean
 no protocol ndp
 no protocol dhcp6
 no protocol arp
 no protocol dhcp4
```

Attach to externalgatewayevi/vlans

<#root>

```
CGW#

show running-config | sec vlan config

vlan configuration 202
 member evpn-instance 202 vni 20201

 device-tracking attach-policy dt-no-glean <-- apply the new device tracking policy to the vlan configur
```

Add static entries into device tracking table for externalgateway mac-ip

<#root>

```
device-tracking binding vlan 202 10.1.202.1 interface TwentyFiveGigE1/0/1 0000.beef.cafe


<-- All static entries in device tracking table should be for external gateway mac-ip’s.
    If there is any other static entry in device tracking table, match ip/ipv6 configurations in route m
```

Create BGP route map to match RT2 MAC-IP prefixes and set the default gateway extendedcommunity

<#root>

```
route-map CGW_DEF_GW permit 10

match evpn route-type 2-mac-ip <-- match RT2 type MAC-IP


 set extcommunity default-gw    <-- Set Default-gateway (DEF GW 0:0) extended community

route-map CGW_DEF_GW permit 20
```

Apply route-map to BGP Route Reflector neighbors

<#root>

```
CGW#

sh run | s r bgp

address-family l2vpn evpn
 neighbor 172.16.255.1 activate
 neighbor 172.16.255.1 send-community both
 neighbor 172.16.255.1

route-map CGW_DEF_GW out  <-- Sets the DEF GW Community when it advertises MAC-IP type RT2 to the RR

 neighbor 172.16.255.2 activate
 neighbor 172.16.255.2 send-community both
 neighbor 172.16.255.2

route-map CGW_DEF_GW out <-- Sets the DEF GW Community when it advertises MAC-IP type RT2 to the RR
```

Verify (Partially Isolated)

EVI Details

<#root>

```
Leaf01#

show l2vpn evpn evi 202 detail

EVPN instance:       202 (VLAN Based)
  RD:                172.16.254.3:202 (auto)
  Import-RTs:        65001:202
  Export-RTs:        65001:202
  Per-EVI Label:     none
  State:             Established
  Replication Type:  Ingress
  Encapsulation:     vxlan
  IP Local Learn:    Enabled (global)
  Adv. Def. Gateway: Enabled (global)
  Re-originate RT5:  Disabled
  Adv. Multicast:    Enabled

  Vlan:              202
    Protected:       True (local access p2p blocked)  <-- Vlan 202 is in protected mode

<...snip...>
```

Local RT2 Generation (Local Host to RT2)

Covered in previous Totally Isolated Example

Remote RT2 Learning (Default Gateway RT2)

Covers the differences from Totally Isolated

CGW Default Gateway Prefix (Leaf)

Check that the prefix has the appropriate attribute in order to be eligible to be installed into hardware

Note: This is critical for DHCP L2 Relay to function

<#root>

```
Leaf01#
```

```
show bgp l2vpn evpn route-type 2 0 0000.beef.cafe 10.1.202.1
```

```
BGP routing table entry for [2][172.16.254.3:202][0][48][0000BEEFCAFE][32][10.1.202.1]/24, version 1846
Paths: (1 available, best #1,
```

```
table evi_202
```

```
)
```

```
<-- the EVI context of 202 which matches the Vlan/EVI we are concerned about
```

```
  Not advertised to any peer
  Refresh Epoch 2
  Local, imported path from [2][172.16.254.6:202][0][48][0000BEEFCAFE][32][10.1.202.1]/24 (global)
    172.16.254.6 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000,
Label1 20201              <-- Correct Segment ID

      Extended Community: RT:65001:202 ENCAP:8

EVPN DEF GW:0:0  <-- prefix has the Default GW attribute added

      Originator: 172.16.255.6, Cluster list: 172.16.255.1
      rx pathid: 0, tx pathid: 0x0
      Updated on Sep 7 2023 19:56:43 UTC
```

FED MATM (Leaf)

<#root>

```
F241.03.23-9300-Leaf01#

show platform software fed active matm macTable vlan 202 mac 0000.beef.cafe

VLAN   MAC                    Type  Seq#    EC_Bi  Flags  machandle           siHandle            riHandl
-------------------------------------------------------------------------------------------------------------

202    0000.beef.cafe

  0x5000001       0      0     64  0x71e058da7858      0x71e05916c0d8      0x71e059171678      0x0

VTEP 172.16.254.6

 adj_id 651

No
<-- MAC of Default GW is installed in FED
```

SISF (CGW)

<#root>

```
CGW#

sh device-tracking database vlanid 202

vlanDB has 1 entries for vlan 202, 0 dynamic
Codes: L - Local, S - Static, ND - Neighbor Discovery, ARP - Address Resolution Protocol, DH4 - IPv4 DH
Preflevel flags (prlvl):
0001:MAC and LLA match     0002:Orig trunk            0004:Orig access
0008:Orig trusted trunk    0010:Orig trusted access   0020:DHCP assigned
0040:Cga authenticated     0080:Cert authenticated    0100:Statically assigned


    Network Layer Address                    Link Layer Address     Interface  vlan       prlvl      ag
S   10.1.202.1                               0000.beef.cafe         Twe1/0/1   202        0100       13
```

IOS MATM (CGW)

<#root>

```
CGW#

show mac address-table address 0000.beef.cafe

          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
 201    0000.beef.cafe    STATIC      Vl201

2021    0000.beef.cafe    STATIC      Vl2021   <-- The Vlan 2021 SVI MAC advertised out Tw1/0/1

 202    0000.beef.cafe    DYNAMIC     Twe1/0/1 <-- The Vlan 2021 SVI MAC learned dynamically after passi
```

Troubleshoot

Address Resolution (ARP)

General steps for isolating ARP issues

- Confirm IMET tunnel is ready
- Capture on CGW Uplink to verify ARP received Encapsulated from Leaf
- If no ARP seen arriving encap on uplink
  - Verify IMET tunnel is ready on both Leaf and CGW
  - Capture on Leaf uplinks to confirm ARP is encapsulated and sent
  - Troubleshoot intermediate path
- If ARP arrives on Border IMET tunnel capture but not programmed in VRF ARP table
  - Troubleshoot CPU/CoPP punt path to confirm ARP punted to CPU
  - Confirm IP address / client info is correct
  - Debug ARP in VRF to see what might be impacting ARP process
- Verify CGW MAC installed as next hop / dest mac on the hosts
- Confirm CGW has both ARP entries with the real host MACs
- Verify firewall policy allows this tyep of traffic

Caution: Be careful when enabling debugs!

Ensure you have disabled flooding suppression

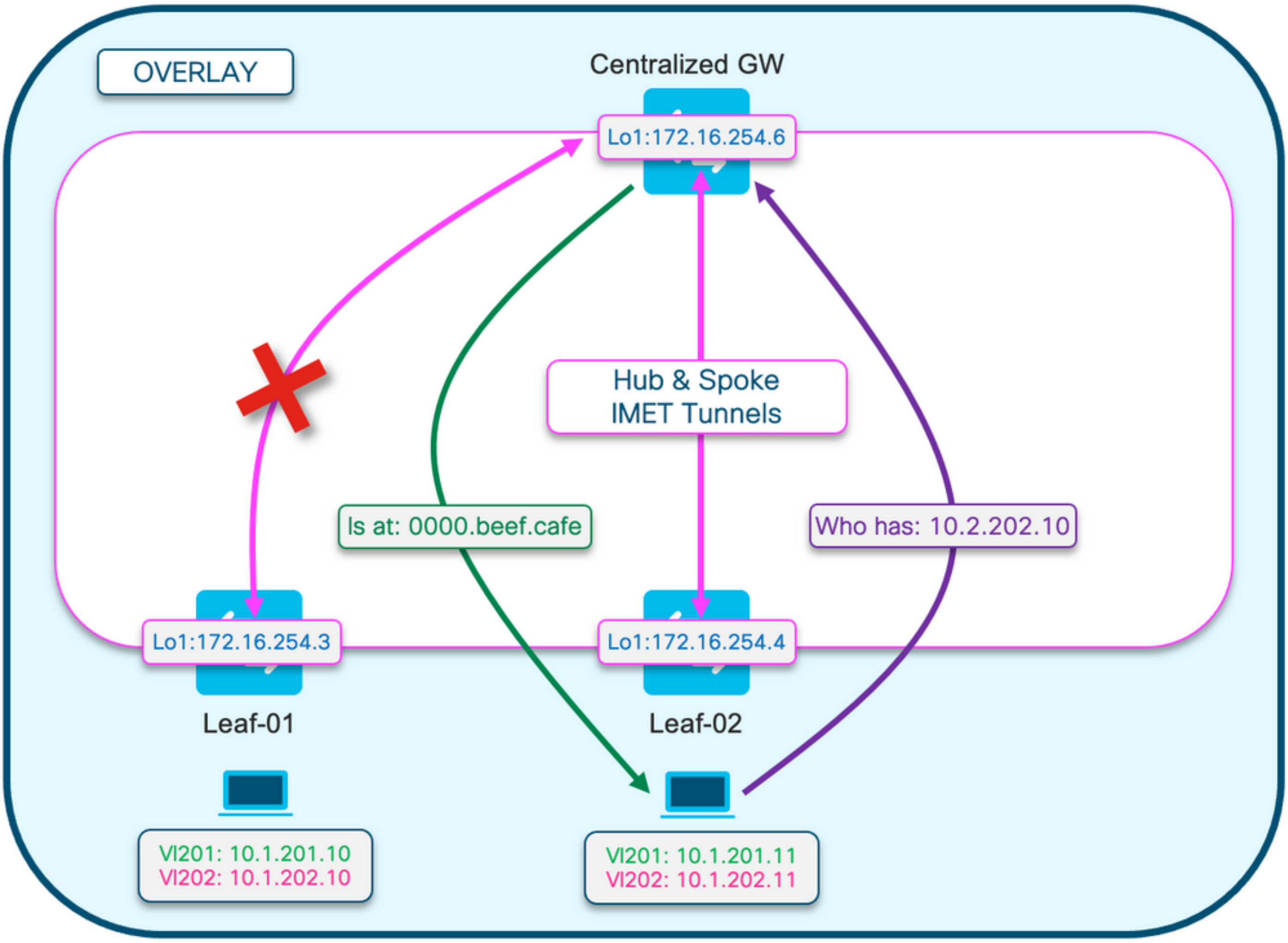
<#root>

```
Leaf-01#

show run | sec l2vpn
l2vpn evpn

 replication-type static

 flooding-suppression address-resolution disable <-- This CLI prevents a VTEP from trying to unicast oth
```

When host off Leaf-02 resolves ARP for host off Leaf-01 the ARP request is not broadcast to Leaf-01 directly

- The ARP is instead passed up the only BUM tunnel programmed on Leaf-02 toward the CGW

- The CGW does not forward this to Leaf-01, and instead replies with its own MAC
- This causes all communication to be passed up to the CGW then routed to between the hosts
- CGW routes packets, even when they are on the same local subnet



This diagram is to help visualize the flow of the ARP resolution process described in this section.

The ARP Request is show in purple

- This ARP request is to resolve the MAC address of the hos 10.1.202.10 off Leaf-01
- Notice that the purple line terminates at the CGW, and does not reach Leaf-01

The ARP Reply is shown in green

- The reply contains the MAC of the CGW SVI for Vlan 202
- Notice that the green line comes from the CGW, not from the actual host

Note: The red X is to indicate that this communication did not involve sending traffic to Leaf-01.

Observe the ARP entries on each respective host

<#root>

```
Leaf02-HOST#

sh ip arp 10.1.202.10

Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.1.202.10            1

0000.beef.cafe

  ARPA   Vlan202

<-- MAC address for Leaf01 host is CGW MAC


Leaf01-HOST#

sh ip arp 10.1.202.11
```

```
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.1.202.11             7

0000.beef.cafe

  ARPA   Vlan202

<-- MAC address for Leaf02 host is CGW MAC
```

Observe on CGW the RT2 prefixes are learned. This is required for the CGW to route packets

<#root>

```
CGW#

sh bgp l2vpn evpn route-type 2 0 0006.f617.eec4 *                    <-- Leaf02 actual MAC

BGP routing table entry for [2][172.16.254.6:202][0][48][0006F617EEC4][0][*]/20, version 235458
Paths: (1 available, best #1,

table evi_202

)
  Not advertised to any peer
  Refresh Epoch 2
  Local, imported path from [2][172.16.254.4:202][0][48][0006F617EEC4][0][*]/20 (global)
    172.16.254.4 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000,

Label1 20201                         <-- correct segment identifier

      Extended Community: RT:65001:202 ENCAP:8

EVPN E-Tree:flag:1

,label:0

<-- prefix contains the Leaf flag indicating this is a normal host


      Originator: 172.16.255.4, Cluster list: 172.16.255.1
      rx pathid: 0, tx pathid: 0x0
      Updated on Apr 9 2025 17:11:22 UTC

CGW#

sh bgp l2vpn evpn route-type 2 0 0006.f601.cd44 *                    <-- Leaf01 actual MAC

BGP routing table entry for [2][172.16.254.6:202][0][48][0006F601CD44][0][*]/20, version 235521
Paths: (1 available, best #1,

table evi_202)

  Not advertised to any peer
  Refresh Epoch 2
  Local, imported path from [2][172.16.254.3:202][0][48][0006F601CD44][0][*]/20 (global)
    172.16.254.3 (metric 3) (via default) from 172.16.255.1 (172.16.255.1)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000,

Label1 20201                         <-- correct segment identifier

      Extended Community: RT:65001:202 ENCAP:8
```

```
EVPN E-Tree:flag:1

,label:0

<-- prefix contains the Leaf flag indicating this is a normal host


      Originator: 172.16.255.3, Cluster list: 172.16.255.1
      rx pathid: 0, tx pathid: 0x0
      Updated on Apr 9 2025 17:17:06 UTC
```

Capture the ARP exchange on the uplinks to confirm bi-directional communication

- You can use Embedded Packet Capture (EPC) on the Fabric uplinks
- This scenario shows EPC on the Leaf01 Uplink. Repeat this same process on CGW if necessary

Configure the EPC

<#root>

```
Leaf01#

monitor capture 1 interface range te 1/1/2 , te 1/1/4 both match any buffer size 100



<-- both Uplinks toward fabric included
```

Start the capture

<#root>

```
Leaf01#

monitor capture 1 start
```

Inititate ping to trigger the ARP request (In this case ping is from Leaf01 host 10.1.201.10 to Leaf02 host 10.1.201.11)

<#root>

```
Leaf01-HOST#

ping vrf red 10.1.201.11


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.201.11, timeout is 2 seconds:
...!!
Success rate is 40 percent (2/5), round-trip min/avg/max = 1/1/1 ms
```

Stop Capture & **Check** for the ARP frames

<#root>

```
Leaf01#

mon cap 1 stop



F241.03.23-9300-Leaf01#

show mon cap 1 buff br | i ARP



11

 8.153510 00:06:f6:01:cd:42 -> ff:ff:ff:ff:ff:ff ARP 110

Who has 10.1.201.11? Tell 10.1.201.10  <-- .10 requests .11 MAC (this is Frame 11)


12 8.154030 00:00:be:ef:ca:fe -> 00:06:f6:01:cd:42 ARP 110 10.1.201.11

is at 00:00:be:ef:ca:fe    <-- CGW replies with its MAC
```

View the capture packets in detail. If you want to see more info about the packest, use the detail option of EPC

- Be aware that this output is clipped in various places for brevity

<#root>

```
Leaf01#

show mon cap 1 buffer detailed | beg Frame 11   <-- begin detail result from Frame 11 (ARP Request)

Frame 11: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface /tmp/epc_ws/wif_to_t

Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)

    Destination: 00:00:00:00:00:00 (00:00:00:00:00:00)
        Address: 00:00:00:00:00:00 (00:00:00:00:00:00)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: 00:00:00:00:00:00 (00:00:00:00:00:00)
        Address: 00:00:00:00:00:00 (00:00:00:00:00:00)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 172.16.254.3, Dst: 172.16.254.6     <--- Outer tunnel IP header

    Source: 172.16.254.3
    Destination: 172.16.254.6
User Datagram Protocol, Src Port: 65483,

Dst Port: 4789  <-- VXLAN Dest port
```

```
Virtual eXtensible Local Area Network
    VXLAN Network Identifier

(VNI): 20101                <-- Verify the VNI for the segment you are investigating

    Reserved: 0

Ethernet II, Src: 00:06:f6:01:cd:42 (00:06:f6:01:cd:42), Dst: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)  <--


  Type: ARP (0x0806)

    Trailer: 000000000000000000000000000000000000
Address Resolution Protocol (

request

)

 <-- is an ARP request


    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)


Sender MAC address: 00:06:f6:01:cd:42 (00:06:f6:01:cd:42)   <-- Sending host
    Sender IP address: 10.1.201.10
    Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)   <-- Trying to resolve MAC for host
    Target IP address: 10.1.201.11



Frame 12:

 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface /tmp/epc_ws/wif_to_ts_pipe, i

<-- ARP reply


Ethernet II,

Src: dc:77:4c:8a:6d:7f

 (dc:77:4c:8a:6d:7f),

Dst: 68:2c:7b:f8:87:48

 (68:2c:7b:f8:87:48)

<-- Underlay MACs



Internet Protocol Version 4, Src: 172.16.254.6, Dst: 172.16.254.3

User Datagram Protocol, Src Port: 65410, Dst Port: 4789
Virtual eXtensible Local Area Network
    VXLAN Network Identifier (VNI): 20101
    Reserved: 0
Ethernet II,

Src: 00:00:be:ef:ca:fe
```

```
 (00:00:be:ef:ca:fe),

Dst: 00:06:f6:01:cd:42

 (00:06:f6:01:cd:42)

 <-- Start of payload



 Type: ARP

 (0x0806)
    Trailer: 000000000000000000000000000000000000
Address Resolution Protocol (

reply

)

<-- is an ARP reply


    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)


 Sender MAC address: 00:00:be:ef:ca:fe (00:00:be:ef:ca:fe)   <-- Reply is that of the CGW MAC due to loc

    Sender IP address: 10.1.201.11
    Target MAC address: 00:06:f6:01:cd:42 (00:06:f6:01:cd:42)
    Target IP address: 10.1.201.10
```

CGW RT2 Gateway Prefix

Gateway Prefix Missing

As mentioned in the previous section on Partially Isolated segments the MAC is required to be learned in the fabric Vlan

- This issue can manifest if there s no traffic destined for the gateway for longer than the MAC aging timer.
- If the CGW Gateway prefix is missing, you need to confirm the MAC is present

<#root>

```
CGW#

show bgp l2vpn evpn route-type 2 0 0000.beef.cafe 10.1.202.1

% Network not in table <-- RT2 not generated on CGW


CGW#

show mac address-table address 0000.beef.cafe

          Mac Address Table
```

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

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
 201    0000.beef.cafe    STATIC      Vl201
2021    0000.beef.cafe    STATIC      Vl2021

<-- MAC is not learned in Fabric Vlan 202

Total Mac Addresses for this criterion: 2
```

Gateway Prefix Missing Remediation

In most production networks there is likely to be some traffic at all times. However, if you are having this issue you can use one of these options to remediate the issue:

- **Add** static MAC entry such as 'mac address-table static 0000.beef.cafe vlan 202 interface TwentyFiveGigE1/0/1'
- Increase the MAC aging timer with 'mac address-table aging-time <seconds>'. (Keep in mind this increases the aging time for all MAC addresses, so the static MAC option is preferred)

Missing DEF GW Attribute

With Partially Isolated Segments there are a number of additional configurations to add this attribute.

Missing DEF GW Attribute Remediation

Confirm these details:

- You are running 17.12.1 or later
- The SISF (Device-Tracking) CLI is present in the configuration
- The route-map match & set commands are configured and route-map is applied to the BGP neighbors
- You have refreshed the BGP advertisements (you must clear BGP to re-advertise the prefix with the new attribute

Wireless Roaming

Frequent roaming can cause BGP to update too frequently & roaming per time interval should be increased before switch declares it owns the MAC and sends RT2 Update

- This occurs when a host moves between two APs that are on different switches.
- Default limit for roam is 5 per 180 seconds

<#root>

```
Leaf01#

sh run | sec l2vpn

l2vpn evpn
 replication-type static
 flooding-suppression address-resolution disable

 ip duplication limit 10 time 180          <--- You can adjust this default in the global l2vpn sectio
 mac duplication limit 10 time 180
```

```
Leaf01#

sh l2vpn evpn summary

L2VPN EVPN
  EVPN Instances (excluding point-to-point): 4
    VLAN Based:   4
  Vlans: 4
  BGP: ASN 65001, address-family l2vpn evpn configured
  Router ID: 172.16.254.3
  Global Replication Type: Static
  ARP/ND Flooding Suppression: Disabled
  Connectivity to Core: UP

  MAC Duplication: seconds 180 limit 10

  MAC Addresses: 13
    Local:     6
    Remote:    7

    Duplicate: 0
  IP Duplication: seconds 180 limit 10

  IP Addresses: 7
    Local:     4
    Remote:    3

    Duplicate: 0

<...snip...>
```

Commands to Collect for TAC

In the event this guide did not resolve your issue please collect the command list shown and attach them to your TAC service request.

Minimum info to collect

(limited time to gather data prior to reload/recovery action)

- Show tech evpn
- Show tech
- Show tech sisf

Detailed info to collect

(If there is time to collect more complete data, this is preferred)

- show tech
- show tech evpn
- show tech platform evpn_vxlan switch <number>

- show tech platform
- show tech resource
- show tech sisf
- show tech isis
- show tech bgp
- show monitor event-trace evpn event all
- show monitor event-trace evpn error all
- request platform software trace archive

Related Information

- Implement BGP EVPN Routing Policy on Catalyst 9000 Series Switches
- DHCP Layer 2 Relay (coming soon)