Catalyst 9000 Series 스위치에서 BGP EVPN DHCP Layer 2 Relay 구현

목차

소개

이 문서에서는 EVPN VxLAN DHCP L2 릴레이 기능을 구성, 확인 및 트러블슈팅하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

- 이 기능은 DHCP가 사용되는 모든 CGW 유형 구축에서 사용됩니다
- 보호되는 세그멘테이션을 구현할 경우 다음 문서를 검토하십시오.
  - Catalyst 9000 Series 스위치에 BGP EVPN 라우팅 정책 구현
  - Catalyst 9000 Series 스위치에서 BGP EVPN Protected Overlay Segmentation 구현

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- Catalyst 9300
- Catalyst 9400
- Catalyst 9500
- Catalyst 9600
- Cisco IOS® XE 17.12.1 이상 버전

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

문서 세부 정보

이 문서는 SVI가 없는 Leaf에서 중앙 게이트웨이로 DHCP를 릴레이해야 하는 모든 CGW 구축에 사용할 수 있습니다.

- 보호된 세그멘테이션을 사용하지 않는 경우 SVI가 패브릭에 보급되는 문서의 섹션을 사용합니다

보호되는 세그멘테이션을 구현하는 경우 이 문서는 3개의 상호 관련 문서 중 2번째 부분입니다.

- 문서 1: Catalyst 9000 Series 스위치에 BGP EVPN 라우팅 정책 구현은 오버레이에서 BGP BUM 트래픽을 제어하는 방법을 다루며, 먼저 구성해야 합니다
- 문서 2:Catalyst 9000 Series 스위치에서 BGP EVPN Protected Overlay Segmentation을 구현합니다. 문서 1의 오버레이 설계 및 정책을 기반으로 하며 'protected' 키워드의 구현에 대해 설명합니다.
- 문서 3: 이 문서. 마지막 두 문서 위에 빌드하고 레이어 2 전용 Leaf 및 CGW로 DHCP 릴레이

를 구현하는 방법을 설명합니다.

L2 릴레이 동작

릴레이	스누핑	코어 플러드	액세스 플러드	IPv4
예	예	아니요	예	• 옵션 82 하위 옵션: (1) 에이전트 회로 ID(vni-mod-port)가 dhcp 스누핑으로 채워집니다. • 하나는 dhcp trust 컨피그레이션으로 액세스 측면을 제한할 수 있습니다 * 권장 모델
예	아니요	예	예	• 옵션 82 하위 옵션: (1) 에이전트 회로 ID(vlan-mod-port)가 dhcp 스누핑으로 채워집니다.
아니요	예	아니요	예	• 옵션 82 하위 옵션: (1) 에이전트 회로 ID(vni-mod-port)가 dhcp 스누핑으로 채워집니다. • 하나는 dhcp trust 컨피그레이션으로 액세스 측면을 제한할 수 있습니다
릴레이	스누핑	코어 플러드	액세스 플러드	IPv6
예	예	예	예	• 옵션 82 하위 옵션: (1) 에이전트 회로 ID(vni-mod-port)가 dhcp 스누핑으로 채워집니다. • 하나는 dhcp trust 컨피그레이션으로 액세스 측면을 제한할 수 있습니다
예	아니요	예	예	• 옵션 82 하위 옵션: (1) 에이전트 회로 ID(vlan-mod-port)가 dhcp 스누핑으로 채워집니다.
아니요	예	예	예	• 옵션 82 하위 옵션: (1) 에이전트 회로 ID(vni-mod-port)가 dhcp 스누핑으로 채워집니다. • 하나는 dhcp trust 컨피그레이션으로 액세스 측면을 제한할 수 있습니다
아니요	아니요	예	예	

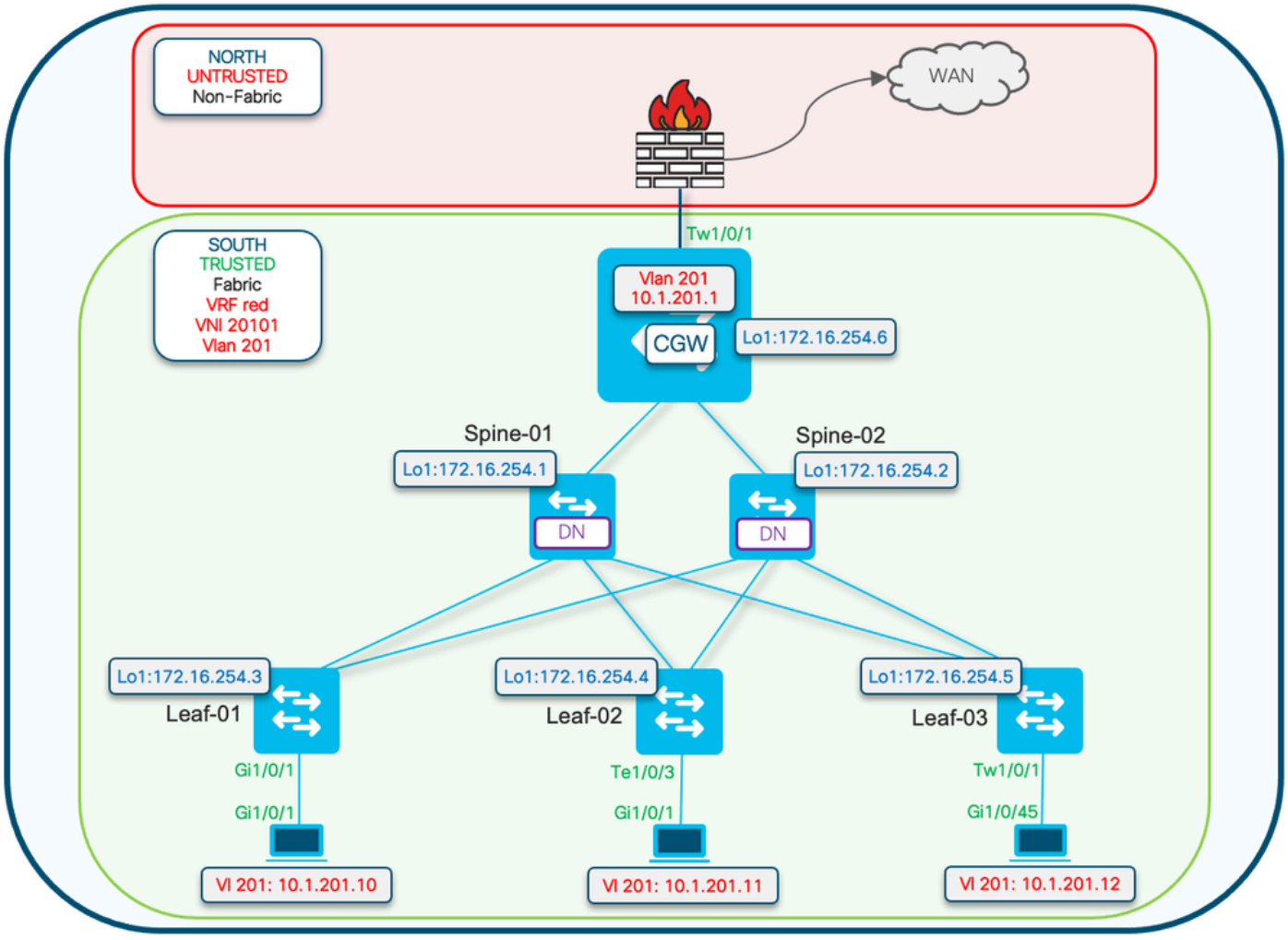
용어

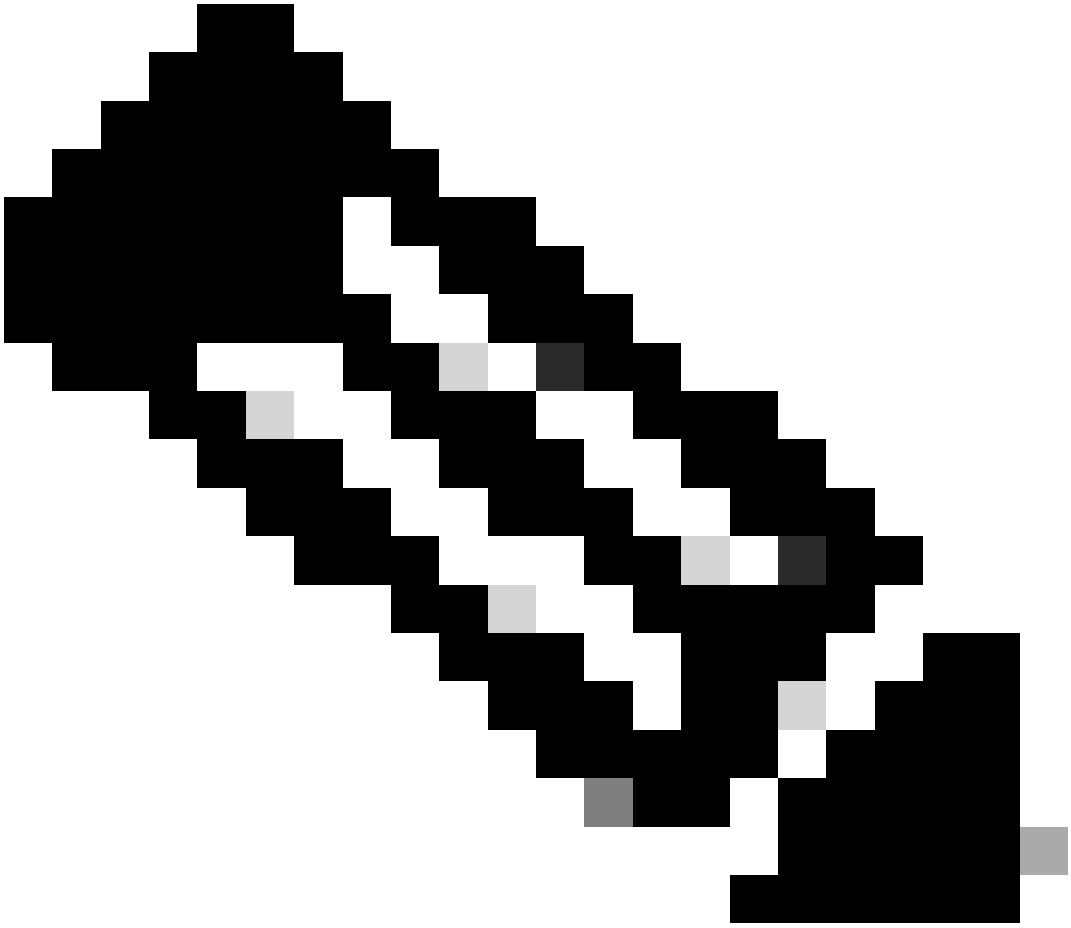
VRF	가상 라우팅 전달	다른 VRF 및 전역 IPv4/IPv6 라우팅 도메인과 구분되는 레이어 3 라우팅 도메인을 정의합니다.
AF	주소군	어떤 유형 접두사 및 라우팅 정보 BGP가 처리되는지 정의합니다.
AS	자동 시스템	단일 엔터티 또는 조직에서 모두 관리, 제어 및 감독하는 네트워크 또는 네트워크 컬렉션에 속하는 인터넷 라우팅 가능 IP 접두사 집합입니다
EVPN	이더넷 가상 사설망	BGP가 레이어 2 MAC 및 레이어 3 IP 정보를 전송할 수 있도록 하는 확장은 EVPN이며, VXLAN 오버레이 네트워크와 관련된 연결 정보를 배포할 프로토콜로 MP-BGP(Multi-Protocol Border Gateway Protocol)를 사용합니다.
VXLAN	가상 확장 LAN(Local Area Network)	VXLAN은 VLAN과 STP의 내재적 한계를 극복하기 위해 설계되었습니다. 이는 VLAN과 동일한 이더넷 레이어 2 네트워크 서비스를 제공하되 더 높은 유연성을 제공하는 IETF 표준[RFC 7348]입니다. 기능적으로 레이어 3 언더레이 네트워크에서 가상 오버레이로 실행되는 MAC-in-UDP 캡슐화 프로토콜입니다.
CGW	중앙 집중식 게이트웨이	게이트웨이 SVI가 각 leaf에 없는 EVPN 구현 대신 모든 라우팅은 비대칭 IRB(Integrated Routing and Bridging)를 사용하여 특정 리프에 의해 수행됩니다
데프 GW	기본 게이트웨이	'l2vpn evpn' 컨피그레이션 섹션 아래에서 "default-gateway advertise enable" 명령을 통해 MAC/IP 접두사에 추가된 BGP 확장 커뮤니티 특성입니다.
IMET(RT3)	포괄적 멀티캐스트 이더넷 태그(경로)	BGP type-3 경로라고도 합니다. 이 경로 유형은 VTEP 간에 BUM(브로드캐스트/알 수 없는 유니캐스트/멀티캐스트) 트래픽을 전달하는 데 EVPN에서 사용됩니다.
RT2	경로 유형 2	호스트 MAC 또는 게이트웨이 MAC-IP를 나타내는 BGP MAC 또는 MAC/IP 접두사
EVPN 관리자	EVPN 관리자	기타 다양한 구성 요소를 위한 중앙 관리 구성 요소(예: SISF에서 학습하고 L2RIB에 신호 전달)

SISF	스위치 통합 보안 기능	EVPN에서 Leaf에 어떤 로컬 호스트가 있는지 학습하는 데 사용되는 비종속적 호스트 추적 테이블
L2RIB	레이어 2 라우팅 정보 베이스	BGP, EVPN Mgr, L2FIB 간의 상호 작용을 관리하는 중간 구성 요소
연방	포워딩 엔진 드라이버	ASIC(하드웨어) 레이어 프로그래밍
매트	Mac 주소 테이블 관리자	IOS MATM: 로컬 주소만 설치하고 FED MATM: 컨트롤 플레인에서 학습한 로컬 및 원격 주소를 설치하고 하드웨어 포워딩 플레인의 일부인 하드웨어 테이블

구성(표준 CGW 구축)

네트워크 다이어그램





참고: 이 섹션에서는 보호 기능을 사용하지 않는 표준 CGW 구축에 대해 다룹니다.

- DHCP DORA 패킷 교환을 표시하는 디버깅은 Protected 세그먼트 예에만 표시됩니다

L2 VTEP(Leaf) 키 세부사항

요청 패킷이 클라이언트에서 옵니다.

- Default gw advertised CGW mac을 사용합니다.
- 1 gw가 둘 이상인 경우 첫 번째 gw mac가 사용됩니다.
- 외부 브로드캐스트 MAC(클라이언트가 시작한 DORA의 D 및 R)를 유니캐스트 GW MAC으로 전환하고 CGW로 전달

DHCP snooping 추가: 옵션 82 하위 옵션: 회선 및 RID

(RID는 CGW에서 응답 pkt 처리에 사용됩니다)

(CGW에 로컬이 아니며 패브릭 릴레이에 다시 L2VTEP로 알립니다)

<#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')
```

- vxlan 터널을 통해 CGW에서 수신된 응답 패킷입니다.
- 리프 스트립 옵션 82.
- 클라이언트 소스 인터페이스를 사용하여 바인딩 항목을 추가합니다. (vxlan-mod-port는 클라이언트 소스 인터페이스를 제공합니다.)
- 클라이언트에 전달된 응답 패킷입니다.

L3 VTEP(CGW) 주요 세부 정보

- DHCP 스누핑 활성화
- SVI에서 DHCP 릴레이 활성화
- 요청은 L2VTEP에서 수신되며 릴레이에 주어집니다.
- 릴레이는 다른 옵션 82 하위 옵션(gi, 서버 재정의 등)을 추가하여 DHCP 서버에 보냅니다.
- dhcp 서버의 DHCP 응답은 먼저 RELAY 구성 요소에 옵니다.
- RELAY가 옵션 82 매개 변수(gi 주소, 서버 재정의 등)를 제거한 후 패킷이 dhcp 스누핑 구성 요소에 전달됩니다.
- 스누핑 구성 요소는 RID(라우터 ID)를 확인하고 로컬이 아니면 옵션 82 하위 버튼 1과 2를 제거하지 않습니다.
- 패브릭 릴레이(RID가 로컬이 아니므로) 패킷이 원격 클라이언트에 직접 전달됩니다.

- 클라이언트 Mac을 사용하고 브리지 삽입을 수행합니다. 하드웨어는 클라이언트 mac 조회를 수행하고 vxlan encap이 포함된 패킷을 원래 L2VTEP로 전달합니다.

L2VTEP

evpn 인스턴스 구성

<#root>

```
Leaf-01#
show run | beg l2vpn evpn instance 201


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

DHCP 스누핑 활성화

<#root>

```
Leaf-01#
show run | sec dhcp snoop


ip dhcp snooping vlan 101,
201


ip dhcp snooping
```

CGW

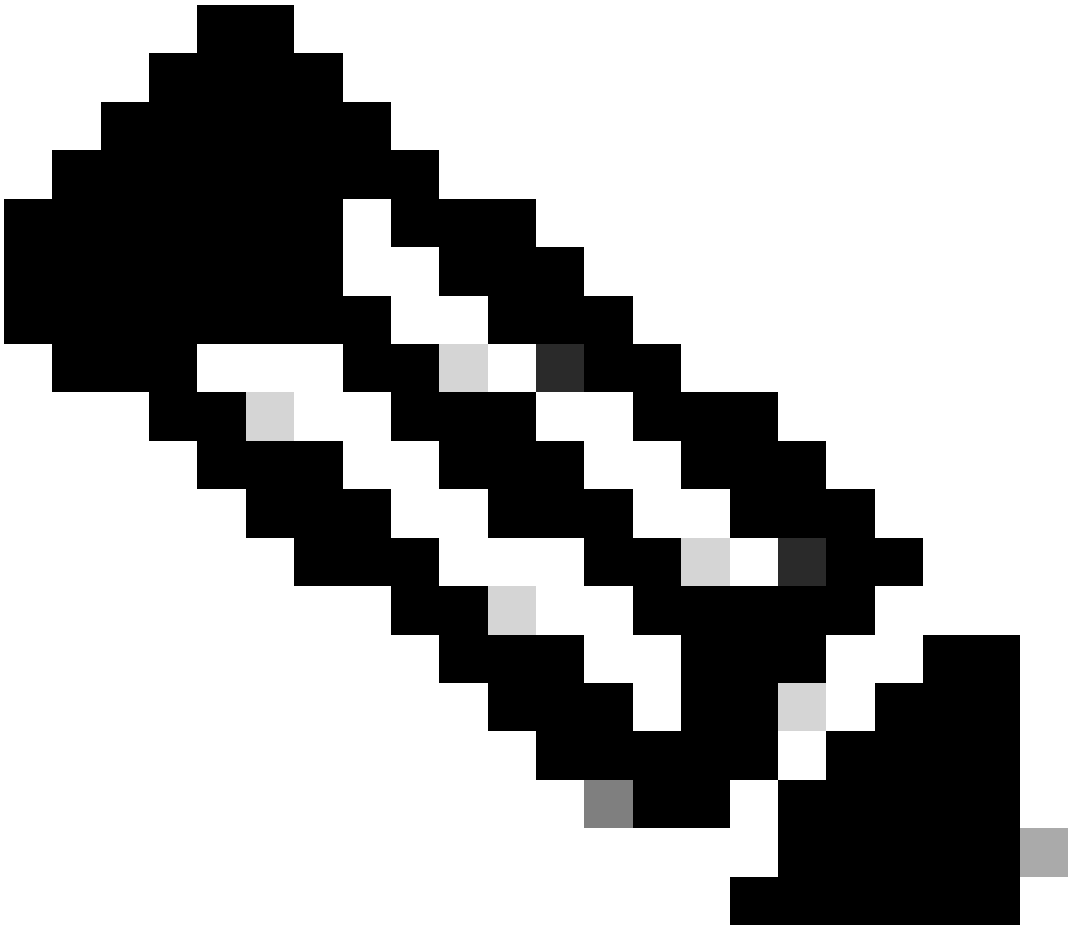
evpn 인스턴스 구성

<#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

---



참고: DEF GW 특성은 L2 릴레이가 DHCP 패킷을 캡슐화하고 전송할 사용자를 파악하는 데 중요합니다.

DHCP 스누핑 활성화

<#root>

Border#

sh run | s dhcp snoop

ip dhcp snooping vlan 101,

201

```
ip dhcp snooping
```

DHCP 릴레이가 추가 옵션을 처리할 수 있는 올바른 컨피그레이션을 가지고 있는지 확인합니다

```
<#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
```

확인(표준 CGW 구축)

게이트웨이 접두사(리프)

```
<#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(리프)

<#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
```

로컬 MAC(리프)

<#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 스누핑(리프 및 CGW)

<#root>

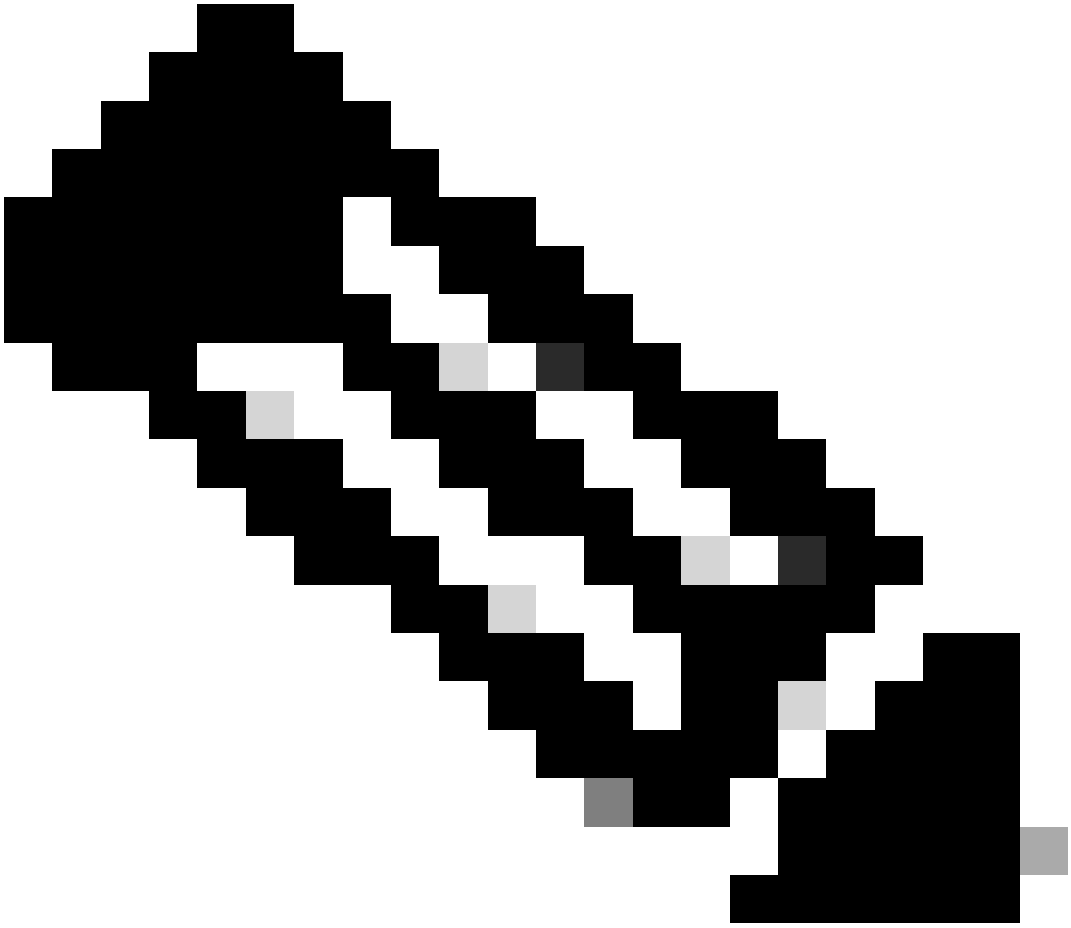
```
Leaf-01#

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
```

구성(부분적으로 격리된 보호)

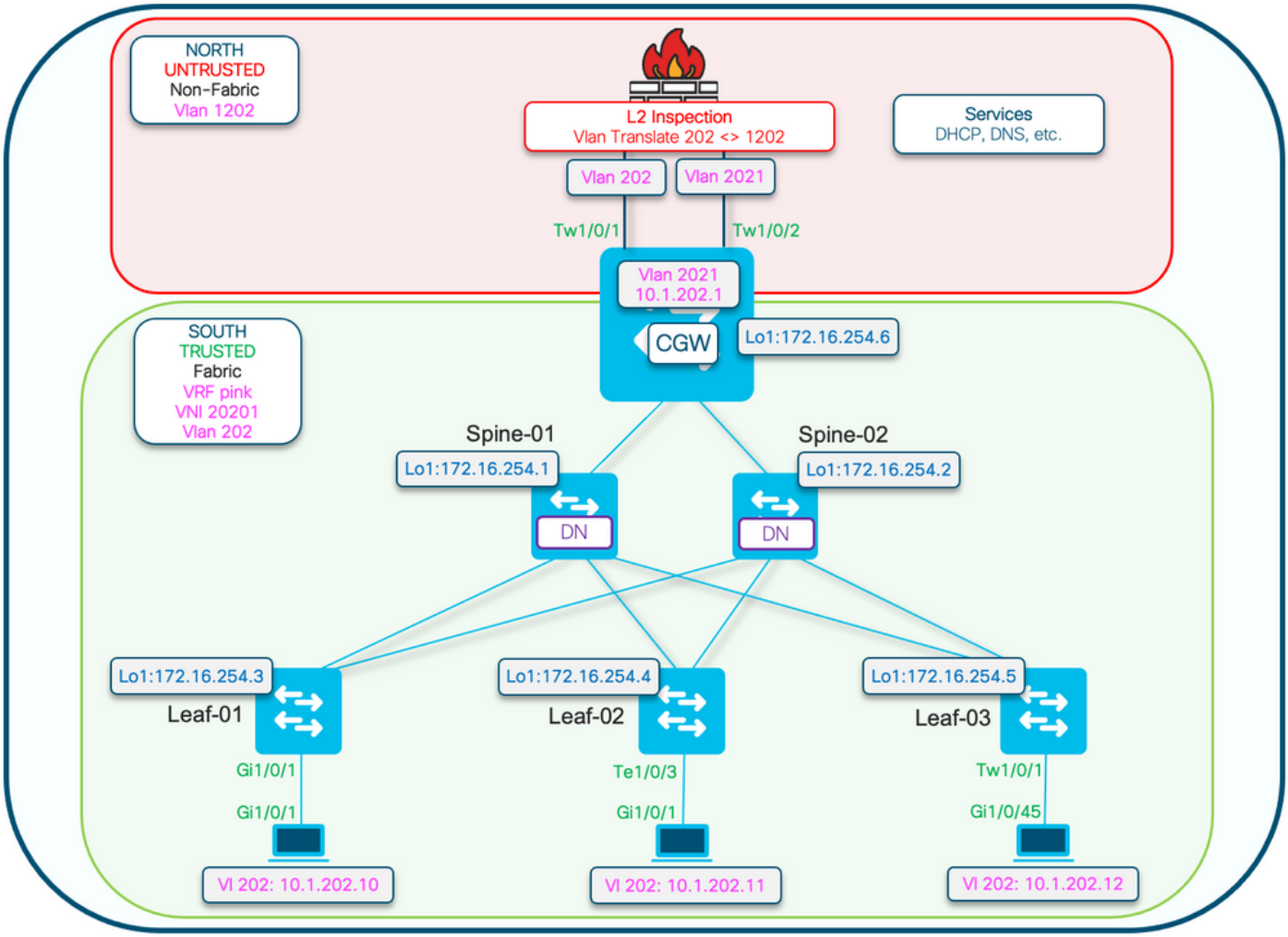
액세스 리프의 DHCP 스누핑은 CGW의 기본 게이트웨이 경로를 사용하여 DHCP 패킷을 전달할 게이트웨이 MAC를 학습합니다.

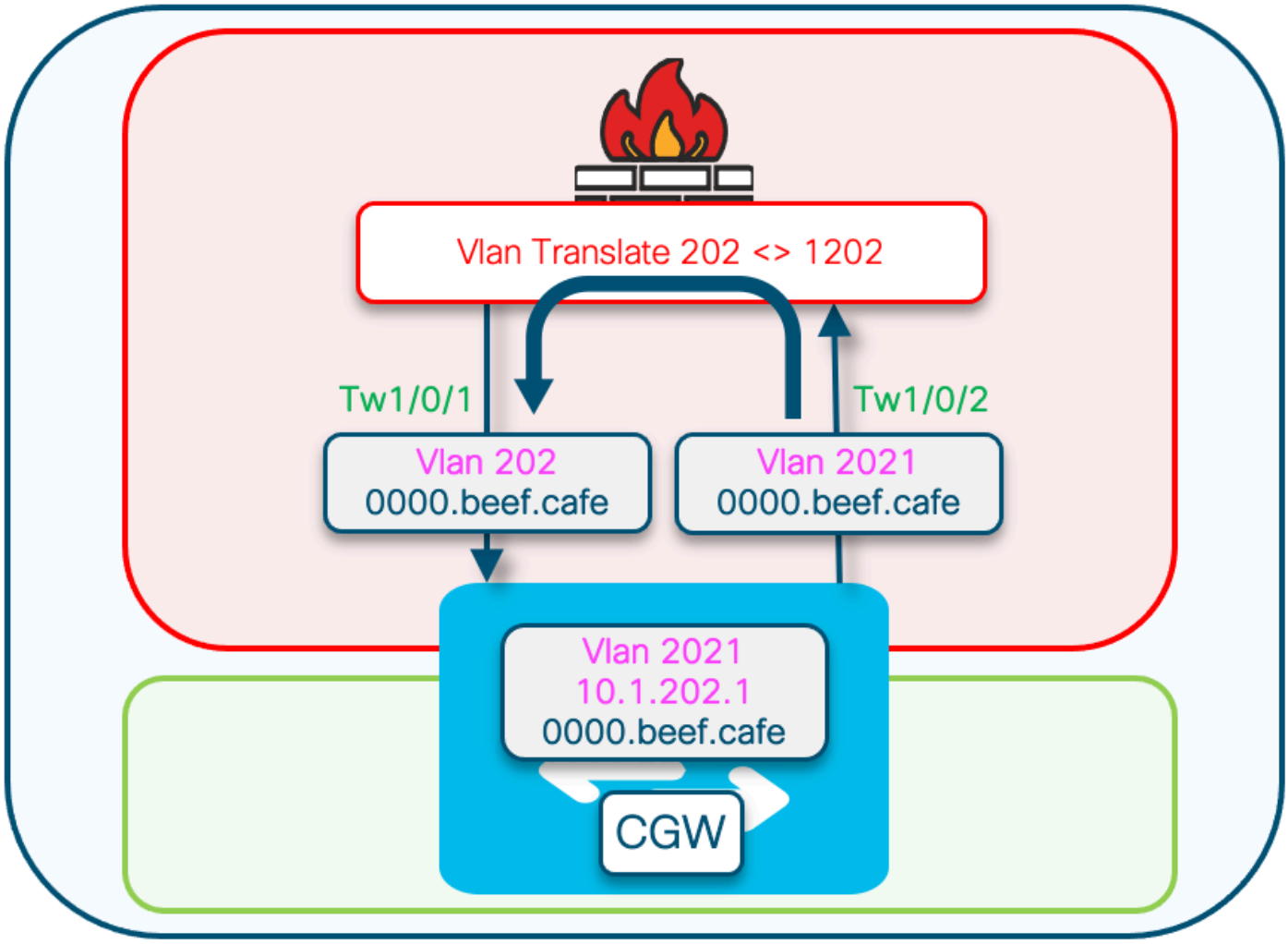
- 외부 게이트웨이와 함께 Partially Isolated 설계를 사용할 경우 MAC-IP RT2를 DEF GW(기본 게이트웨이) 특성으로 광고하려면 CGW에서 추가 컨피그레이션이 필요합니다.



참고: 참고: 이 섹션에서는 Totally Isolated Protected Segment 구현에 대해 설명합니다. 이 섹션에서는 패브릭에 광고되는 GW를 사용합니다(패브릭 외부의 GW와 다름).

네트워크 다이어그램


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) 키 세부사항

요청 패킷이 클라이언트에서 옵니다.

- Default gw advertised CGW mac을 사용합니다.
- 1 gw가 둘 이상인 경우 첫 번째 gw mac가 사용됩니다.
- 외부 브로드캐스트 MAC(클라이언트가 시작한 DORA의 D 및 R)를 유니캐스트 GW MAC으로 전환하고 CGW로 전달

DHCP snooping 추가: 옵션 82 하위 옵션: 회선 및 RID

(RID는 CGW에서 응답 pkt 처리에 사용됩니다)

(CGW에 로컬이 아니며 패브릭 릴레이에 다시 L2VTEP로 알립니다)

<#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')
```

- vxlan 터널을 통해 CGW에서 수신된 응답 패킷입니다.
- 리프 스트립 옵션 82.
- 클라이언트 소스 인터페이스를 사용하여 바인딩 항목을 추가합니다. (vxlan-mod-port는 클라이언트 소스 인터페이스를 제공합니다.)
- 클라이언트에 전달된 응답 패킷입니다.

L3 VTEP(CGW) 주요 세부 정보

- DHCP 스누핑 활성화
- SVI에서 DHCP 릴레이 활성화
- 요청은 L2VTEP에서 수신되며 릴레이에 주어집니다.
- 릴레이는 다른 옵션 82 하위 옵션(gi, 서버 재정의 등)을 추가하여 DHCP 서버에 보냅니다.
- dhcp 서버의 DHCP 응답은 먼저 RELAY 구성 요소에 옵니다.
- RELAY가 옵션 82 매개 변수(gi 주소, 서버 재정의 등)를 제거한 후 패킷이 dhcp 스누핑 구성 요소에 전달됩니다.
- 스누핑 구성 요소는 RID(라우터 ID)를 확인하고 로컬이 아니면 옵션 82 하위 버튼 1과 2를 제거하지 않습니다.
- 패브릭 릴레이(RID가 로컬이 아니므로) 패킷이 원격 클라이언트에 직접 전달됩니다.
- 클라이언트 Mac을 사용하고 브리지 삽입을 수행합니다. 하드웨어는 클라이언트 mac 조회를 수행하고 vxlan encap이 포함된 패킷을 원래 L2VTEP로 전달합니다.

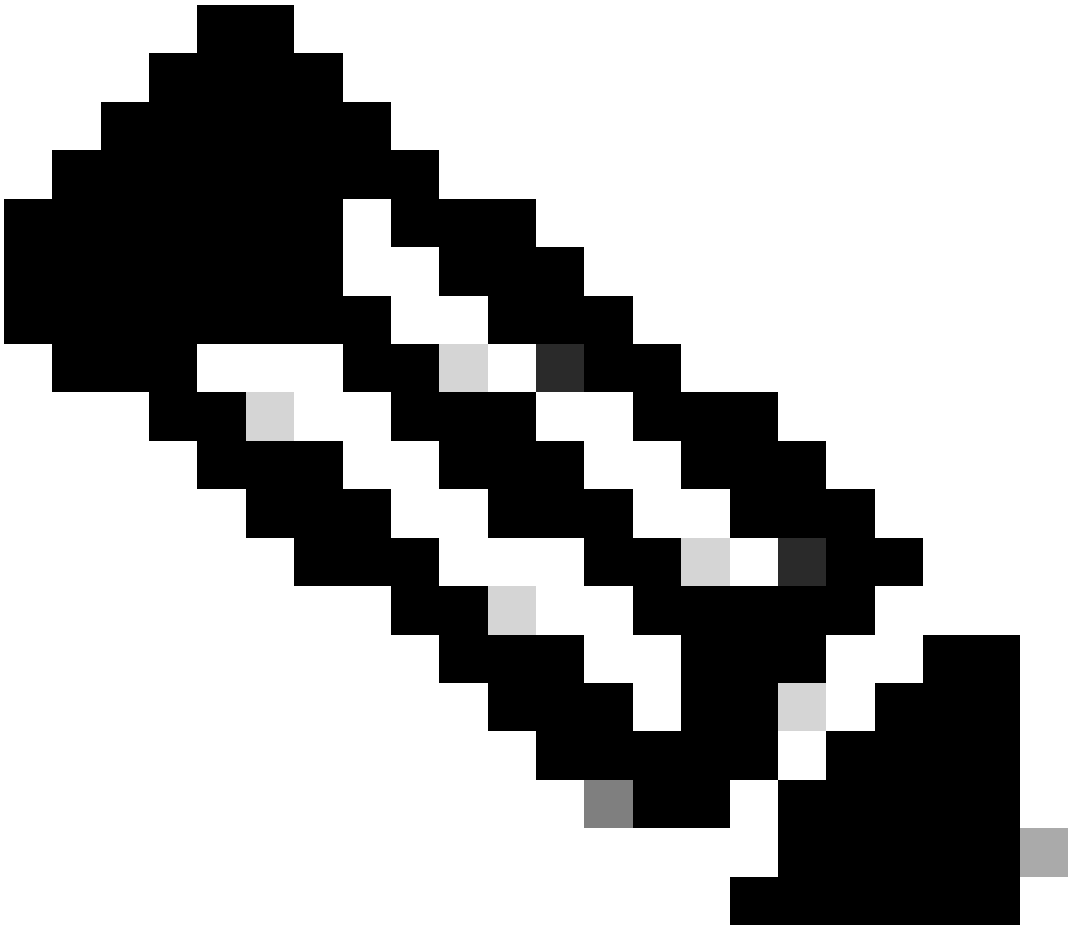
DHCP L2 릴레이 지원에 필요한 단계:

1. ip 로컬 학습 활성화

2. 청소가 비활성화된 정책 생성
3. 외부 게이트웨이 evi/vlan에 연결
4. 외부 게이트웨이 mac-ip에 대한 디바이스 추적 테이블에 고정 항목 추이
5. RT2 MAC-IP 접두사를 일치시키고 기본 게이트웨이 확장 커뮤니티를 설정하기 위한 BGP 경로 맵 생성
6. BGP 경로 리플렉터 네이버에 경로 지도 적용

7. DHCP 릴레이가 추가 옵션을 처리할 수 있는 올바른 컨피그레이션을 가지고 있는지 확인합니다

8. 패브릭 VLAN 및 외부 GW VLAN에 DHCP 스누핑 구성



참고: 외부 게이트웨이와의 DHCP L2 릴레이를 지원하기 위해 액세스 Leaf에서 컨피그레이션을 변경할 필요는 없습니다.

CGW

ip 로컬 학습 활성화

<#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
```

청소가 비활성화된 정책 생성

<#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
```

외부 게이트웨이 evi/vlan에 연결

<#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
```

외부 게이트웨이 mac-ip에 대한 디바이스 추적 테이블에 고정 항목 추가

<#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

RT2 MAC-IP 접두사를 일치시키고 기본 게이트웨이 확장 커뮤니티를 설정하기 위한 BGP 경로 맵 생성

<#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
```

BGP 경로 리플렉터 네이버에 경로 지도 적용

<#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
```

DHCP 릴레이가 추가 옵션을 처리할 수 있는 올바른 컨피그레이션을 가지고 있는지 확인합니다

<#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
```

패브릭 VLAN 및 외부 GW VLAN에 DHCP 스누핑 구성

<#root>

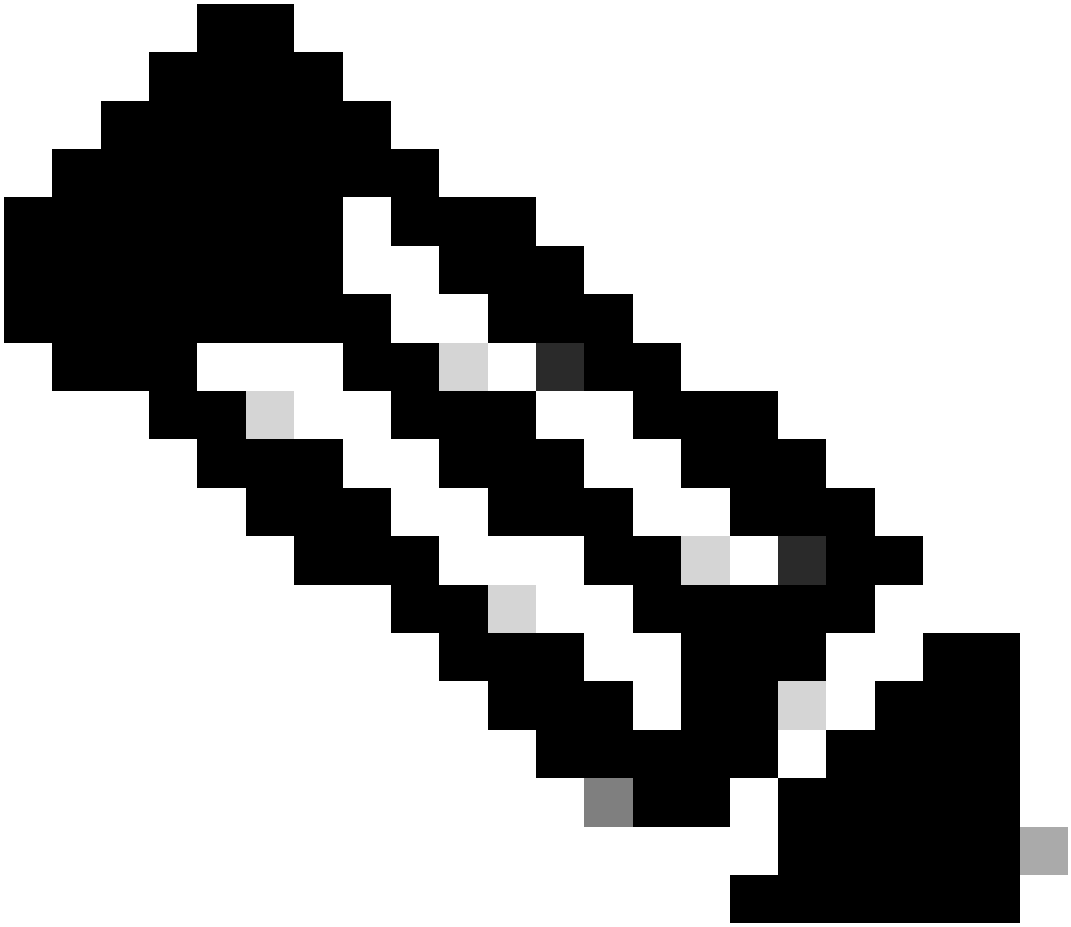
```
Leaf01#

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
```

DHCP 서버에 대한 업링크가 CGW에서 신뢰되는지 확인합니다.

<#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
```



참고: 서버가 방화벽 디바이스 트러스트에 배치되는 방식 때문에 이 디바이스를 향하는 두 링크에 모두 구성되었습니다. 확대 다이어그램에서는 이 설계에서 오퍼가 Tw1/0/1 및 Tw1/0/2에 모두 도착함을 확인할 수 있습니다.

확인(부분적으로 격리됨 보호)

게이트웨이 접두사(리프)

<#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(리프)

Leaf가 하드웨어에 CGW 원격 MAC을 설치했는지 확인합니다.

<#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
```

로컬 MAC(리프)

<#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 스누핑(리프 및 CGW)

패브릭 VLAN의 Leaf에서 DHCP 스누핑이 활성화되었는지 확인합니다

<#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...>
```

DHCP 스누핑이 패브릭 및 외부 게이트웨이 VLAN의 CGW에서 활성화되었는지 확인합니다.

```
<#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:
```

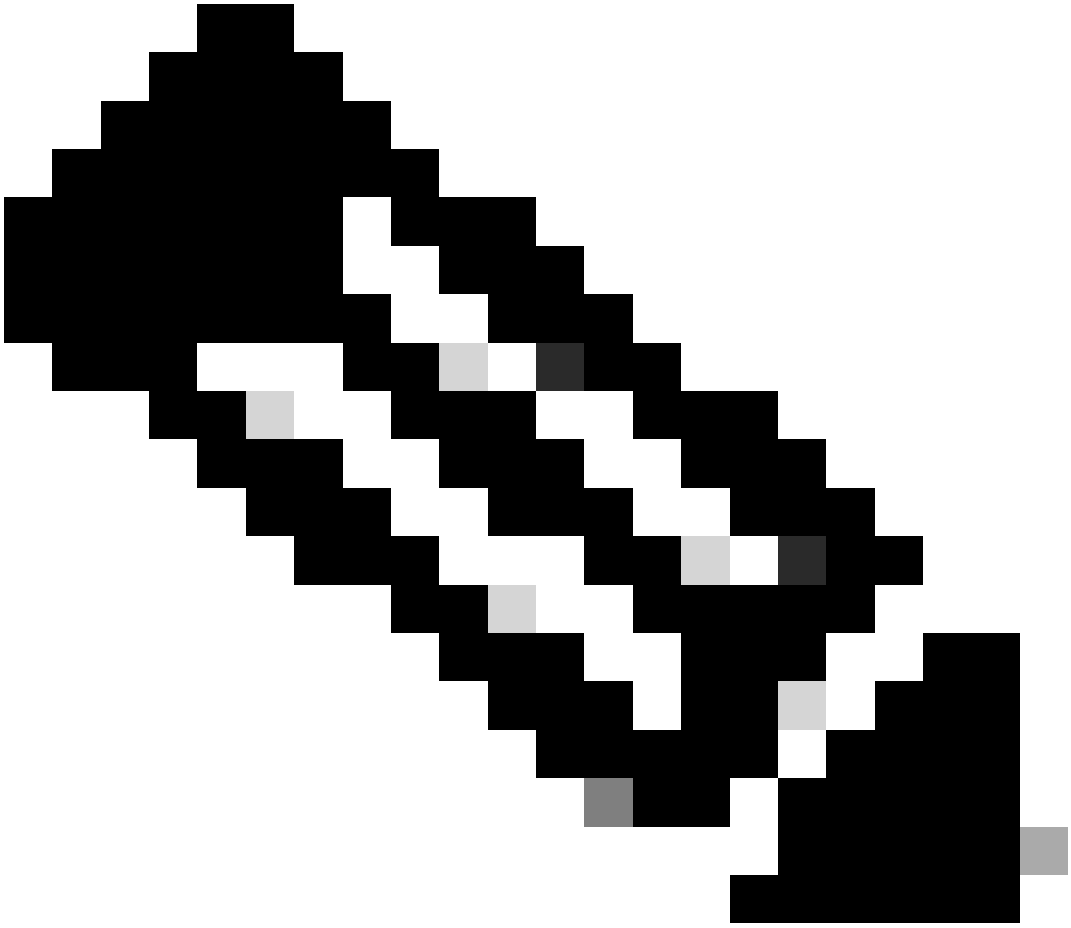
DHCP 스누핑 바인딩이 생성되었는지 확인합니다.

```
<#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
```

문제 해결(모든 CGW 유형)

디버그는 DHCP 스누핑 및 L2 릴레이 프로세스에서 DHCP 패킷을 처리하는 방법을 보여주는 데 유용합니다.



참고: 이러한 디버그는 CGW와 DCHP L2 릴레이를 사용하는 모든 구축 유형에 사용할 수 있습니다.

DHCP 스누핑 디버그(리프)

디버그 스누핑으로 패킷 처리 확인

<#root>

Leaf01#

debug ip dhcp snooping packet

DHCP Snooping Packet debugging is on

Leaf01#

show debugging

DHCP Snooping packet debugging is on

호스트 DHCP 주소 시도 시작

- 이 문서에서는 DORA 교환을 트리거하기 위해 DHCP를 통해 주소가 지정된 SVI의 종료/비종료를 수행했습니다
- Windows 호스트의 경우 ipconfig /release > ipconfig /renew를 수행할 수 있습니다.

show logging 또는 터미널 창에서 디버그를 수집합니다.

DHCP 검색

호스트 연결 포트에서 검색이 표시됨

<#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 오퍼

패브릭 터널 인터페이스에서 오퍼 수신 확인

<#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 요청

호스트 연결 포트에서 요청이 표시됨

<#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
```

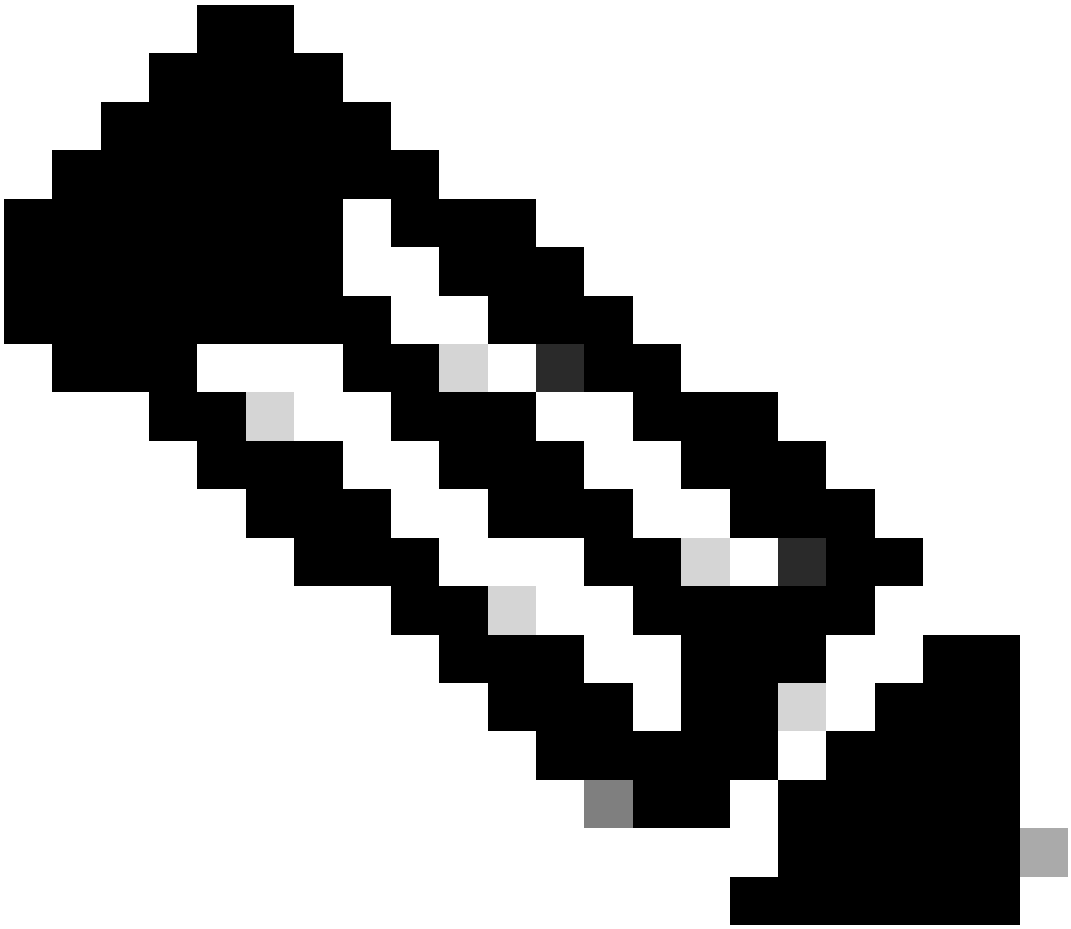
DHCP ACK

패브릭 터널 인터페이스에서 ACK 수신 확인

<#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.
```



참고: 이러한 디버그는 스니핑됩니다. 패킷의 메모리 덤프를 생성하지만 디버그 결과의 이 부분에 대한 주석은 이 문서의 범위를 벗어납니다.

DHCP 스누핑 디버그(CGW)

DHCP 검색

패킷이 CGW에서 전송 및 수신되는 방식(방화벽에서 헤어핀)으로 인해 디버그가 두 번 실행됩니다

Fabric on Tunnel(터널 인터페이스)에서 Fabric(패브릭) VLAN 202의 방화벽으로 Tw 1/0/1 전송

<#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
```

SVI 및 헬퍼로 DHCP 서버로 전송하기 위해 Vlan 2021의 두 1/0/2에 있는 방화벽에서 도착

<#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 오퍼

DHCP 서버에서 도우미가 구성되어 방화벽으로 전달되는 SVI 2021로 다시 전달됩니다.

<#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: Vlan2
*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

패브릭 VLAN의 방화벽에서 도착하여 CGW에서 패브릭으로 Leaf로 전송

<#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 요청

<#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
```

임베디드 캡처

EPC를 사용하여 DHCP 패킷 교환 및 매개변수가 정확한지 확인합니다.

- 이는 CGW의 관점에서 표시되지만 패킷 교환을 확인하기 위해 Leaf에서 프로세스를 반복할 수 있습니다

- 이 예에서는 다른 DHCP 패킷에 대해 프로세스와 분석이 동일하므로 Discover(검색)를 보여 줍니다

리프 루프백에 대한 경로 확인

<#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
```

Leaf01을 향하는 링크에서 실행되도록 캡처 구성

```
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
```

캡처를 시작하고, DHCP IP 주소를 요청하도록 호스트를 트리거하고, 캡처를 중지합니다.

<#root>

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

DHCP Discover(DHCP 검색)로 시작하는 캡처 결과 보기(트랜잭션 ID에 주목하여 모두 동일한 DORA 이벤트인지 확인)

<#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

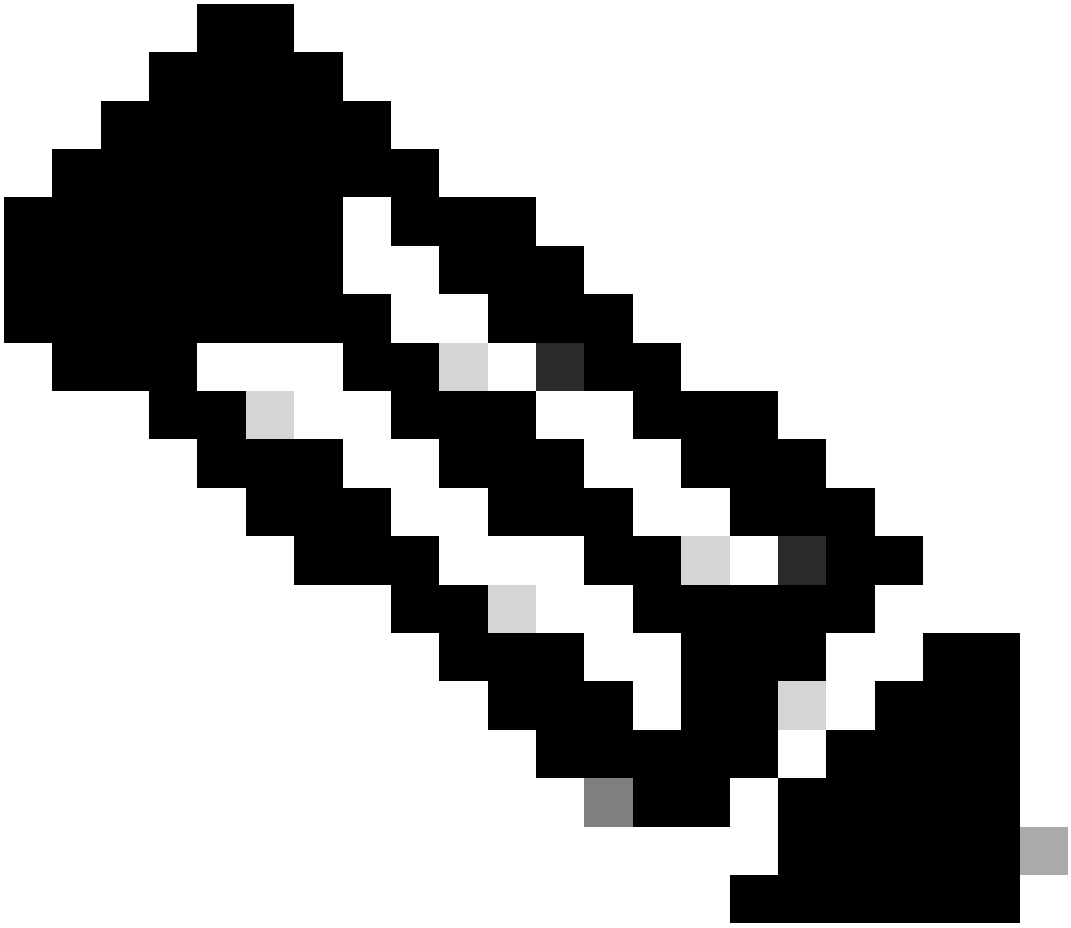
Option: (53) DHCP Message Type (Discover)

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
```



참고: 캡처 툴을 모든 Leaf 또는 CGW에서 사용하여 DHCP DORA 교환의 일부에 장애가 의심되는 마지막 지점을 확인할 수 있습니다.

오류에 대한 스누핑 통계 확인

<#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.

DHCP Snooping에 대한 펀트 경로 확인

- CoPP는 punt 경로의 패킷을 삭제하는 기본 구성 요소입니다

<#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
```

가능한 패킷 플러드가 발생하는 위치를 찾는 데 매우 유용한 또 다른 명령은 'show platform software fed switch active punt rate interfaces'입니다.

- 이는 펀트 경로를 혼잡하게 만들고 합법적인 CPU 바운드 트래픽에 영향을 주는 플러딩이 발생하는 소스 인터페이스를 찾는 데 매우 유용합니다

<#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 스누핑 클라이언트 통계

이 명령을 사용하여 DHCP 메시지 교환을 확인합니다. Leaf 또는 CGW에서 모두 실행하여 이벤트 추적을 볼 수 있습니다

<#root>

Leaf01#

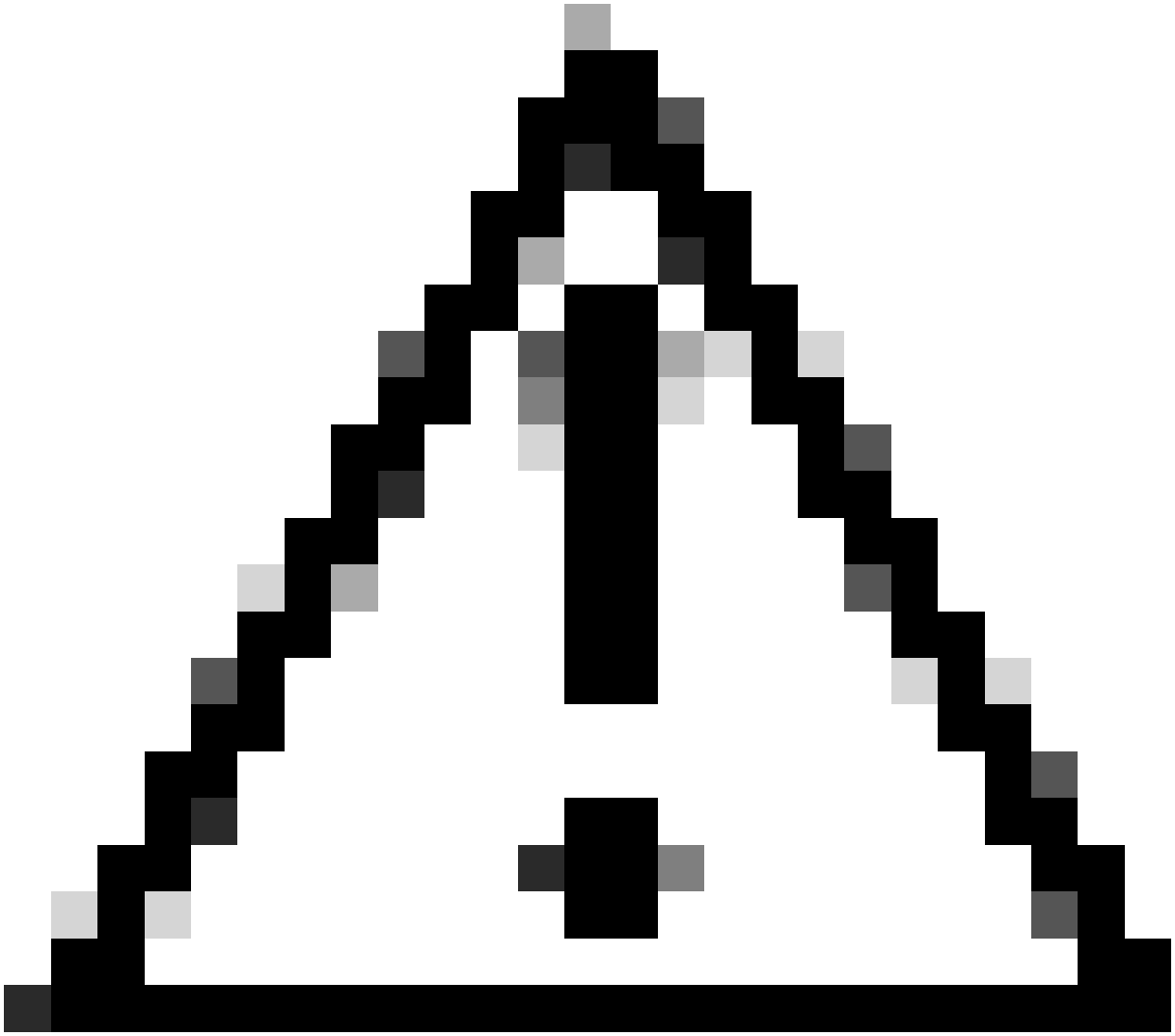
show platform dhcpsnooping client stats 0006.F601.CD43

```
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
```

추가 디버그

```
debug ip dhcp server packet detail
```

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



주의: 디버그를 실행할 때는 주의하십시오!

관련 정보

- Catalyst 9000 Series 스위치에 BGP EVPN 라우팅 정책 구현
- Catalyst 9000 Series 스위치에서 BGP EVPN Protected Overlay Segmentation 구현
- Catalyst 9000 스위치에서 DHCP 스누핑 운영 및 문제 해결

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.