Implement BGP EVPN DHCP Layer 2 Relay on Catalyst 9000 Series Switches

Contents

Introduction

This document describes how to configure, verify, and troubleshoot the EVPN VxLAN DHCP L2 Relay feature.

Prerequisites

Requirements

- This feature is used in any CGW type deployment where DHCP is used
- If implementing Protected Segmentation please review these documents
  - Implement BGP EVPN Routing Policy on Catalyst 9000 Series Switches
  - Implement BGP EVPN Protected Overlay Segmentation on Catalyst 9000 Series Switches

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

Document Details

This document can be used for any CGW deployment where DHCP needs to be relayed from a Leaf with no SVI toward the Central Gateway.

- If you are not using protected segmentation use the section of the document where SVI is advertised into the fabric

If you are implementing protected segmentation this document is part 2 of 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:**Implement BGP EVPN Protected Overlay Segmentation on Catalyst 9000 Series Switches builds upon the Overlay design and policy of document 1, describes the implementation of the 'protected' keyword.
- **Document 3:** This document. Builds on top of the last two documents and describes the way DHCP relay is implemented with layer 2 only Leafs and CGW

L2 Relay Behavior

Relay	**Snooping**	**Core Flood**	**Access Flood**	**IPv4**
yes	yes	no	yes	• Option 82 Suboption: (1) Agent Circuit ID (vni-mod-port) gets populated with dhcp snooping • One can limit the access side with dhcp trust configuration * **RECOMMENDED MODEL**
yes	no	yes	yes	• Option 82 Suboption: (1) Agent Circuit ID (vlan-mod-port) gets populated with dhcp snooping
no	yes	no	yes	• Option 82 Suboption: (1) Agent Circuit ID (vni-mod-port) gets populated with dhcp snooping • One can limit the access side with dhcp trust configuration
Relay	**Snooping**	**Core Flood**	**Access Flood**	**IPv6**
yes	yes	yes	yes	• Option 82 Suboption: (1) Agent Circuit ID (vni-mod-port) gets populated with dhcp snooping • One can limit the access side with dhcp trust configuration
yes	no	yes	yes	• Option 82 Suboption: (1) Agent Circuit ID (vlan-mod-port) gets populated with dhcp snooping
no	yes	yes	yes	• Option 82 Suboption: (1) Agent Circuit ID (vni-mod-port) gets populated with dhcp snooping • One can limit the access side with dhcp trust configuration
no	no	yes	yes	

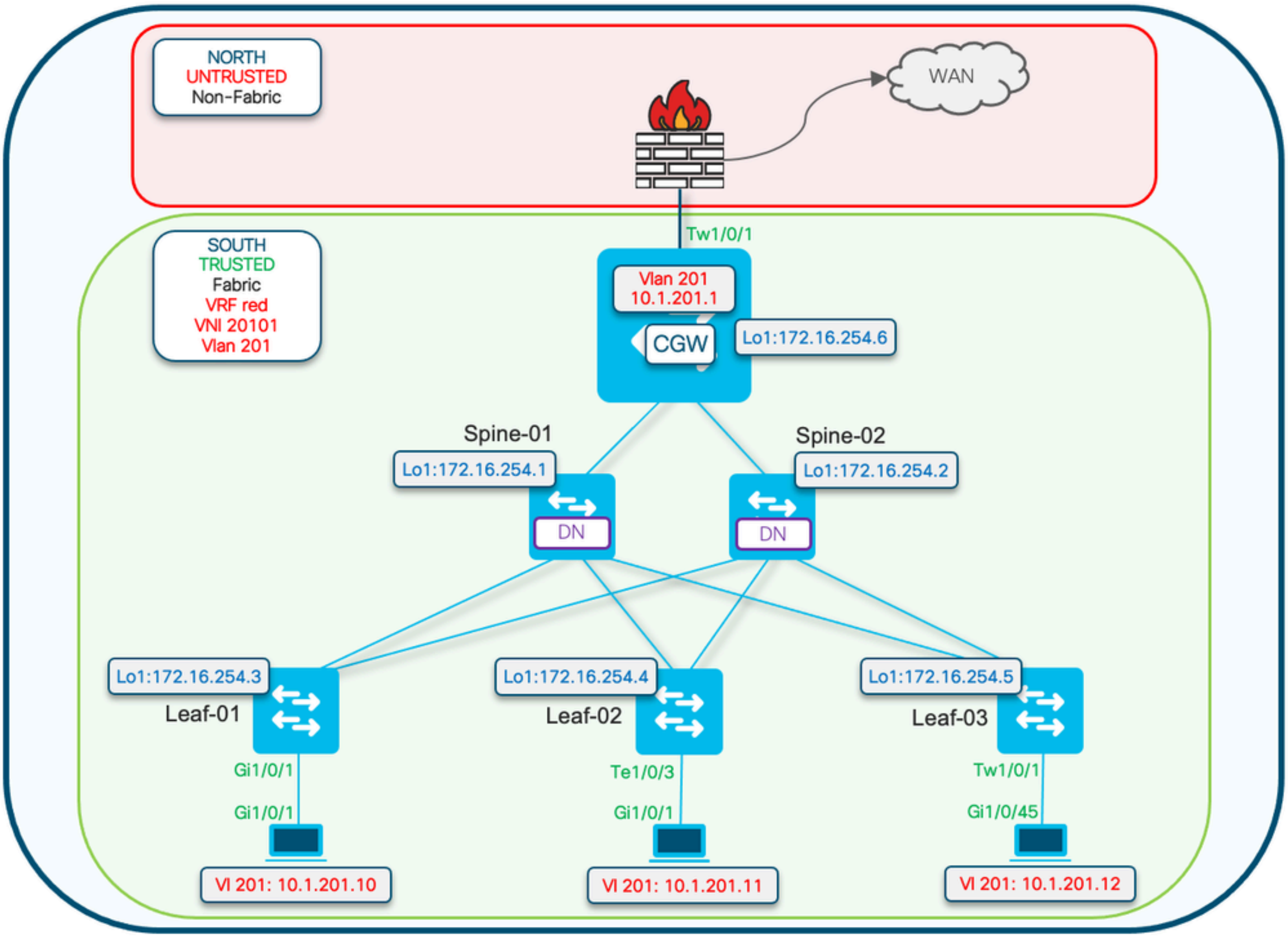
Terminology

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

Configure (Standard CGW Deployment)

Network Diagram



Note: This section covers a standard CGW deployment without the use of the protected feature.

- Debugs showing the DHCP DORA packet exchange are only shown in the Protected segment example

L2 VTEP (Leaf) Key Details

Request packet comes from client

- Use the Default gw advertised CGW mac.
- If more than one gw exists first gw mac would be used.
- Convert outer broadcast MAC (client initiated: D and R in DORA) to unicast GW mac and forward to CGW

DHCP snooping adds: option 82 suboptions: circuit and RID

(RID is used by response pkt processing on CGW)

(Informs CGW its not local and to fabric relay back to L2VTEP)

<#root>

```
Option: (82) Agent Information Option
        Length: 24
        Option 82 Suboption: (1) Agent Circuit ID
            Length: 12
            Agent Circuit ID: 010a0008000027750101 0000

        Option 82 Suboption: (2) Agent Remote ID

            Length: 8
          Agent Remote ID:
          000

682c7bf88700 <-- switch base mac 682c.7bf8.8700 (from 'show switch')
```

- Response pkts received from CGW over vxlan tunnel.
- Leaf Strips option 82.
- Adds binding entries with client source interface. (vxlan-mod-port gives the client source interface).
- Response packet forwarded to client.

L3 VTEP (CGW) Key Details

- Enable DHCP SNOOPING
- Enable DHCP RELAY in SVI
- Request is received from L2VTEP, and is given to relay.
- Relay adds other option 82 suboptions (gi, server override, and so on) and sends to DHCP server.
- DHCP response from dhcp server first comes to RELAY component.
- After RELAY strips off option 82 parameters (gi address, server override, and so on) the packet is passed to dhcp snooping component.
- Snooping component checks the RID (Router ID) and if its not local doesn’t remove option 82 subopton 1 and 2.
- Fabric relays (since RID is not local) packet is direct forwarded to remote client.
- Uses client Mac and does bridge inject. Hardware does client mac lookup and forwards the packet with vxlan encap to the originating L2VTEP.

L2VTEP

Configure the evpn instance

<#root>

```
Leaf-01#

show run | beg l2vpn evpn instance 201


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

Enable DHCP Snooping

<#root>

```
Leaf-01#

show run | sec dhcp snoop


ip dhcp snooping vlan 101,

201


ip dhcp snooping
```

CGW

Configure the evpn instance

<#root>

```
Border#

sh run | s l2vpn evpn instance 201


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

default-gateway advertise enable  <-- Enable to add BGP DEF GW ext. community attribute
```

Note: The DEF GW attribute is critical for L2 Relay to know who to encapsulate & send the DHCP packet to.

Enable DHCP snooping

<#root>

```
Border#

sh run | s dhcp snoop


ip dhcp snooping vlan 101,

201


ip dhcp snooping
```

Ensure the DHCP relay has the correct configuration to handle the additional options

<#root>

```
Border#

sh run int vl 201


Building configuration...

interface Vlan201
 mac-address 0000.beef.cafe
 vrf forwarding red

 ip dhcp relay information option vpn-id  <-- Ensure the vrf info is passed to the server




ip dhcp relay source-interface Loopback0 <-- Sets the relay source interface to the loopback


 ip address 10.1.201.1 255.255.255.0


ip helper-address global 10.1.33.33      <-- In this scenario the DHCP server is in the global routing t
```

Verify (Standard CGW Deployment)

Gateway Prefix (Leaf)

<#root>

```
Leaf-01#

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


BGP routing table entry for [2][172.16.255.3:201][0][48][0000BEEFCAFE][32][10.1.201.1]/24, version 8964
Paths: (1 available, best #1,

table evi_201

)

<-- In the EVI context for the segment


Not advertised to any peer
Refresh Epoch 3
Local, imported path from [2][172.16.255.6:201][0][48][0000BEEFCAFE][32][10.1.201.1]/24 (global)
  172.16.255.6 (metric 30) (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 ID


    Extended Community: RT:65001:201 ENCAP:8

EVPN DEF GW:0:0  <-- GW attribute added indicating this is GW prefix which L2 Relay uses
```

```
Originator: 172.16.255.6

, Cluster list: 172.16.255.1

<-- Learned from the Border (CGW)


    rx pathid: 0, tx pathid: 0x0
    Updated on Nov 14 2023 16:06:40 UTC
```

FED MATM (Leaf)

<#root>

```
Leaf-01#

show platform software fed switch active matm macTable vlan 201


VLAN   MAC                    Type  Seq#    EC_Bi  Flags  machandle           siHandle            riHandl

-----------------------------------------------------------------------------------------------------------
201    0006.f601.cd42         0x1   32436      0      0  0x71e058dc3368      0x71e058655018      0x0

201    0006.f601.cd01         0x1   32437      0      0  0x71e058dae308      0x71e058655018      0x0


201    0000.beef.cafe  0x5000001

       0      0     64  0x71e059177138      0x71e058eeb418      0x71e058df81f8      0x0

VTEP 172.16.255.6 adj_id 1371



 No



 <--- The GW MAC shows learnt via the Border Leaf Loopback with the right flags



Total Mac number of addresses:: 3
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:: 1          <---


*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                                <-- these 3 values added = 0x5000001 (not
```

Local MAC (Leaf)

<#root>

```
Leaf-01#

show switch


Switch/Stack Mac Address : 682c.7bf8.8700 - Local Mac Address
Mac persistency wait time: Indefinite
                                             H/W   Current
Switch#   Role    Mac Address     Priority Version  State
-------------------------------------------------------------------------------------
*1       Active

682c.7bf8.8700

     1      V01     Ready

<--- Use to validate the Agent ID in DHCP Option 82
```

DHCP Snooping (Leaf & CGW)

<#root>

```
Leaf-01#

show ip dhcp snooping



Switch DHCP snooping is enabled


Switch DHCP gleaning is disabled
DHCP snooping is configured on following VLANs:
101,201

DHCP snooping is operational on following VLANs:


101,201



Insertion of option 82 is enabled
circuit-id default format: vlan-mod-port
remote-id: 682c.7bf8.8700 (MAC)          <--- Leaf-01 adds the switch MAC to Option 82 to indicate to CG
```

```
CGW#

show ip dhcp snooping



Switch DHCP snooping is enabled


Switch DHCP gleaning is disabled
DHCP snooping is configured on following VLANs:
101,201

DHCP snooping is operational on following VLANs:


101,201
```

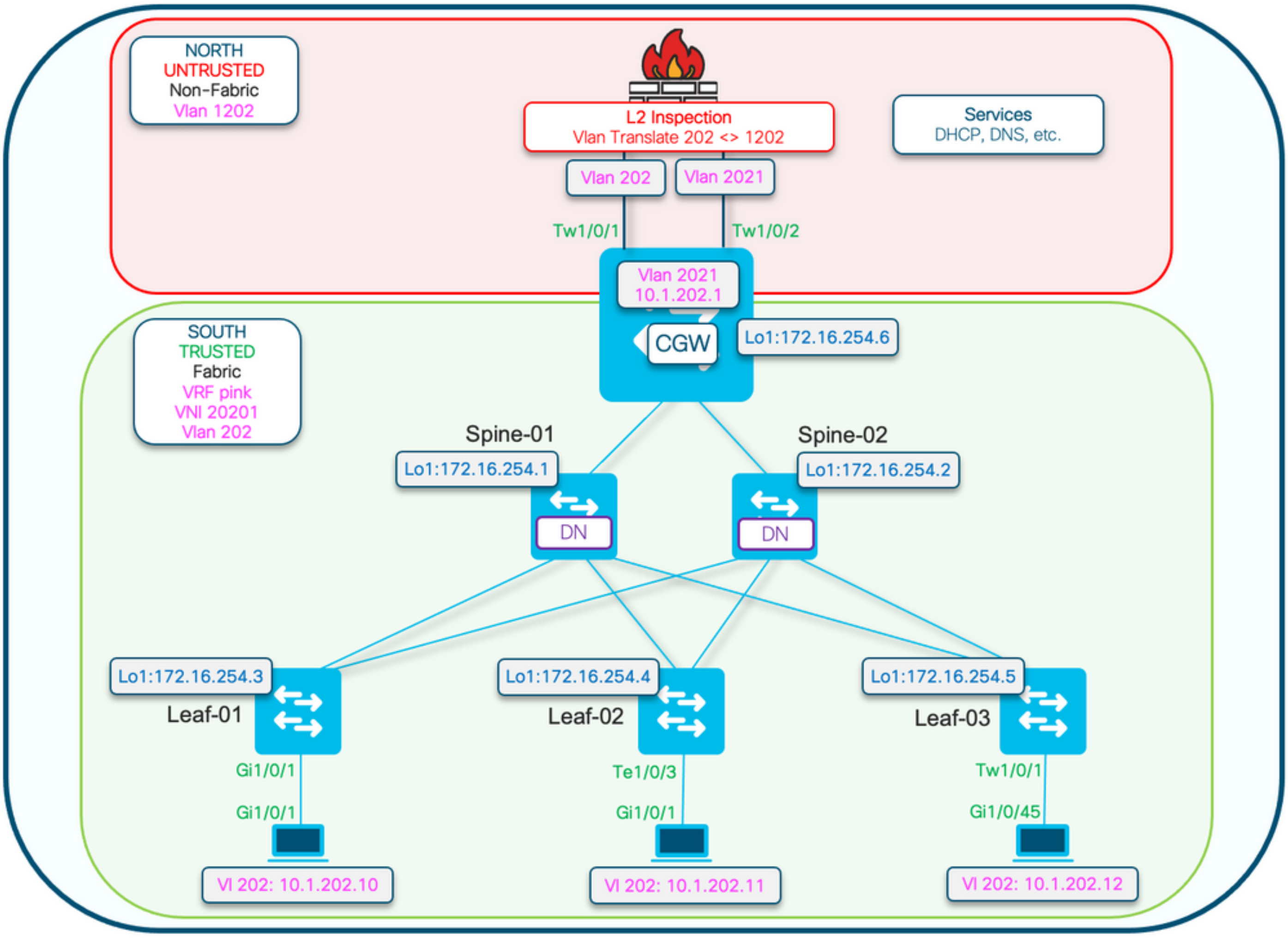
Configure (Partially Isolated Protected)

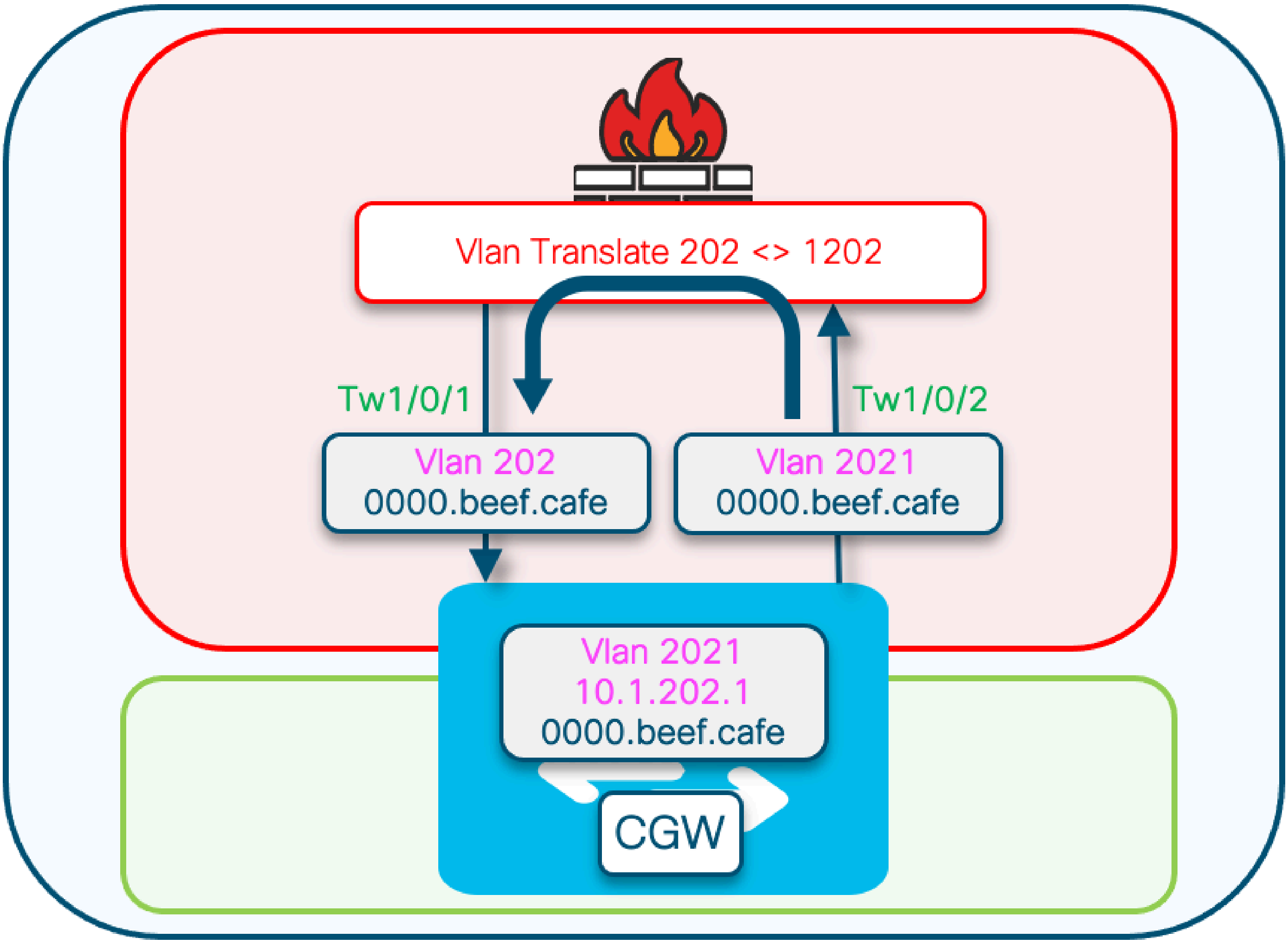
DHCP snooping on the Access Leaf relies on the default gateway route from CGW to learn the gateway MAC to forward DHCP packets to.

- When using the Partially Isolated design with external gateway additional configurations are required on CGW to advertise the MAC-IP RT2 with the default gateway (DEF GW) attribute.

Note: Note: This section also works to describe a Totally Isolated Protected Segment implementation, which also uses a GW that is advertised into the fabric (versus GW outside the fabric).

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



L2 VTEP (Leaf) Key Details

Request packet comes from client

- Use the Default gw advertised CGW mac.
- If more than one gw exists first gw mac would be used.
- Convert outer broadcast MAC (client initiated: D and R in DORA) to unicast GW mac and forward to CGW

DHCP snooping adds: option 82 suboptions: circuit and RID

(RID is used by response pkt processing on CGW)

(Informs CGW its not local and to fabric relay back to L2VTEP)

<#root>

```
Option: (82) Agent Information Option
        Length: 24
        Option 82 Suboption: (1) Agent Circuit ID
```

```
            Length: 12
            Agent Circuit ID: 010a00080000277501010000

        Option 82 Suboption: (2) Agent Remote ID

            Length: 8
          Agent Remote ID:
          000

682c7bf88700 <-- switch base mac 682c.7bf8.8700 (from 'show switch')
```

- Response pkts received from CGW over vxlan tunnel.
- Leaf Strips option 82.
- Adds binding entries with client source interface. (vxlan-mod-port gives the client source interface).
- Response packet forwarded to client.

L3 VTEP (CGW) Key Details

- Enable DHCP SNOOPING
- Enable DHCP RELAY in SVI
- Request is received from L2VTEP, and is given to relay.
- Relay adds other option 82 suboptions (gi, server override, and so on) and sends to DHCP server.
- DHCP response from dhcp server first comes to RELAY component.
- After RELAY strips off option 82 parameters (gi address, server override, and so on) the packet is passed to dhcp snooping component.
- Snooping component checks the RID (Router ID) and if its not local doesn’t remove option 82 subopton 1 and 2.
- Fabric relays (since RID is not local) packet is direct forwarded to remote client.
- Uses client Mac and does bridge inject. Hardware does client mac lookup and forwards the packet with vxlan encap to the originating L2VTEP.

Steps required to support DHCP L2 Relay:

1. **Enable** ip local learning
2. **Create** a policy with gleaning disabled
3. **Attach** to external gateway evi/vlans
4. **Add** static entries into device tracking table for external gateway mac-ip
5. **Create** BGP route map to match RT2 MAC-IP prefixes and set the default gateway extended community
6. **Apply** route-map to BGP Route Reflector neighbors

7. **Ensure** the DHCP relay has the correct configuration to handle the additional option

8. **Configure** DHCP Snooping on fabric vlan and the External GW vlan

Note: No configuration changes are required on the access Leafs to support DHCP L2 Relay with external gateway.

CGW

Enable ip local learning

```
<#root>

CGW#

show running-config | beg l2vpn evpn instance 202

l2vpn evpn instance 202 vlan-based
 encapsulation vxlan
 replication-type ingress
 ip local-learning enable
```

```
<-- to advertise RT-2 with default gateway EC, ip local-learning must be enabled on the CGW.
```

```
    Use additional device-tracking policy shown in the next output to prevent MAC-IP binding flapping wh
```

```
 multicast advertise enable
```

```
<--- There is no default-gateway advertise enable cli here, as the SVI (Vlan 2021) used by this segment
```

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 external gateway evi/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 external gateway 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 extended community

<#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 | sec router 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
```

Ensure the DHCP relay has the correct configuration to handle the additional options

<#root>

```
CGW#

show run int vl 2021

Building configuration...

Current configuration : 315 bytes
!
interface Vlan2021
 mac-address 0000.beef.cafe
 vrf forwarding pink

 ip dhcp relay information option vpn-id   <-- Ensure the vrf info is passed to the server
 ip dhcp relay source-interface Loopback0  <-- sets the relay source interface to the loopback

 ip address 10.1.202.1 255.255.255.0
 ip helper-address global 10.1.33.33       <-- In this scenario the next hop to the DHCP server is in th

 no ip redirects
 ip local-proxy-arp
 ip route-cache same-interface
 no autostate
```

Configure DHCP Snooping on fabric vlans and the External GW vlan

<#root>

```
Leaf01#

sh run | s dhcp snoop

ip dhcp snooping vlan 202
ip dhcp snooping

CGW#

sh run | s dhcp snoop

ip dhcp snooping vlan 202,2021  <-- snooping is required in both the fabric vlan and the external GW vla

ip dhcp snooping
```

Ensure that the uplink to the DHCP server is trusted on the CGW

<#root>

```
CGW#

sh run int tw 1/0/1


interface TwentyFiveGigE1/0/1
 switchport trunk allowed vlan 202
 switchport mode trunk

 ip dhcp snooping trust

end

CGW#

sh run int tw 1/0/2


interface TwentyFiveGigE1/0/2
 switchport trunk allowed vlan 33,2021
 switchport mode trunk

 ip dhcp snooping trust

end
```

Note: Due to the way the server is placed on the Firewall device trust was configured on both links facing this device. In the zoomed in diagram you can see that the Offer arrives at both Tw1/0/1 & Tw1/0/2 in this design.

Verify (Partially Isolated Protected)

Gateway Prefix (Leaf)

<#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 3411
Paths: (1 available, best #1, table evi_202)
  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
      Extended Community: RT:65001:202 ENCAP:8
```

EVPN DEF GW:0:0   <-- GW attribute added indicating this is GW prefix which L2 Relay uses

```
      Originator: 172.16.255.6, Cluster list: 172.16.255.1
      rx pathid: 0, tx pathid: 0x0
      Updated on Sep 19 2023 19:57:25 UTC
```

FED MATM (Leaf)

Confirm the Leaf has installed the CGW remote MAC in hardware

<#root>

```
Leaf01#
```

show platform software fed switch active matm macTable vlan 202

```
VLAN   MAC                    Type  Seq#    EC_Bi  Flags  machandle          siHandle            riHandl
----------------------------------------------------------------------------------------------------------
202    0006.f601.cd01          0x1    1093      0      0  0x71e05918f138      0x71e05917a1a8      0x0

202    0006.f601.cd44          0x1   14309      0      0  0x71e058cdc208      0x71e05917a1a8      0x0
```

202

0000.beef.cafe 0x5000001

```
       0      0     64  0x71e058ee5d88      0x71e059195f88      0x71e059171678      0x0
```

<--- The GW MAC shows learnt via the Border Leaf Loopback

```
Total Mac number of addresses:: 3
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:: 1
*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

```
<-- these 3 values added = 0x5000001 (note that 0x4000000 = GW type address
```

Local MAC (Leaf)

<#root>

```
Leaf01#

show switch

Switch/Stack Mac Address : 682c.7bf8.8700 - Local Mac Address
Mac persistency wait time: Indefinite
                                              H/W   Current
Switch#   Role    Mac Address     Priority Version  State
--------------------------------------------------------------------------------------
*1       Active

682c.7bf8.8700

     1      V01     Ready

<-- this is the MAC that will be added to DHCP Agent Remote ID
```

DHCP Snooping (Leaf & CGW)

Confirm that DHCP snooping is enabled on the Leaf in the fabric vlan

<#root>

```
Leaf01#

show ip dhcp snooping

Switch DHCP snooping is enabled
Switch DHCP gleaning is disabled
DHCP snooping is configured on following VLANs:
202

DHCP snooping is operational on following VLANs:   <-- Snooping on in the Fabric Vlan
202

<...snip...>


Insertion of option 82 is enabled
   circuit-id default format: vlan-mod-port
   remote-id: 682c.7bf8.8700 (MAC)            <--- Remote ID (RID) inserted by Leaf to DHCP packets

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

Confirm that DHCP snooping is enabled on the CGW in the fabric and external gateway vlans

<#root>

```
CGW#
```

```
show ip dhcp snooping

Switch DHCP snooping is enabled
Switch DHCP gleaning is disabled
DHCP snooping is configured on following VLANs:
202,2021

DHCP snooping is operational on following VLANs:  <-- Snooping on in the Fabric and External GW Vlans
202,2021

<...snip...>

DHCP snooping trust/rate is configured on the following Interfaces:

Interface                  Trusted    Allow option    Rate limit (pps)
-----------------------    -------    ------------    ----------------

TwentyFiveGigE1/0/1

             yes        yes             unlimited

<-- Trust set on ports the OFFER arrives on

Interface                  Trusted    Allow option    Rate limit (pps)
-----------------------    -------    ------------    ----------------
  Custom circuit-ids:

TwentyFiveGigE1/0/2

              yes        yes             unlimited

<-- Trust set on ports the OFFER arrives on

  Custom circuit-ids:
```

Confirm that DHCP snooping binding is created

<#root>

```
Leaf01#

show ip dhcp snooping binding

MacAddress



IpAddress

        Lease(sec)  Type           VLAN

Interface

------------------  ---------------  ----------  -------------  ----  --------------------
00:06:F6:01:CD:43

10.1.202.10

      34261       dhcp-snooping   202

GigabitEthernet1/0/1  <-- DHCP Snooping has created the binding

Total number of bindings: 1
```

Troubleshoot (Any CGW Type)

Debugs are helpful to show how the DHCP snooping and L2 Relay processes are handling DHCP packets.

Note: These debugs can be used for any type of deployment that uses CGW with DCHP L2 Relay.

DHCP Snooping Debugs (Leaf)

Debug Snooping to confirm packet processing

<#root>

```
Leaf01#

debug ip dhcp snooping packet

DHCP Snooping Packet debugging is on

Leaf01#
```

```
show debugging
```

```
DHCP Snooping packet debugging is on
```

Start the host DHCP address attempt

- For this document a shut/no shut of the SVI which is addressed via DHCP was performed to trigger the DORA exchange
- For windows host you can do a ipconfig /release > ipconfig /renew

Collect the debugs from show logging or from the terminal window

DHCP DISCOVER

Discover is seen coming from the host facing port

<#root>

```
*Sep 19 20:16:31.164:

DHCP_SNOOPING: received new DHCP packet from input interface (GigabitEthernet1/0/1)  <-- host facing por


*Sep 19 20:16:31.177:

DHCP_SNOOPING: process new DHCP packet, message type: DHCPDISCOVER, input interface: Gi1/0/1

, MAC da: ffff.ffff.ffff,

MAC sa: 0006.f601.cd43

, IP da: 255.255.255.255, IP sa: 0.0.0.0, DHCP ciaddr: 0.0.0.0, DHCP yiaddr: 0.0.0.0, DHCP siaddr: 0.0.
*Sep 19 20:16:31.177: DHCP_SNOOPING: add relay information option.
*Sep 19 20:16:31.177:

DHCP_SNOOPING: Encoding opt82 CID in vlan-mod-port format         <-- Option 82 encoding


*Sep 19 20:16:31.177: DHCP_SNOOPING:VxLAN : vlan_id 202 VNI 20201 mod 1 port 1
*Sep 19 20:16:31.177:

DHCP_SNOOPING: Encoding opt82 RID in MAC address format           <-- Encoding the switch Remote ID (loc

*Sep 19 20:16:31.177: DHCP_SNOOPING: binary dump of relay info option, length: 26 data:
0x52 0x18 0x1 0xC 0x1 0xA 0x0 0x8 0x0 0x0 0x4E 0xE9 0x1 0x1 0x0 0x0 0x2 0x8 0x0 0x6

0x68 0x2C 0x7B 0xF8 0x87 0x0  <-- the switch local MAC 682c.7bf8.8700

*Sep 19 20:16:31.177: DHCPS BRIDGE PAK: vlan=202 platform_flags=1
*Sep 19 20:16:31.177: DHCP_SNOOPING: bridge packet get invalid mat entry: FFFF.FFFF.FFFF, packet is flo
*Sep 19 20:16:31.177:

DHCP_SNOOPING: L2RELAY: sent unicast packet to default gw: 0000.beef.cafe vlan 0 src intf GigabitEtherne
```

DHCP OFFER

Offer is seen arriving from the fabric Tunnel interface

<#root>

```
*Sep 19 20:16:33.180:

DHCP_SNOOPING: received new DHCP packet from input interface (Tunnel0)

*Sep 19 20:16:33.194:

DHCP_SNOOPING: process new DHCP packet, message type: DHCPOFFER, input interface: Tu0, MAC da: 0006.f601

, IP da: 255.255.255.255, IP sa: 10.1.202.1, DHCP ciaddr: 0.0.0.0, DHCP yiaddr: 10.1.202.18, DHCP siadd
*Sep 19 20:16:33.194: DHCP_SNOOPING: binary dump of option 82, length: 26 data:
0x52 0x18 0x1 0xC 0x1 0xA 0x0 0x8 0x0 0x0 0x4E 0xE9 0x1 0x1 0x0 0x0 0x2 0x8 0x0 0x6 0x68 0x2C 0x7B 0xF8
*Sep 19 20:16:33.194: DHCP_SNOOPING: binary dump of extracted circuit id, length: 14 data:
0x1 0xC 0x1 0xA 0x0 0x8 0x0 0x0 0x4E 0xE9 0x1 0x1 0x0 0x0
*Sep 19 20:16:33.194: DHCP_SNOOPING: binary dump of extracted remote id, length: 10 data:
0x2 0x8 0x0 0x6

0x68 0x2C 0x7B 0xF8 0x87 0x0                                <-- the switch local MAC 682c.7bf8.8700

*Sep 19 20:16:33.194:  actual_fmt_cid OPT82_FMT_CID_VXLAN_MOD_PORT_INTF global_opt82_fmt_rid OPT82_FMT_
*Sep 19 20:16:33.194: dhcp_snooping_platform_is_local_dhcp_packet: VXLAN-MOD-PORT opt82 vni 20201, vlan
*Sep 19 20:16:33.194:

DHCP_SNOOPING: opt82 data indicates local packet      <-- switch found its own RID in Option 82 paramete


*Sep 19 20:16:33.194: DHCP_SNOOPING: remove relay information option.
*Sep 19 20:16:33.194: DHCP_SNOOPING opt82_fmt_cid_intf OPT82_FMT_CID_VXLAN_MOD_PORT_INTF opt82_fmt_cid_
*Sep 19 20:16:33.194:

DHCP_SNOOPING: VxLAN vlan_id 202 VNI 20201 mod 1 port 1

*Sep 19 20:16:33.194:

DHCP_SNOOPING: mod 1 port 1 idb Gi1/0/1 found for 0006.f601.cd43

*Sep 19 20:16:33.194: DHCP_SNOOPING: calling forward_dhcp_reply
*Sep 19 20:16:33.194: platform lookup dest vlan for input_if: Tunnel0, is  tunnel, if_output: NULL, if_
*Sep 19 20:16:33.194: DHCP_SNOOPING opt82_fmt_cid_intf OPT82_FMT_CID_VXLAN_MOD_PORT_INTF opt82_fmt_cid_
*Sep 19 20:16:33.194: DHCP_SNOOPING: VxLAN vlan_id 202 VNI 20201 mod 1 port 1
*Sep 19 20:16:33.194: DHCP_SNOOPING: mod 1 port 1 idb Gi1/0/1 found for 0006.f601.cd43
*Sep 19 20:16:33.194: DHCP_SNOOPING: vlan 202 after pvlan check
*Sep 19 20:16:33.207:

DHCP_SNOOPING: direct forward dhcp replyto output port: GigabitEthernet1/0/1.  <-- sending packet to hos
```

DHCP REQUEST

Request is seen from the host facing port

<#root>

```
*Sep 19 20:16:33.209:

DHCP_SNOOPING: received new DHCP packet from input interface (GigabitEthernet1/0/1)


*Sep 19 20:16:33.222:

DHCP_SNOOPING: process new DHCP packet, message type: DHCPREQUEST
```

```
, input interface: Gi1/0/1, MAC da: ffff.ffff.ffff, MAC sa: 0006.f601.cd43, IP da: 255.255.255.255, IP 
*Sep 19 20:16:33.222: DHCP_SNOOPING: add relay information option.
*Sep 19 20:16:33.222: DHCP_SNOOPING: Encoding opt82 CID in vlan-mod-port format
*Sep 19 20:16:33.222: DHCP_SNOOPING:VxLAN : vlan_id 202 VNI 20201 mod 1 port 1
*Sep 19 20:16:33.222: DHCP_SNOOPING: Encoding opt82 RID in MAC address format
*Sep 19 20:16:33.222: DHCP_SNOOPING: binary dump of relay info option, length: 26 data:
0x52 0x18 0x1 0xC 0x1 0xA 0x0 0x8 0x0 0x0 0x4E 0xE9 0x1 0x1 0x0 0x0 0x2 0x8 0x0 0x6 0x68 0x2C 0x7B 0xF8
*Sep 19 20:16:33.222: DHCPS BRIDGE PAK: vlan=202 platform_flags=1
*Sep 19 20:16:33.222: DHCP_SNOOPING: bridge packet get invalid mat entry: FFFF.FFFF.FFFF, packet is flo
*Sep 19 20:16:33.222:
```

DHCP_SNOOPING: L2RELAY: sent unicast packet to default gw: 0000.beef.cafe vlan 0 src intf GigabitEtherne

DHCP ACK

Ack is seen arriving from the fabric Tunnel interface

<#root>

```
*Sep 19 20:16:33.225:
```

DHCP_SNOOPING: received new DHCP packet from input interface (Tunnel0)

```
*Sep 19 20:16:33.238:
```

DHCP_SNOOPING: process new DHCP packet, message type: DHCPACK, input interface: Tu0, MAC da: 0006.f601.c

```
, IP da: 255.255.255.255, IP sa: 10.1.202.1, DHCP ciaddr: 0.0.0.0, DHCP yiaddr: 10.1.202.10, DHCP siadd
*Sep 19 20:16:33.238: DHCP_SNOOPING: binary dump of option 82, length: 26 data:
0x52 0x18 0x1 0xC 0x1 0xA 0x0 0x8 0x0 0x0 0x4E 0xE9 0x1 0x1 0x0 0x0 0x2 0x8 0x0 0x6 0x68 0x2C 0x7B 0xF8
*Sep 19 20:16:33.239: DHCP_SNOOPING: binary dump of extracted circuit id, length: 14 data:
0x1 0xC 0x1 0xA 0x0 0x8 0x0 0x0 0x4E 0xE9 0x1 0x1 0x0 0x0
*Sep 19 20:16:33.239: DHCP_SNOOPING: binary dump of extracted remote id, length: 10 data:
0x2 0x8 0x0 0x6 0x68 0x2C 0x7B 0xF8 0x87 0x0
*Sep 19 20:16:33.239:  actual_fmt_cid OPT82_FMT_CID_VXLAN_MOD_PORT_INTF global_opt82_fmt_rid OPT82_FMT_
*Sep 19 20:16:33.239: dhcp_snooping_platform_is_local_dhcp_packet: VXLAN-MOD-PORT opt82 vni 20201, vlan
*Sep 19 20:16:33.239:
```

DHCP_SNOOPING: opt82 data indicates local packet

```
*Sep 19 20:16:33.239:
```

dhcp_snooping_platform_is_local_dhcp_packet: VXLAN-MOD-PORT opt82 vni 20201, vlan_id 202

```
*Sep 19 20:16:33.239: DHCP_SNOOPING: opt82 data indicates local packet
*Sep 19 20:16:33.239: DHCP_SNOOPING opt82_fmt_cid_intf OPT82_FMT_CID_VXLAN_MOD_PORT_INTF opt82_fmt_cid_
*Sep 19 20:16:33.239: DHCP_SNOOPING: VxLAN vlan_id 202 VNI 20201 mod 1 port 1
*Sep 19 20:16:33.239:
```

DHCP_SNOOPING: mod 1 port 1 idb Gi1/0/1 found for 0006.f601.cd43

```
*Sep 19 20:16:33.239: DHCP_SNOOPING: Reroute dhcp pak, message type: DHCPACK
*Sep 19 20:16:33.239: DHCP_SNOOPING: remove relay information option.
*Sep 19 20:16:33.239: DHCP_SNOOPING opt82_fmt_cid_intf OPT82_FMT_CID_VXLAN_MOD_PORT_INTF opt82_fmt_cid_
*Sep 19 20:16:33.239: DHCP_SNOOPING: VxLAN vlan_id 202 VNI 20201 mod 1 port 1
*Sep 19 20:16:33.239: DHCP_SNOOPING: mod 1 port 1 idb Gi1/0/1 found for 0006.f601.cd43
*Sep 19 20:16:33.239: DHCP_SNOOPING: calling forward_dhcp_reply
*Sep 19 20:16:33.239: platform lookup dest vlan for input_if: Tunnel0, is  tunnel, if_output: NULL, if_
*Sep 19 20:16:33.239: DHCP_SNOOPING opt82_fmt_cid_intf OPT82_FMT_CID_VXLAN_MOD_PORT_INTF opt82_fmt_cid_
*Sep 19 20:16:33.239: DHCP_SNOOPING: VxLAN vlan_id 202 VNI 20201 mod 1 port 1
*Sep 19 20:16:33.239: DHCP_SNOOPING: mod 1 port 1 idb Gi1/0/1 found for 0006.f601.cd43
*Sep 19 20:16:33.239: DHCP_SNOOPING: vlan 202 after pvlan check
```

```
*Sep 19 20:16:33.252:

DHCP_SNOOPING: direct forward dhcp replyto output port: GigabitEthernet1/0/1.
```

Note: These debugs are snipped. They produce a memory dump of the packet, but annotation of this part of the debug result is outside the scope of this document.

DHCP Snooping Debugs (CGW)

DHCP DISCOVER

Because of how the packet is sent out and received back on the CGW (hairpinned at the firewall) the debugs fire twice

Arriving from fabric on Tunnel interface & sent out Tw 1/0/1 toward Firewall in Fabric vlan 202

<#root>

```
*Apr 16 14:37:43.890:
 DHCP_SNOOPING: received new DHCP packet from input interface (Tunnel0)  <-- Discover sent from Leaf01 a


*Apr 16 14:37:43.901: DHCP_SNOOPING: process new DHCP packet, message type: DHCPDISCOVER, input interfa
*Apr 16 14:37:43.901: DHCPS BRIDGE PAK: vlan=202 platform_flags=1
*Apr 16 14:37:43.901:

DHCP_SNOOPING: bridge packet send packet to port: TwentyFiveGigE1/0/1, pak_vlan 202. <-- Sent to Firewal
```

Arriving from Firewall on Tw 1/0/2 in Vlan 2021 to be sent to the SVI & helper to DHCP server

<#root>

```
*Apr 16 14:37:43.901:

DHCP_SNOOPING: received new DHCP packet from input interface (TwentyFiveGigE1/0/2) <-- Firewall sends di


*Apr 16 14:37:43.911: DHCP_SNOOPING: process new DHCP packet, message type: DHCPDISCOVER, input interfa
*Apr 16 14:37:43.911:

DHCPS BRIDGE PAK: vlan=2021 platform_flags=1      <-- Vlan discover seen is now 2021


*Apr 16 14:37:43.911:

DHCP_SNOOPING: Packet destined to SVI Mac:0000.beef.cafe


*Apr 16 14:37:43.911:

DHCP_SNOOPING: bridge packet send packet to cpu port: Vlan2021.  <-- Packet punted to CPU for handling b
```

DHCP OFFER

Arrives from DHCP server back to the SVI 2021 where the helper is configured and forwarded to firewall

<#root>

```
*Apr 16 14:37:45.913:

DHCP_SNOOPING: received new DHCP packet from input interface (Vlan2021)  <-- Arriving from the DHCP serv


*Apr 16 14:37:45.923:

DHCP_SNOOPING: process new DHCP packet, message type: DHCPOFFER, input interface: Vl2021

, MAC da: ffff.ffff.ffff, MAC sa: 0000.beef.cafe, IP da: 255.255.255.255, IP sa: 10.1.202.1, DHCP ciadd
*Apr 16 14:37:45.923: DHCP_SNOOPING: binary dump of option 82, length: 26 data:
0x52 0x18 0x1 0xC 0x1 0xA 0x0 0x8 0x0 0x0 0x4E 0xE9 0x1 0x1 0x0 0x0 0x2 0x8 0x0 0x6 0x68 0x2C 0x7B 0xF8
*Apr 16 14:37:45.924: DHCP_SNOOPING: binary dump of extracted circuit id, length: 14 data:
0x1 0xC 0x1 0xA 0x0 0x8 0x0 0x0 0x4E 0xE9 0x1 0x1 0x0 0x0
*Apr 16 14:37:45.924: DHCP_SNOOPING: binary dump of extracted remote id, length: 10 data:
0x2 0x8 0x0 0x6 0x68 0x2C 0x7B 0xF8 0x87 0x0
*Apr 16 14:37:45.924: actual_fmt_cid OPT82_FMT_CID_VXLAN_MOD_PORT_INTF global_opt82_fmt_rid OPT82_FMT_R
*Apr 16 14:37:45.924: dhcp_snooping_platform_is_local_dhcp_packet: VXLAN-MOD-PORT opt82 vni 20201, vlan
```

```
*Apr 16 14:37:45.924:

DHCP_SNOOPING: opt82 data indicates not a local packet


*Apr 16 14:37:45.924: DHCP_SNOOPING: can't parse option 82 data of the message,it is either in wrong fo

<-- This is expected even in working scenario (disregard it)


*Apr 16 14:37:45.924: DHCP_SNOOPING: calling forward_dhcp_reply
*Apr 16 14:37:45.924: platform lookup dest vlan for input_if: Vlan2021, is NOT tunnel, if_output: Vlan20
*Apr 16 14:37:45.924: DHCP_SNOOPING: vlan 2021 after pvlan check
*Apr 16 14:37:45.934:

DHCP_SNOOPING: direct forward dhcp replyto output port: TwentyFiveGigE1/0/2. <-- sending back toward the
```

Arrives from Firewall in Fabric vlan and sent from CGW into fabric toward Leaf

<#root>

```
*Apr 16 14:37:45.934:

DHCP_SNOOPING: received new DHCP packet from input interface (TwentyFiveGigE1/0/1)


*Apr 16 14:37:45.944:

DHCP_SNOOPING: process new DHCP packet, message type: DHCPOFFER, input interface: Twe1/0/1

, MAC da: ffff.ffff.ffff, MAC sa: 0000.beef.cafe, IP da: 255.255.255.255, IP sa: 10.1.202.1, DHCP ciadd
*Apr 16 14:37:45.944: DHCP_SNOOPING: binary dump of option 82, length: 26 data:
0x52 0x18 0x1 0xC 0x1 0xA 0x0 0x8 0x0 0x0 0x4E 0xE9 0x1 0x1 0x0 0x0 0x2 0x8 0x0 0x6 0x68 0x2C 0x7B 0xF8
*Apr 16 14:37:45.944: DHCP_SNOOPING: binary dump of extracted circuit id, length: 14 data:
0x1 0xC 0x1 0xA 0x0 0x8 0x0 0x0 0x4E 0xE9 0x1 0x1 0x0 0x0
*Apr 16 14:37:45.944: DHCP_SNOOPING: binary dump of extracted remote id, length: 10 data:
0x2 0x8 0x0 0x6 0x68 0x2C 0x7B 0xF8 0x87 0x0
*Apr 16 14:37:45.944: actual_fmt_cid OPT82_FMT_CID_VXLAN_MOD_PORT_INTF global_opt82_fmt_rid OPT82_FMT_R
*Apr 16 14:37:45.944: dhcp_snooping_platform_is_local_dhcp_packet: VXLAN-MOD-PORT opt82 vni 20201, vlan
*Apr 16 14:37:45.945:

DHCP_SNOOPING: opt82 data indicates not a local packet


*Apr 16 14:37:45.945: DHCP_SNOOPING: EVPN enabled Ex GW:fabric relay can't parse option 82 data of the 
*Apr 16 14:37:45.945: DHCP_SNOOPING: client address lookup failed to locate client interface, retry loo
*Apr 16 14:37:45.945: DHCP_SNOOPING: lookup packet destination port failed to get mat entry for mac: 00
*Apr 16 14:37:45.945:

DHCP_SNOOPING: L2RELAY: Ex GW unicast bridge packet to fabric: vlan id 202 from Twe1/0/1  <-- L2 RELAY f
```

DHCP REQUEST

<#root>

```
*Apr 16 14:37:45.967:

DHCP_SNOOPING: received new DHCP packet from input interface (Tunnel0)
```

```
*Apr 16 14:37:45.978:
```

DHCP_SNOOPING: process new DHCP packet, message type: DHCPREQUEST

```
, input interface: Tu0, MAC da: 0000.beef.cafe, MAC sa: 0006.f601.cd43, IP da: 255.255.255.255, IP sa: 
*Apr 16 14:37:45.978: DHCPS BRIDGE PAK: vlan=202 platform_flags=1
*Apr 16 14:37:45.978:
```

DHCP_SNOOPING: bridge packet send packet to port: TwentyFiveGigE1/0/1, pak_vlan 202. <-- Send toward Fir

<#root>

```
*Apr 16 14:37:45.978:
```

DHCP_SNOOPING: received new DHCP packet from input interface (TwentyFiveGigE1/0/2) <-- Receive from Fire

```
*Apr 16 14:37:45.989:
```

DHCP_SNOOPING: process new DHCP packet, message type: DHCPREQUEST

```
, input interface: Twe1/0/2, MAC da: 0000.beef.cafe, MAC sa: 0006.f601.cd43, IP da: 255.255.255.255, IP
*Apr 16 14:37:45.989: DHCPS BRIDGE PAK: vlan=2021 platform_flags=1
*Apr 16 14:37:45.989: DHCP_SNOOPING: Packet destined to SVI Mac:0000.beef.cafe
*Apr 16 14:37:45.989:
```

DHCP_SNOOPING: bridge packet send packet to cpu port: Vlan2021. <-- Punt to CPU / DHCP helper

DHCP ACK

<#root>

```
*Apr 16 14:37:45.990:
```

DHCP_SNOOPING: received new DHCP packet from input interface (Vlan2021) <-- Packet back to SVI from DHC

```
*Apr 16 14:37:46.000:
```

DHCP_SNOOPING: process new DHCP packet, message type: DHCPACK, input interface: Vl2021

```
, MAC da: ffff.ffff.ffff, MAC sa: 0000.beef.cafe, IP da: 255.255.255.255, IP sa: 10.1.202.1, DHCP ciadd
*Apr 16 14:37:46.000: DHCP_SNOOPING: binary dump of option 82, length: 26 data:
0x52 0x18 0x1 0xC 0x1 0xA 0x0 0x8 0x0 0x0 0x4E 0xE9 0x1 0x1 0x0 0x0 0x2 0x8 0x0 0x6 0x68 0x2C 0x7B 0xF8
*Apr 16 14:37:46.000: DHCP_SNOOPING: binary dump of extracted circuit id, length: 14 data:
0x1 0xC 0x1 0xA 0x0 0x8 0x0 0x0 0x4E 0xE9 0x1 0x1 0x0 0x0
*Apr 16 14:37:46.000: DHCP_SNOOPING: binary dump of extracted remote id, length: 10 data:
0x2 0x8 0x0 0x6 0x68 0x2C 0x7B 0xF8 0x87 0x0
*Apr 16 14:37:46.001: actual_fmt_cid OPT82_FMT_CID_VXLAN_MOD_PORT_INTF global_opt82_fmt_rid OPT82_FMT_R
*Apr 16 14:37:46.001: dhcp_snooping_platform_is_local_dhcp_packet: VXLAN-MOD-PORT opt82 vni 20201, vlan
*Apr 16 14:37:46.001:
```

DHCP_SNOOPING: opt82 data indicates not a local packet   <-- found this is coming from Leaf01 RID

```
*Apr 16 14:37:46.001: DHCP_SNOOPING: can't parse option 82 data of the message,it is either in wrong fo
```

```
*Apr 16 14:37:46.001: DHCP_SNOOPING: calling forward_dhcp_reply
*Apr 16 14:37:46.001: platform lookup dest vlan for input_if: Vlan2021, is NOT tunnel, if_output: Vlan2
*Apr 16 14:37:46.001: DHCP_SNOOPING: vlan 2021 after pvlan check
*Apr 16 14:37:46.011:

DHCP_SNOOPING: direct forward dhcp replyto output port: TwentyFiveGigE1/0/2.  <-- Send to Firewall
```

<#root>

```
*Apr 16 14:37:46.011:

DHCP_SNOOPING: received new DHCP packet from input interface (TwentyFiveGigE1/0/1)  <-- Coming back in f


*Apr 16 14:37:46.022:

DHCP_SNOOPING: process new DHCP packet, message type: DHCPACK, input interface: Twe1/0/1,

 MAC da: ffff.ffff.ffff, MAC sa: 0000.beef.cafe, IP da: 255.255.255.255, IP sa: 10.1.202.1, DHCP ciaddr
*Apr 16 14:37:46.022: DHCP_SNOOPING: binary dump of option 82, length: 26 data:
0x52 0x18 0x1 0xC 0x1 0xA 0x0 0x8 0x0 0x0 0x4E 0xE9 0x1 0x1 0x0 0x0 0x2 0x8 0x0 0x6 0x68 0x2C 0x7B 0xF8
*Apr 16 14:37:46.022: DHCP_SNOOPING: binary dump of extracted circuit id, length: 14 data:
0x1 0xC 0x1 0xA 0x0 0x8 0x0 0x0 0x4E 0xE9 0x1 0x1 0x0 0x0
*Apr 16 14:37:46.022: DHCP_SNOOPING: binary dump of extracted remote id, length: 10 data:
0x2 0x8 0x0 0x6 0x68 0x2C 0x7B 0xF8 0x87 0x0
*Apr 16 14:37:46.022: actual_fmt_cid OPT82_FMT_CID_VXLAN_MOD_PORT_INTF global_opt82_fmt_rid OPT82_FMT_R
*Apr 16 14:37:46.022: dhcp_snooping_platform_is_local_dhcp_packet: VXLAN-MOD-PORT opt82 vni 20201, vlan_
*Apr 16 14:37:46.022:

DHCP_SNOOPING: opt82 data indicates not a local packet


*Apr 16 14:37:46.022: DHCP_SNOOPING: EVPN enabled Ex GW:fabric relay can't parse option 82 data of the
*Apr 16 14:37:46.022: DHCP_SNOOPING: client address lookup failed to locate client interface, retry loo
*Apr 16 14:37:46.022: DHCP_SNOOPING: lookup packet destination port failed to get mat entry for mac: 00
*Apr 16 14:37:46.022: DHCP_SNOOPING: can't find client's destination port, packet is assumed to be not
*Apr 16 14:37:46.022: DHCP_SNOOPING: client address lookup failed to locate client interface, retry loo
*Apr 16 14:37:46.022: DHCP_SNOOPING: lookup packet destination port failed to get mat entry for mac: 00
*Apr 16 14:37:46.022:

DHCP_SNOOPING: L2RELAY: Ex GW unicast bridge packet to fabric: vlan id 202 from Twe1/0/1  <-- Send packe
```

Embedded Capture

Use EPC to confirm DHCP packet exchange & parameters are correct

- This is shown from the perspective of the CGW, but the process can be repeated on Leaf to verify the packet exchange
- This example shows the Discover as the process and analysis are the same for the other DHCP packets

Check the route to the Leaf Loopback

<#root>

```
CGW#

show ip route 172.16.254.3

Routing entry for 172.16.254.3/32
  Known via "ospf 1", distance 110, metric 3, type intra area
  Last update from 172.16.1.25 on TwentyFiveGigE1/0/47, 2w6d ago
  Routing Descriptor Blocks:
  * 172.16.1.29, from 172.16.255.3, 2w6d ago,

via TwentyFiveGigE1/0/48

      Route metric is 3, traffic share count is 1
    172.16.1.25, from 172.16.255.3, 2w6d ago,

via TwentyFiveGigE1/0/47

      Route metric is 3, traffic share count is 1
```

Configure capture to run on links facing the Leaf01

```
   monitor capture 1 interface TwentyFiveGigE1/0/47 BOTH
   monitor capture 1 interface TwentyFiveGigE1/0/48 BOTH
   monitor capture 1 match any
   monitor capture 1 buffer size 100
   monitor capture 1 limit pps 1000
```

Start the capture, trigger your host to request a DHCP IP address, stop the capture

<#root>

```
monitor capture 1 start
  (have the host request dhcp ip)
monitor capture 1 stop
```

View the capture result starting with the DHCP Discover (Pay attention to the Transaction ID to confirm this is all the same DORA event)

<#root>

```
CGW#

show monitor cap 1 buff brief | i DHCP



 16

  12.737135      0.0.0.0 -> 255.255.255.255 DHCP 434

DHCP Discover

 -

Transaction ID 0x78b <-- Discover starts at Frame 16 with all same transaction ID
```

```
   18  14.740041   10.1.202.1 -> 255.255.255.255 DHCP 438 DHCP

Offer

    - Transaction ID

0x78b

   19  14.742741      0.0.0.0 -> 255.255.255.255 DHCP 452 DHCP

Request

  - Transaction ID

0x78b

   20  14.745646   10.1.202.1 -> 255.255.255.255 DHCP 438 DHCP

ACK

      - Transaction ID

0x78b
```

<#root>

```
CGW#

sh mon cap 1 buff detailed | b Frame 16

Frame 16:

 434 bytes on wire (3472 bits), 434 bytes captured (3472 bits) on interface /tmp/epc_ws/wif_to_ts_pipe,
    [Protocols in frame: eth:ethertype:ip:udp:vxlan:eth:ethertype:ip:udp:dhcp]
Ethernet II,

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

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

Dst: 10:f9:20:2e:9f:82

 (10:f9:20:2e:9f:82)

<-- Underlay Interface MACs

    Type: IPv4 (0x0800)
Internet Protocol Version 4,

Src: 172.16.254.3, Dst: 172.16.254.6

User Datagram Protocol, Src Port: 65281,

Dst Port: 4789                  <-- VXLAN Port

Virtual eXtensible Local Area Network
    VXLAN Network Identifier

(VNI): 20201                                <-- Correct VNI / Segment

    Reserved: 0
Ethernet II,

Src: 00:06:f6:01:cd:43

 (00:06:f6:01:cd:43),

Dst: 00:00:be:ef:ca:fe
```

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

<-- Inner Packet destined to CGW MAC


Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
User Datagram Protocol,

Src Port: 68, Dst Port: 67                       <-- DHCP ports


Dynamic Host Configuration Protocol (Discover)                           <-- DHCP Discover Packet


 Client MAC address: 00:06:f6:01:cd:43

 (00:06:f6:01:cd:43)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP


 Option: (53) DHCP Message Type (Discover)

        Length: 1
        DHCP: Discover (1)
    Option: (57) Maximum DHCP Message Size
        Length: 2
        Maximum DHCP Message Size: 1152
    Option: (61) Client identifier
        Length: 27
        Type: 0
        Client Identifier: cisco-0006.f601.cd43-Vl202
    Option: (12) Host Name
        Length: 17


Host Name: 9300-HOST-3750X-2

    Option: (55) Parameter Request List
        Length: 8
        Parameter Request List Item: (1) Subnet Mask
        Parameter Request List Item: (6) Domain Name Server
        Parameter Request List Item: (15) Domain Name
        Parameter Request List Item: (44) NetBIOS over TCP/IP Name Server
        Parameter Request List Item: (3) Router
        Parameter Request List Item: (33) Static Route
        Parameter Request List Item: (150) TFTP Server Address
        Parameter Request List Item: (43) Vendor-Specific Information
    Option: (60) Vendor class identifier
        Length: 8
        Vendor class identifier: ciscopnp


Option: (82) Agent Information Option

        Length: 24
        Option 82 Suboption: (1) Agent Circuit ID
            Length: 12
            Agent Circuit ID: 010a000800004ee901010000


Option 82 Suboption: (2) Agent Remote ID
```

```
            Length: 8

Agent Remote ID:

000

6682c7bf88700  <-- switch base mac 682c.7bf8.8700 (from 'show switch')

        Option: (255) End
            Option End: 255
```

Note: Capture tool can be used on any Leafs or CGW to determine the last point a part of the DHCP DORA exchange is suspected to be failing.

Verify snooping statistics for errors

<#root>

```
Leaf01#
```

```
show ip dhcp snooping statistics detail

 Packets Processed by DHCP Snooping                    = 1288


Packets Dropped Because

   IDB not known                                        = 0
   Queue full                                           = 0
   Interface is in errdisabled                          = 0
   Rate limit exceeded                                  = 0
   Received on untrusted ports                          = 0
   Nonzero giaddr                                       = 0
   Source mac not equal to chaddr                       = 0
   No binding entry                                     = 0
   Insertion of opt82 fail                              = 0
   Unknown packet                                       = 0
   Interface Down                                       = 0
   Unknown output interface                             = 0
   Misdirected Packets                                  = 0
   Packets with Invalid Size                            = 0
   Packets with Invalid Option                          = 0

<-- Look for any drop counter that is actively incrementing when the issue is seen.
```

Verify punt path for DHCP Snooping

- CoPP is the main component that drops packets in the punt path

<#root>

```
Leaf01#

show platform hardware switch active qos queue stats internal cpu policer


                         CPU Queue Statistics
============================================================================================
                                              (default) (set)     Queue        Queue
QId

PlcIdx

  Queue Name                Enabled   Rate     Rate      Drop(Bytes)

Drop(Frames)

--------------------------------------------------------------------------------------------
17

6



DHCP Snooping

               Yes     400       400      0

 0
```

```
              CPU Queue Policer Statistics
====================================================================

Policer

    Policer Accept   Policer Accept  Policer Drop  Policer Drop


Index

         Bytes          Frames        Bytes          Frames
-------------------------------------------------------------------
6          472723           1288           0             0
```

Another very helpful command to locate where a possible packet flood is occurring is 'show platform software fed switch active punt rates interfaces"

- This is very helpful to find a source interface where flooding is happening which is congesting the punt path and affecting legitimate CPU bound traffic

<#root>

```
Leaf01#

show platform software fed switch active punt rates interfaces

Punt Rate on Interfaces Statistics

Packets per second averaged over 10 seconds, 1 min and 5 mins
===========================================================================================
                             |           |  Recv |  Recv |  Recv |  Drop |  Drop   |  Drop

<-- Receive and drop rates for this port

 Interface Name              |  IF_ID    |  10s  |  1min |  5min |  10s  |  1min   |  5min
===========================================================================================


GigabitEthernet1/0/1          0x0000000a

       2       2       2       0       0       0

<-- the port and its IF-ID which can be used in the next command


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

<#root>

```
Leaf01#

show platform software fed switch active punt rates interfaces 0xa  <-- From previous command (omit the

Punt Rate on Single Interfaces Statistics


Interface : GigabitEthernet1/0/1 [if_id: 0xA]


  Received                         Dropped
  --------                         -------
```

```
   Total          : 8032546         Total           : 0
   10 sec average : 2               10 sec average  : 0
    1 min average : 2                1 min average  : 0
    5 min average : 2                5 min average  : 0


Per CPUQ punt stats on the interface

 (rate averaged over 10s interval)
==========================================================================
 Q  |         Queue                   |  Recv    |  Recv   |  Drop   |  Drop   |
 no |         Name                    |  Total   |  Rate   |  Total  |  Rate   |
==========================================================================
 17

 CPU_Q_DHCP_SNOOPING

                 1216        0        0        0
<...snip...>
```

DHCP Snooping Client Stats

Observe the DHCP message exchange using this command. This can be run on both Leaf or CGW to see the event trace

<#root>

```
Leaf01#

show platform dhcpsnooping client stats 0006.F601.CD43

DHCPSN: DHCP snooping server
DHCPD: DHCP protocol daemen
L2FWD: Transmit Packet to driver in L2 format
FWD: Transmit Packet to driver

(B): Dhcp message's response expected as 'B'roadcast
(U): Dhcp message's response expected as 'U'nicast

Packet Trace for client MAC 0006.F601.CD43:

Timestamp                Destination MAC  Destination Ip  VLAN  Message      Handler:Action
------------------------ ---------------- --------------- ----  ------------ ----------------
2023/09/28 14:53:59.866  FFFF.FFFF.FFFF   255.255.255.255 202  DHCPDISCOVER(B)  PUNT:RECEIVED
2023/09/28 14:53:59.866  FFFF.FFFF.FFFF   255.255.255.255 202  DHCPDISCOVER(B)  PUNT:TO_DHCPSN
2023/09/28 14:53:59.867  FFFF.FFFF.FFFF   255.255.255.255 202  DHCPDISCOVER(B)  BRIDGE:RECEIVED
2023/09/28 14:53:59.867  0000.BEEF.CAFE   255.255.255.255 202  DHCPDISCOVER(B)  L2INJECT:TO_FWD
2023/09/28 14:53:59.867  FFFF.FFFF.FFFF   255.255.255.255 202  DHCPDISCOVER(B)  BRIDGE:TO_INJECT
2023/09/28 14:53:59.867  FFFF.FFFF.FFFF   255.255.255.255 202  DHCPDISCOVER(B)  L2INJECT:TO_FWD
2023/09/28 14:54:01.871  0006.F601.CD43   255.255.255.255 202  DHCPOFFER(B)     PUNT:RECEIVED
2023/09/28 14:54:01.871  0006.F601.CD43   255.255.255.255 202  DHCPOFFER(B)     PUNT:TO_DHCPSN
2023/09/28 14:54:01.874  FFFF.FFFF.FFFF   255.255.255.255 202  DHCPREQUEST(B)   PUNT:RECEIVED
2023/09/28 14:54:01.874  FFFF.FFFF.FFFF   255.255.255.255 202  DHCPREQUEST(B)   PUNT:TO_DHCPSN
2023/09/28 14:54:01.874  FFFF.FFFF.FFFF   255.255.255.255 202  DHCPREQUEST(B)   BRIDGE:RECEIVED
2023/09/28 14:54:01.874  0000.BEEF.CAFE   255.255.255.255 202  DHCPREQUEST(B)   L2INJECT:TO_FWD
2023/09/28 14:54:01.874  FFFF.FFFF.FFFF   255.255.255.255 202  DHCPREQUEST(B)   BRIDGE:TO_INJECT
2023/09/28 14:54:01.874  FFFF.FFFF.FFFF   255.255.255.255 202  DHCPREQUEST(B)   L2INJECT:TO_FWD
2023/09/28 14:54:01.877  0006.F601.CD43   255.255.255.255 202  DHCPACK(B)       PUNT:RECEIVED
2023/09/28 14:54:01.877  0006.F601.CD43   255.255.255.255 202  DHCPACK(B)       PUNT:TO_DHCPSN
```

Additional Debugs

```
debug ip dhcp server packet detail
debug ip dhcp server packet
debug ip dhcp server events
debug ip dhcp snooping packet
debug dhcp detail
```

⚠

Caution: Use caution when running debugs!

Related Information

- Implement BGP EVPN Routing Policy on Catalyst 9000 Series Switches
- Implement BGP EVPN Protected Overlay Segmentation on Catalyst 9000 Series Switches
- Operate and Troubleshoot DHCP Snooping on Catalyst 9000 Switches