Catalyst 9000シリーズスイッチでのBGP EVPN DHCPレイヤ2リレーの実装

内容

はじめに

このドキュメントでは、EVPN VxLAN DHCP L2リレー機能の設定、確認、トラブルシューティングの方法について説明します。

前提条件

要件

- この機能は、DHCPが使用されるすべてのCGWタイプの導入で使用されます
- 保護セグメンテーションを実装する場合は、次のドキュメントを確認してください
  - Catalyst 9000シリーズスイッチでのBGP EVPNルーティングポリシーの実装
  - Catalyst 9000シリーズスイッチでのBGP EVPN保護オーバーレイセグメンテーションの実装

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- Catalyst 9300
- Catalyst 9400
- Catalyst 9500
- Catalyst 9600
- Cisco IOS® XE 17.12.1以降のバージョン

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

ドキュメントの詳細

このドキュメントは、SVIのないリーフから中央ゲートウェイに向けてDHCPをリレーする必要があるすべてのCGW導入に使用できます。

- 保護されたセグメンテーションを使用していない場合は、SVIがファブリックにアドバタイズされるドキュメントのセクションを使用します

保護されたセグメンテーションを実装している場合、このドキュメントは相互に関連する3つのドキュメントの第2部です。

- ドキュメント1:Catalyst 9000シリーズスイッチでのBGP EVPNルーティングポリシーの実装には、オーバーレイでのBGP BUMトラフィックの制御方法が記載されており、最初に設定する必要があります

- ドキュメント2:Catalyst 9000シリーズスイッチでのBGP EVPN保護オーバーレイセグメンテーションの実装は、ドキュメント1のオーバーレイ設計とポリシーに基づいて構築されており、「protected」キーワードの実装について説明しています。
- ドキュメント3:このドキュメント。最後の2つのドキュメントの上に構築され、レイヤ2のみのリーフとCGWを使用してDHCPリレーを実装する方法について説明します。

L2リレーの動作

リレー	スヌーピング	コアフラッド	アクセスフラッド	IPv4
yes	yes	いいえ	yes	• オプション82サブオプション：(1)エージェント回線ID(vni-mod-port)にdhcpスヌーピングが設定される • DHCP信頼設定を使用して、アクセス側を制限できます *推奨モデル
yes	いいえ	yes	yes	• オプション82サブオプション：(1)エージェント回線ID(vlan-mod-port)にdhcpスヌーピングが設定される
いいえ	yes	いいえ	yes	• オプション82サブオプション：(1)エージェント回線ID(vni-mod-port)にdhcpスヌーピングが設定される • DHCP信頼設定を使用して、アクセス側を制限できます
リレー	スヌーピング	コアフラッド	アクセスフラッド	IPv6
yes	yes	yes	yes	• オプション82サブオプション：(1)エージェント回線ID(vni-mod-port)にdhcpスヌーピングが設定される • DHCP信頼設定を使用して、アクセス側を制限できます
yes	いいえ	yes	yes	• オプション82サブオプション：(1)エージェント回線ID(vlan-mod-port)にdhcpスヌーピングが設定される
いいえ	yes	yes	yes	• オプション82サブオプション：(1)エージェント回線ID(vni-mod-port)にdhcpスヌーピングが設定される • DHCP信頼設定を使用して、アクセス側を制限できます
いいえ	いいえ	yes	yes	

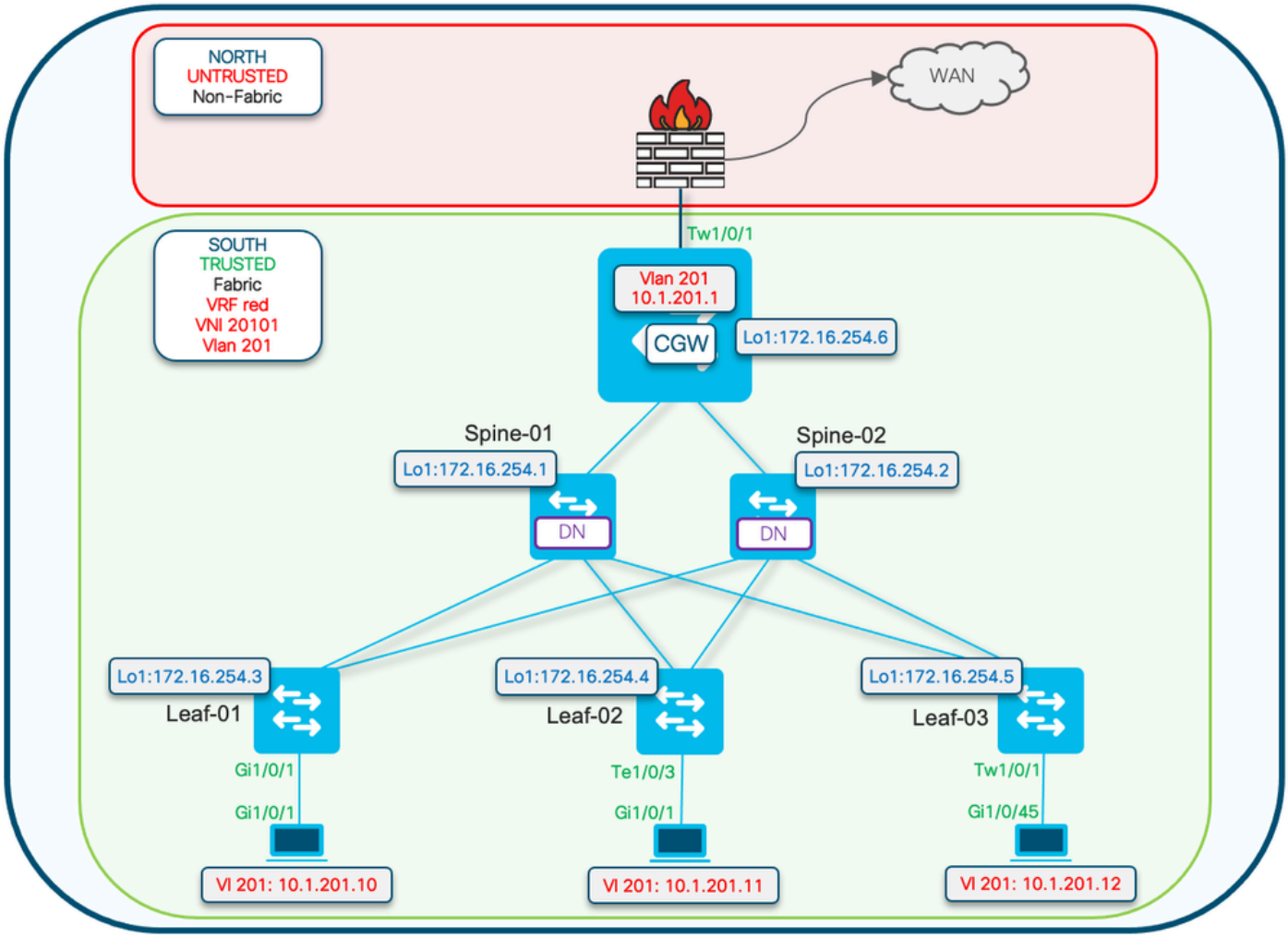
用語

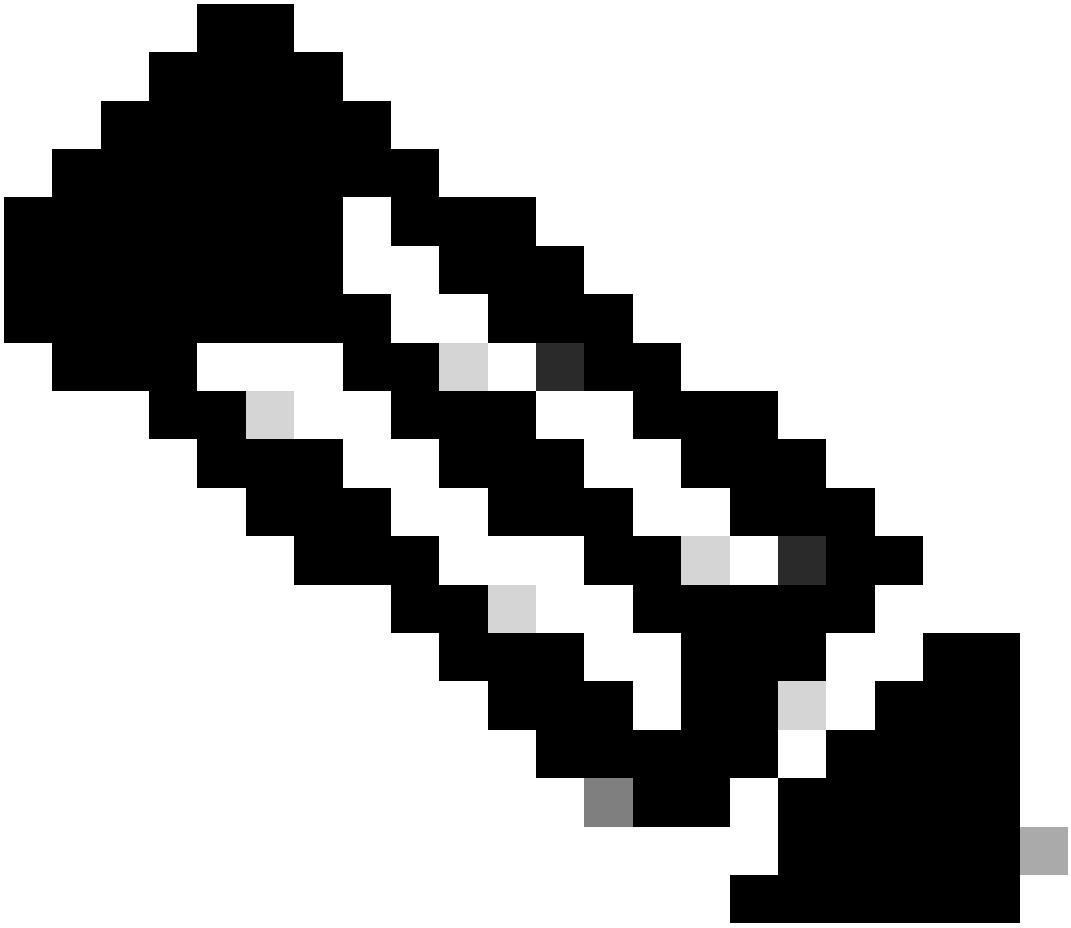
VRF	仮想ルーティング転送	他のVRFおよびグローバルIPv4/IPv6ルーティングドメインから分離されたレイヤ3ルーティングドメインを定義する
AF	アドレスファミリ	BGPが処理するプレフィックスとルーティング情報のタイプを定義します。
AS	自律システム	ネットワークまたはネットワークの集合に属し、単一のエンティティまたは組織によってすべて管理、制御、および監視される一連のインターネットルーティング可能なIPプレフィクス
EVPN	イーサネット仮想プライベートネットワーク	BGPにレイヤ2 MACおよびレイヤ3 IP情報の転送を許可する拡張はEVPNであり、VXLANオーバーレイネットワークに関連する到達可能性情報を配布するプロトコルとしてマルチプロトコルボーダーゲートウェイプロトコル(MP-BGP)を使用します。
VXLAN	仮想拡張LAN（ローカルエリアネットワーク）	VXLANは、VLANとSTPに固有の制限を克服するように設計されています。VLANと同じイーサネットレイヤ2ネットワークサービスを提供するIETF標準[RFC 7348]として提案されていますが、柔軟性に優れています。機能的には、レイヤ3アンダーレイネットワーク上で仮想オーバーレイとして動作するMAC-in-UDPカプセル化プロトコルです。
CGW	中央集中型ゲートウェイ	ゲートウェイSVIが各リーフ上にない場合のEVPNの実装。その代わり、すべてのルーティングは非対称IRB(Integrated Routing and Bridging)を使用して特定のリーフによって実行されます
DEF GW	[Default Gateway]	BGP拡張コミュニティアトリビュートが、「l2vpn evpn」設定セクションで「default-gateway advertise enable」コマンドを使用してMAC/IPプレフィクスに追加された。
IMET(RT3)	包括マルチキャストイーサネットタグ（ルート）	BGPタイプ3ルートとも呼ばれます。このルートタイプは、VTEP間でBUM（ブロードキャスト/不明ユニキャスト/マルチキャスト）トラフィックを配信するためにEVPNで使用されます。
RT2	ルートタイプ2	ホストMACまたはゲートウェイMAC-IPを表すBGP MACまたはMAC/IPプレフィクス

EVPNマネージャ	EVPNマネージャ	その他のさまざまなコンポーネントの中央管理コンポーネント（例：SISFから学習し、L2RIBに信号を送信）
SISF	スイッチ統合セキュリティ機能	リーフ上に存在するローカルホストを学習するためにEVPNによって使用される非依存のホストトラッキングテーブル
L2リブ	レイヤ2ルーティング情報ベース	BGP間のインタラクションを管理するための中間コンポーネント、EVPN Mgr、L2FIB
FED	転送エンジンドライバ	ASIC（ハードウェア）層をプログラムする
マトム	Macアドレステーブルマネージャ	IOS MATM：ローカルアドレスと FED MATM：コントロールプレーンから学習したローカルアドレスとリモートアドレスをインストールするハードウェアテーブル。ハードウェアフォワーディングプレーンの一部です。

設定（標準のCGW導入）

ネットワーク図


NORTH
UNTRUSTED
Non-Fabric
WAN
SOUTH
TRUSTED
Fabric
VRF red
VNI 20101
Vlan 201
Tw1/0/1
Vlan 201
10.1.201.1
CGW
Lo1:172.16.254.6
Spine-01
Lo1:172.16.254.1
DN
Spine-02
Lo1:172.16.254.2
DN
Lo1:172.16.254.3
Leaf-01
Gi1/0/1
Gi1/0/1
Vl 201: 10.1.201.10
Lo1:172.16.254.4
Leaf-02
Te1/0/3
Gi1/0/1
Vl 201: 10.1.201.11
Lo1:172.16.254.5
Leaf-03
Tw1/0/1
Gi1/0/45
Vl 201: 10.1.201.12



注：このセクションでは、保護機能を使用しない標準的なCGWの導入について説明します。

- DHCP DORAパケット交換を示すデバッグは、保護セグメントの例にのみ示されています

L2 VTEP（リーフ）キーの詳細

要求パケットがクライアントから送信される

0.

- Default gw advertised CGW macを使用します。
- 複数のgwが存在する場合、最初にgw macが使用されます。
- 外部ブロードキャストMAC（クライアント開始：DORAのDおよびR）をユニキャストGW MACに変換し、CGWに転送する

0.

DHCPスヌーピング追加：オプション82サブオプション：回線およびRID

（RIDはCGWの応答パケット処理で使用される）

(CGWにローカルではなく、L2VTEPへのファブリックリレーに戻るように通知)

<#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トンネルを介してCGWから受信した応答パケット
- リーフストリップオプション82
- クライアントのソースインターフェイスでバインディングエントリを追加します。（vxlan-mod-portはクライアントの送信元インターフェイスを提供）
- クライアントに転送される応答パケット

L3 VTEP(CGW)キーの詳細

- Dhcpスヌーピングを有効にする
- SVIでDHCPリレーを有効にする
- 要求がL2VTEPから受信され、リレーに渡されます
- リレーは他のオプション82サブオプション（gi、server overrideなど）を追加し、DHCPサーバに送信します
- DHCPサーバからのDHCP応答が最初にリレーコンポーネントに到達する
- RELAYがオプション82パラメータ（GIアドレス、サーバの上書きなど）を削除すると、パケットはdhcpスヌーピングコンポーネントに渡されます
- スヌーピングコンポーネントはRID（ルータID）をチェックし、ローカルでない場合はオプション82サブオプション1および2を削除しません

- ファブリックリレー（RIDはローカルではないため）パケットはリモートクライアントに直接転送される
- クライアントMacを使用し、ブリッジインジェクトを行う ハードウェアはクライアントMACルックアップを行い、vxlan encapを使用してパケットを発信元のL2VTEPに転送します。

  0.

L2VTEP

evpnインスタンスの設定

<#root>

Leaf-01#

show run | beg l2vpn evpn instance 201

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

DHCPスヌーピングの有効化

<#root>

Leaf-01#

show run | sec dhcp snoop

```
ip dhcp snooping vlan 101,
201

ip dhcp snooping
```

CGW

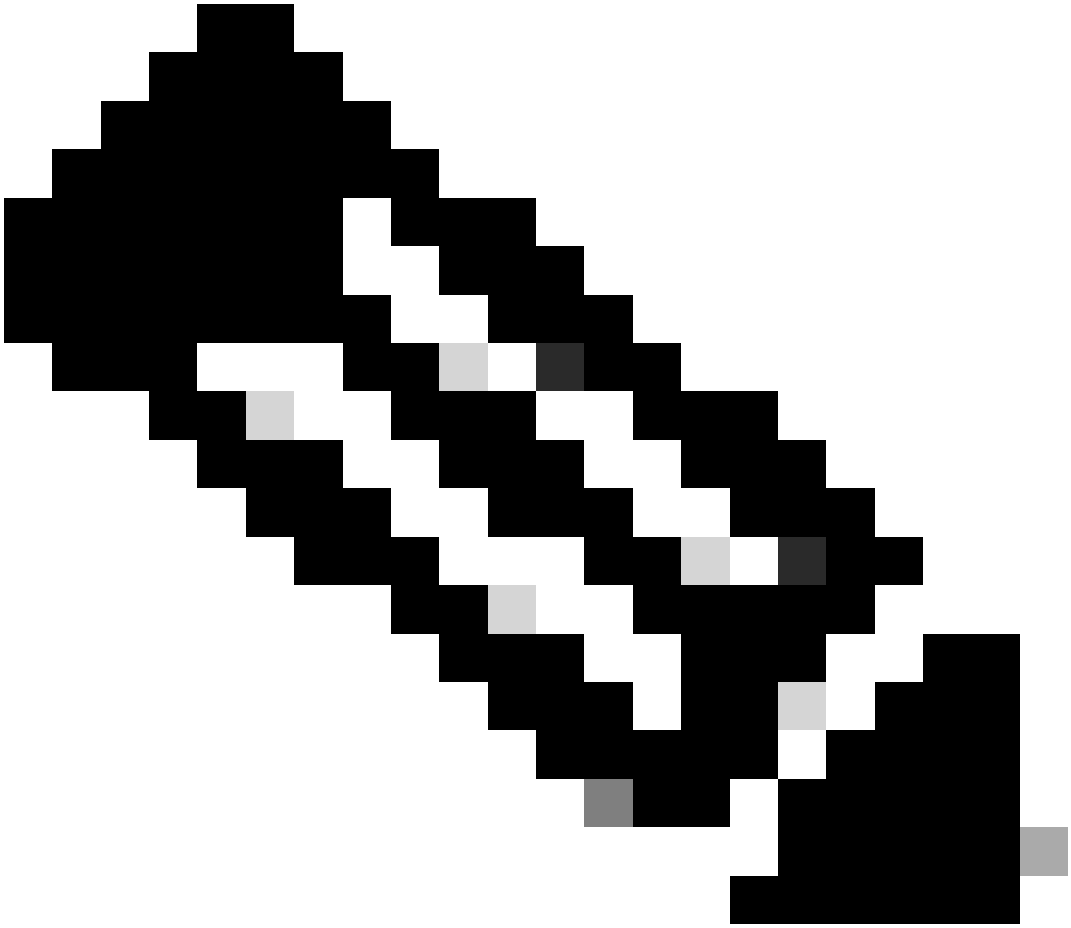
evpnインスタンスの設定

<#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
```



注：DEF GW属性は、L2リレーがDHCPパケットのカプセル化と送信先を知るために重要です。

DHCPスヌーピングの有効化

<#root>

Border#

sh run | s dhcp snoop

```
ip dhcp snooping vlan 101,

201


ip dhcp snooping
```

追加オプションを処理するための正しい設定がDHCPリレーにあることを確認します

<#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
```

検証（標準のCGW導入）

ゲートウェイプレフィクス（リーフ）

<#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 (リーフ)

<#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
```

ローカルMAC (リーフ)

<#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スヌーピング (リーフおよびCGW)

<#root>

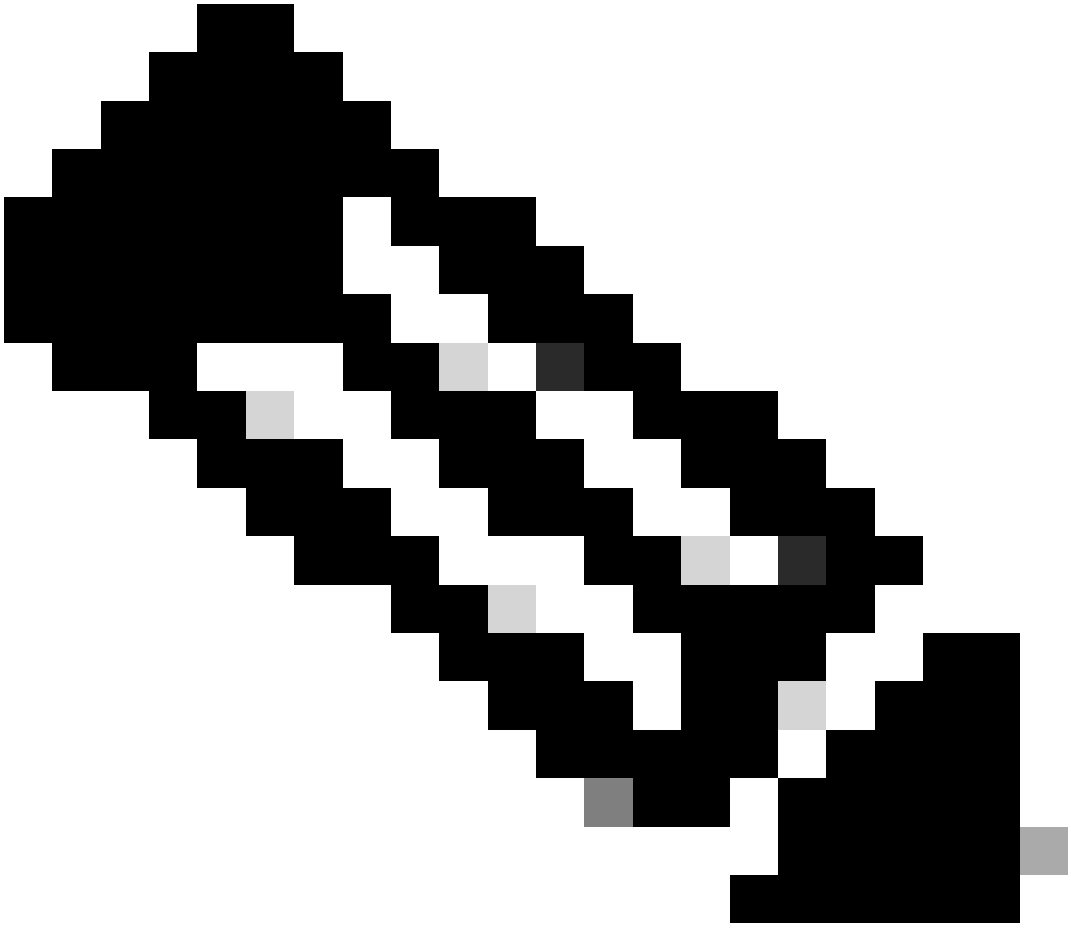
```
Leaf-01#

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
```

構成（部分的に分離された保護）

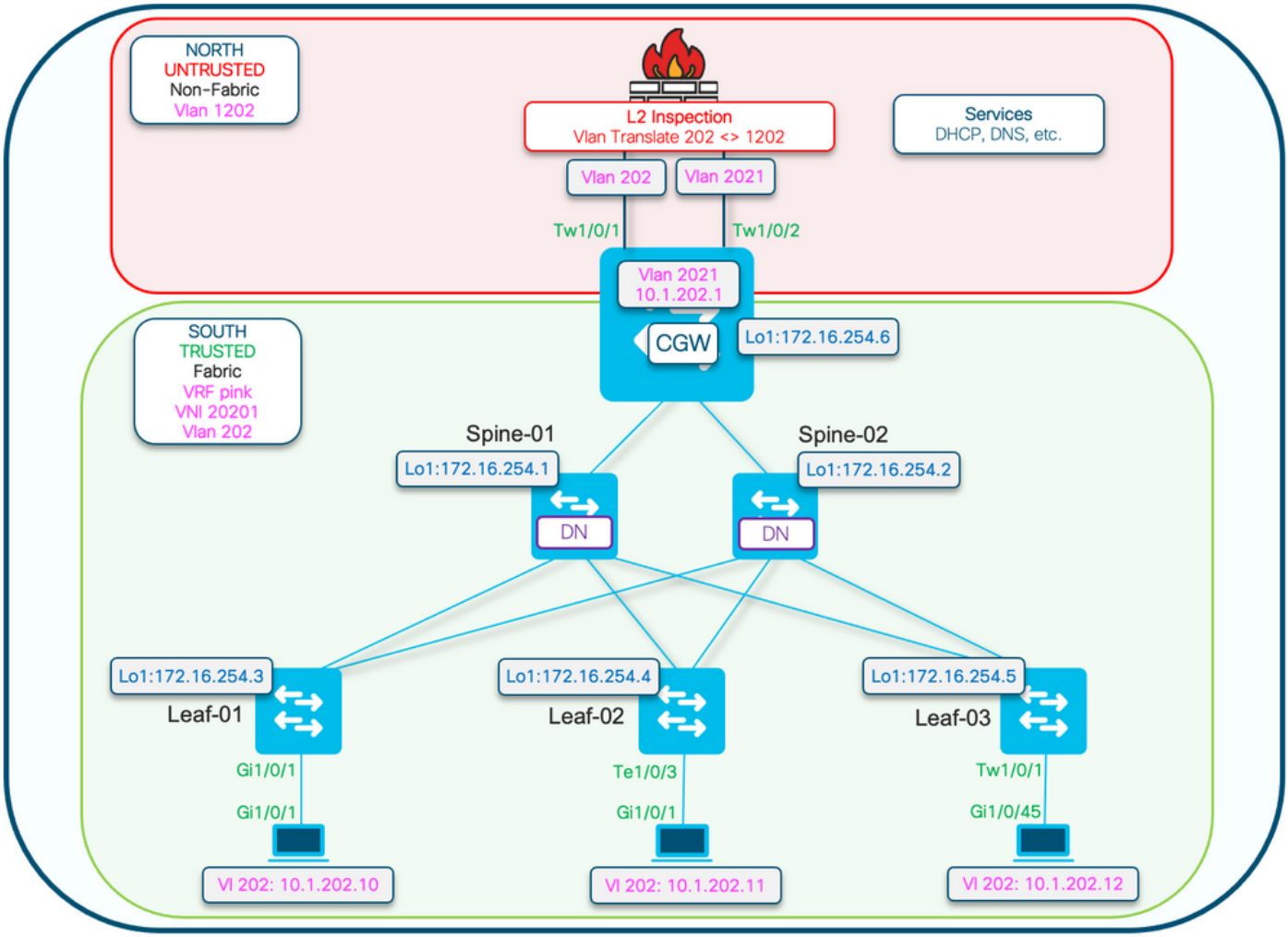
アクセスリーフでのDHCPスヌーピングは、CGWからのデフォルトゲートウェイルートを使用して、DHCPパケットの転送先ゲートウェイMACを学習します。

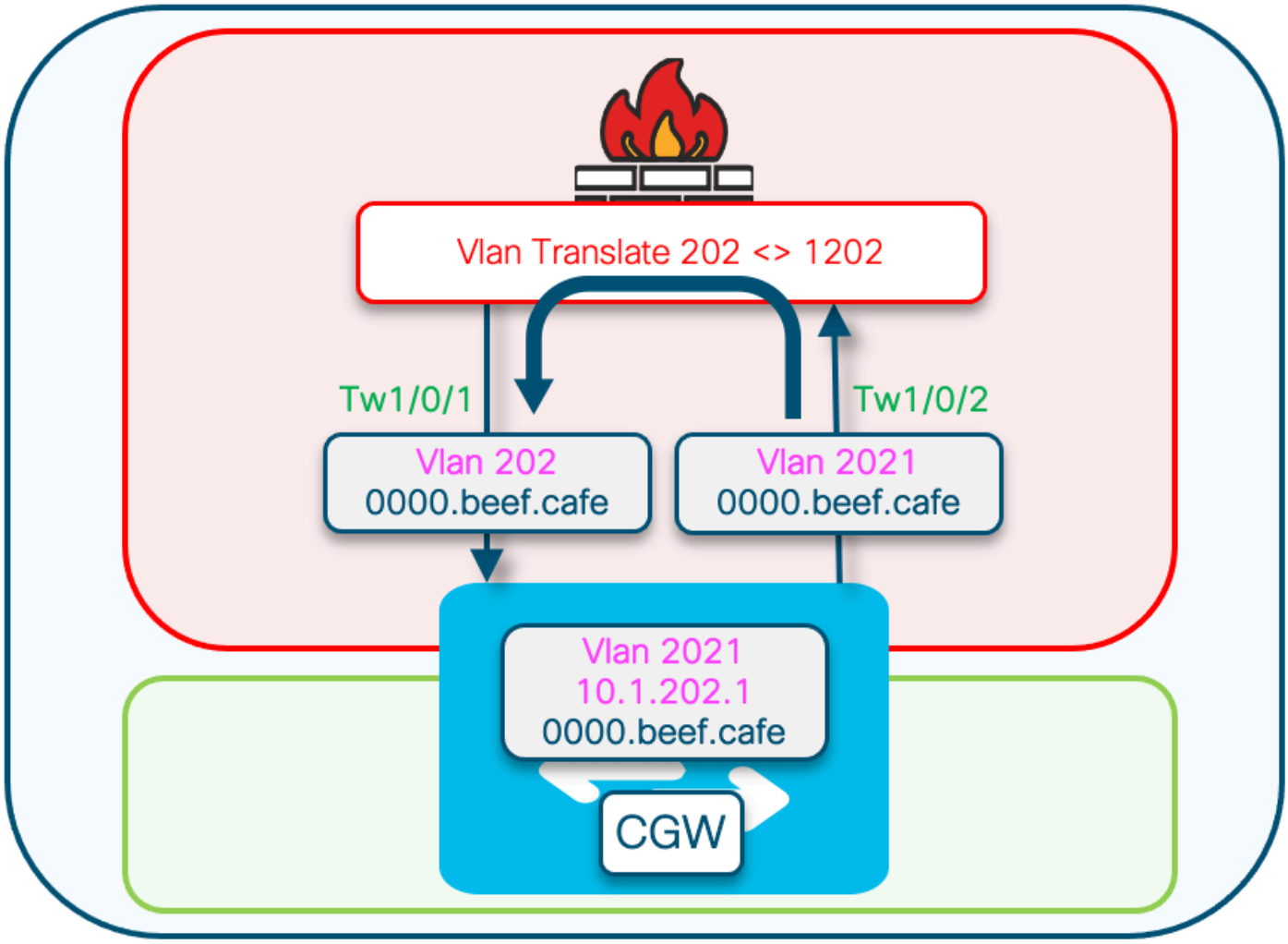
- 外部ゲートウェイを使用して部分的に分離された設計を使用する場合、デフォルトゲートウェイ(DEF GW)属性でMAC-IP RT2をアドバタイズするために、CGWで追加の設定が必要になります。



注：注：このセクションでは、完全に隔離された保護セグメントの実装についても説明します。この実装では、（ファブリック外のGWではなく）ファブリックにアドバタイズされるGWも使用します。

ネットワーク図


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（リーフ）キーの詳細

要求パケットがクライアントから送信される

0.

- Default gw advertised CGW macを使用します。
- 複数のgwが存在する場合、最初にgw macが使用されます。
- 外部ブロードキャストMAC（クライアント開始：DORAのDおよびR）をユニキャストGW MACに変換し、CGWに転送する

0.

DHCPスヌーピング追加：オプション82サブオプション：回線およびRID

（RIDはCGWの応答パケット処理で使用される）

(CGWにローカルではなく、L2VTEPへのファブリックリレーに戻るように通知)

<#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トンネルを介してCGWから受信した応答パケット
- リーフストリップオプション82
- クライアントのソースインターフェイスでバインディングエントリを追加します。（vxlan-mod-portはクライアントの送信元インターフェイスを提供）
- クライアントに転送される応答パケット

L3 VTEP(CGW)キーの詳細

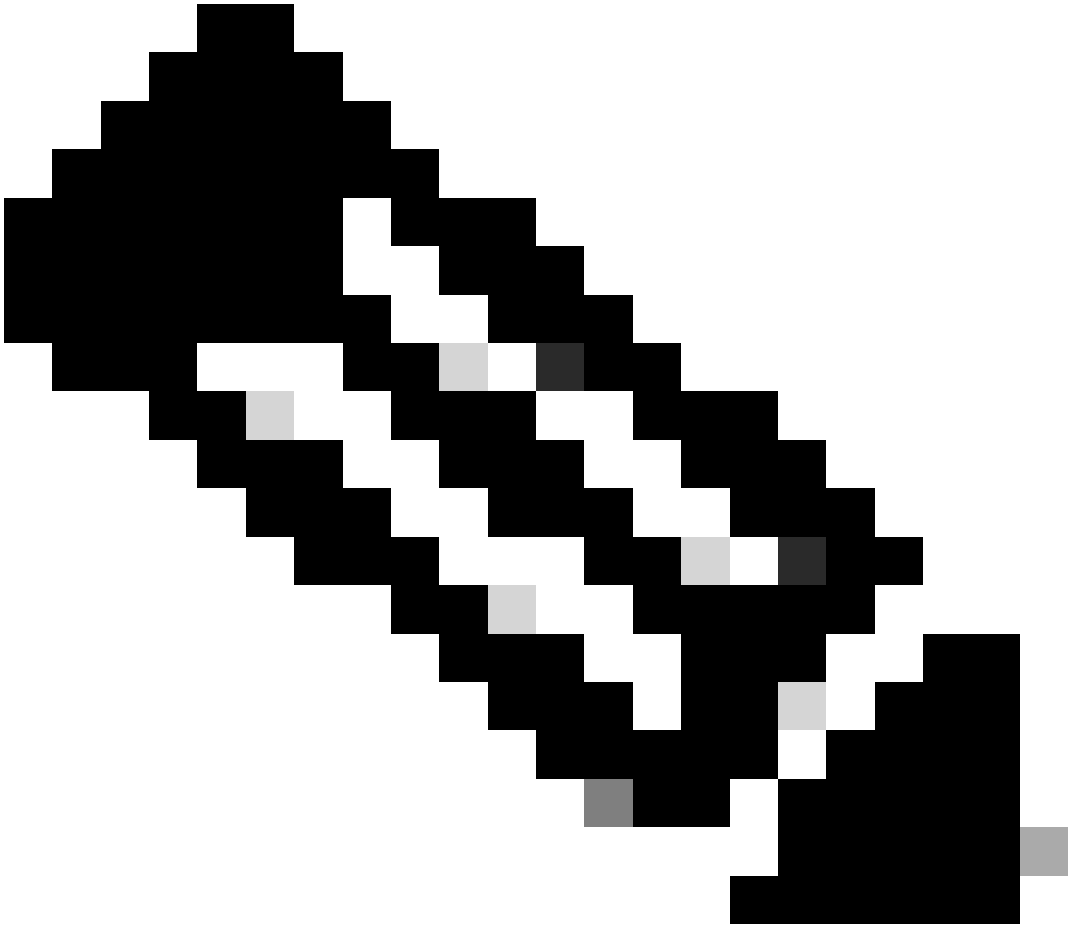
- Dhcpスヌーピングを有効にする
- SVIでDHCPリレーを有効にする
- 要求がL2VTEPから受信され、リレーに渡されます
- リレーは他のオプション82サブオプション（gi、server overrideなど）を追加し、DHCPサーバに送信します
- DHCPサーバからのDHCP応答が最初にリレーコンポーネントに到達する
- RELAYがオプション82パラメータ（GIアドレス、サーバの上書きなど）を削除すると、パケットはdhcpスヌーピングコンポーネントに渡されます
- スヌーピングコンポーネントはRID（ルータID）をチェックし、ローカルでない場合はオプション82サブオプション1および2を削除しません
- ファブリックリレー（RIDはローカルではないため）パケットはリモートクライアントに直接転送される
- クライアントMacを使用し、ブリッジインジェクトを行う ハードウェアはクライアントMACルックアップを行い、vxlan encapを使用してパケットを発信元のL2VTEPに転送します。

  0.

0.

DHCP L2リレーのサポートに必要な手順：

1. IPローカルラーニングの有効化
2. 収集を無効にしてポリシーを作成します
3. 外部ゲートウェイevi/vlanへの接続
4. 外部ゲートウェイmac-ipのデバイストラッキングテーブルにスタティックエントリを追加する
5. RT2 MAC-IPプレフィクスに一致するBGPルートマップを作成し、デフォルトゲートウェイ拡張コミュニティを設定します
6. BGPルートリフレクタネイバーへのルートマップの適用
7. 追加オプションを処理するための正しい設定がDHCPリレーにあることを確認します
8. ファブリックVLANと外部GW VLANでのDHCPスヌーピングの設定



注：外部ゲートウェイでDHCP L2リレーをサポートするために、アクセスリーフの設定を変更する必要はありません。

CGW

IPローカルラーニングの有効化

<#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
```

収集を無効にしてポリシーを作成します

<#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
```

外部ゲートウェイevi/vlanへの接続

<#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
```

外部ゲートウェイmac-ipのデバイストラッキングテーブルにスタティックエントリを追加する

<#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プレフィクスに一致するBGPルートマップを作成し、デフォルトゲートウェイ拡張コミュニティを設定します

<#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ルートリフレクタネイバーへのルートマップの適用

<#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
```

追加オプションを処理するための正しい設定がDHCPリレーにあることを確認します

<#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
```

ファブリックVLANおよび外部GW VLANでのDHCPスヌーピングの設定

<#root>

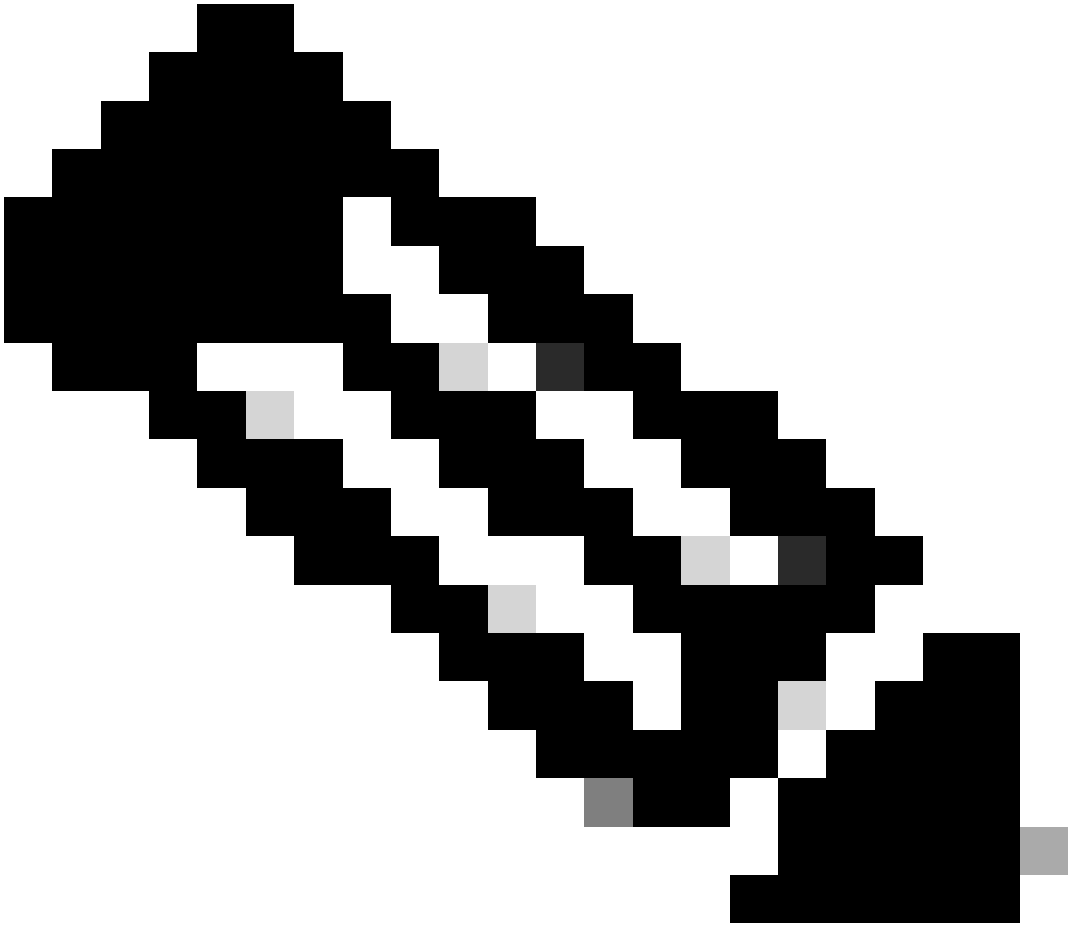
```
Leaf01#

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サーバへのアップリンクがCGWで信頼されていることを確認します

<#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
```



注：サーバがファイアウォールデバイスの信頼に配置される方法が、このデバイスに面

する両方のリンクに設定されています。拡大された図では、この設計でオファーがTw1/0/1とTw1/0/2の両方に到達することがわかります。

検証（部分的に分離された保護）

ゲートウェイプレフィクス（リーフ）

<#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（リーフ）

リーフがCGWリモートMACをハードウェアにインストールしたことを確認します

<#root>

```
Leaf01#

show platform software fed switch active matm macTable vlan 202

VLAN   MAC                 Type  Seq#    EC_Bi  Flags  machandle          siHandle           riHandle
-------------------------------------------------------------------------------------------------------
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_RESY
MAT_DO_NOT_AGE           0x100  MAT_SECURE_ADDR          0x200  MAT_NO_PORT              0x400  MAT_DROP
MAT_DUP_ADDR            0x1000  MAT_NULL_DESTINATION    0x2000  MAT_DOT1X_ADDR          0x4000  MAT_ROUT
MAT_WIRELESS_ADDR      0x10000  MAT_SECURE_CFG_ADDR    0x20000  MAT_OPQ_DATA_PRESENT   0x40000  MAT_WIRE
MAT_DLR_ADDR          0x100000  MAT_MRP_ADDR          0x200000  MAT_MSRP_ADDR         0x400000  MAT_LISP

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
```

ローカルMAC（リーフ）

<#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スヌーピング（リーフおよびCGW）

ファブリックVLANのリーフでDHCPスヌーピングが有効になっていることを確認します

<#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...>
```

ファブリックおよび外部ゲートウェイVLANのCGWでDHCPスヌーピングが有効になっていることを確認します

<#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スヌーピングバインディングが作成されたことを確認します

<#root>

Leaf01#

show ip dhcp snooping binding

MacAddress

IpAddress

Lease(sec) Type VLAN

Interface

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

00:06:F6:01:CD:43

10.1.202.10

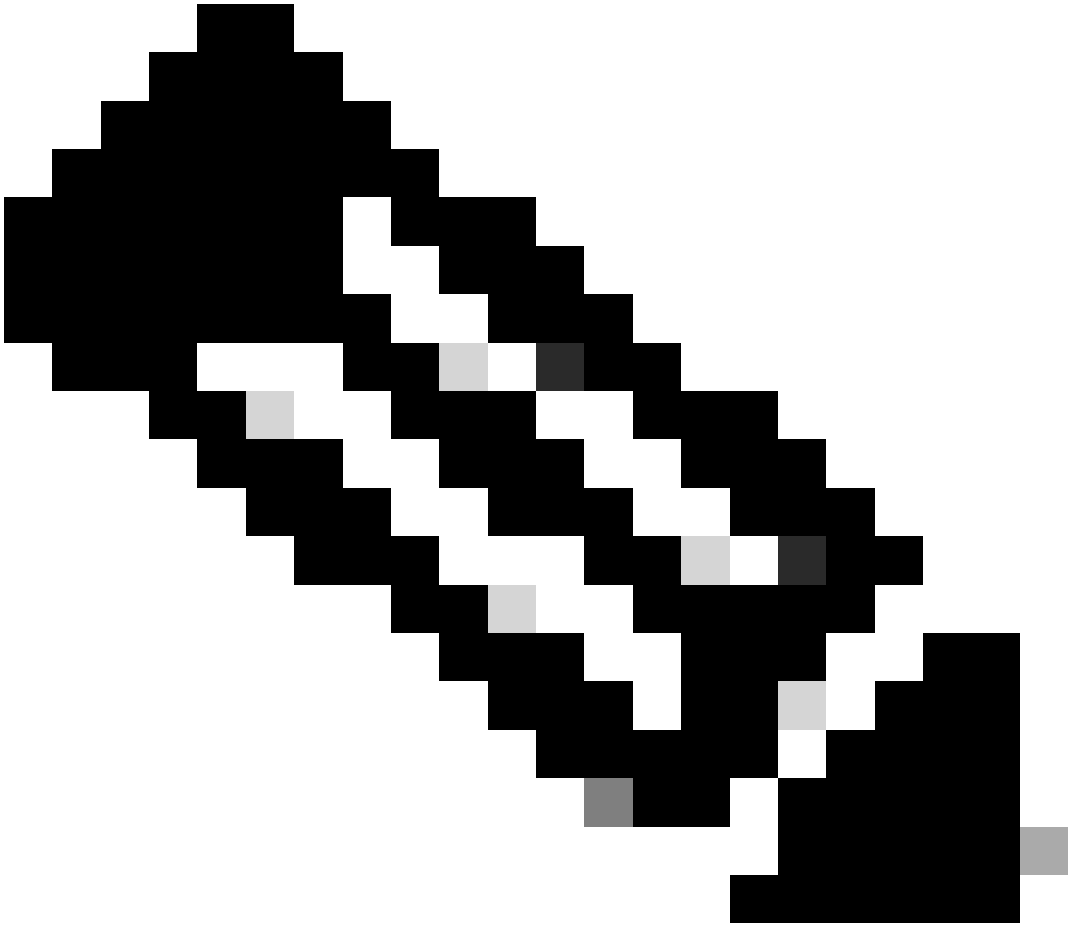
34261 dhcp-snooping 202

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

Total number of bindings: 1

トラブルシューティング（任意のCGWタイプ）

デバッグは、DHCPスヌーピングとL2リレープロセスがDHCPパケットをどのように処理しているかを示すのに役立ちます。



注：これらのデバッグは、DCHP L2リレーを使用するCGWを使用するすべてのタイプの導入に使用できます。

DHCPスヌーピングデバッグ（リーフ）

パケット処理を確認するためのデバッグスヌーピング

<#root>

```
Leaf01#
debug ip dhcp snooping packet
DHCP Snooping Packet debugging is on
Leaf01#
show debugging
DHCP Snooping packet debugging is on
```

ホストのDHCPアドレスの試行を開始する

- このドキュメントでは、DHCPを介してアドレス指定されたSVIのshut/no shutを実行してDORA交換をトリガーします
- Windowsホストの場合、ipconfig /release > ipconfig /renewを実行できます。

show loggingまたはターミナルウィンドウからデバッグを収集します

DHCP DISCOVER

ディスカバリはホスト側のポートから行われていることが確認できます

<#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

ファブリックトンネルインターフェイスからオファーが到達していることが確認できる

<#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

要求がホスト側のポートから見える

<#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
```

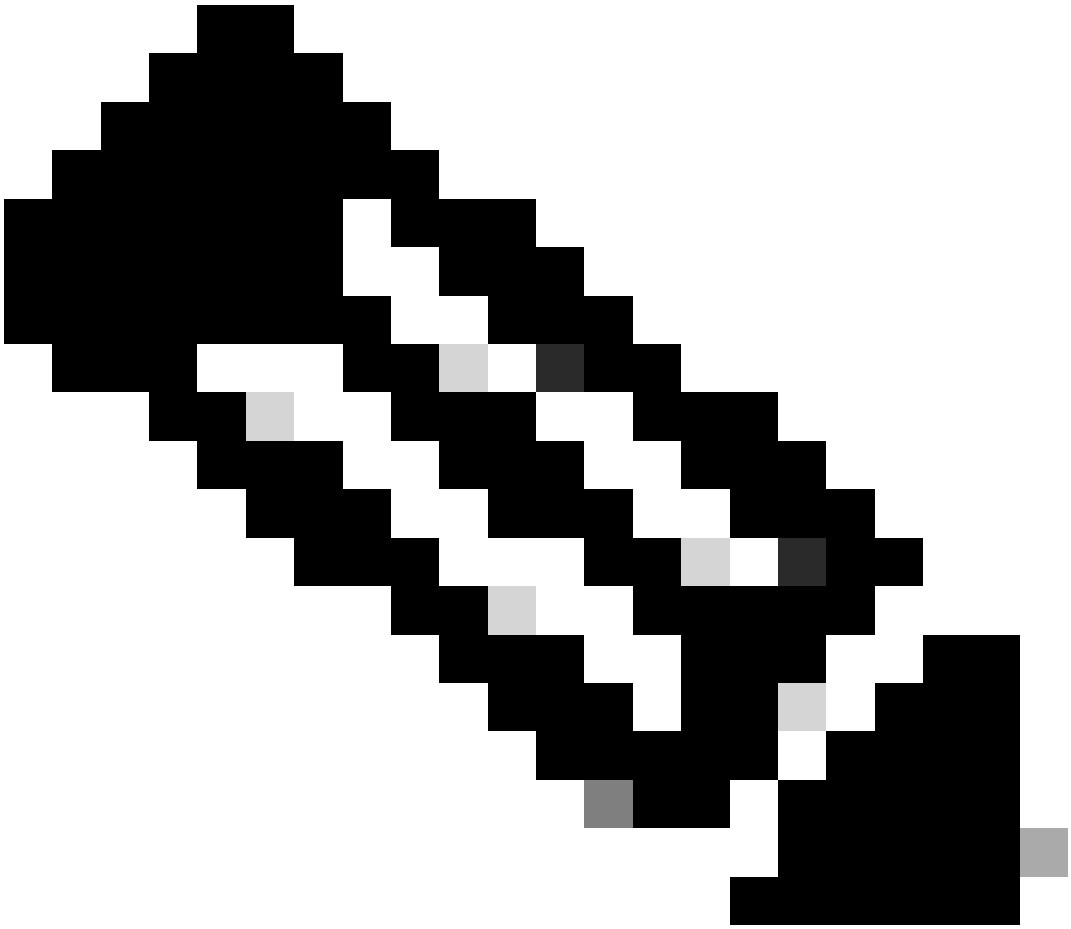
DHCP ACK

ファブリックトンネルインターフェイスからACKが到達していることが確認される

<#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.
```



> 注：これらのデバッグは省略されています。これらのコマンドではパケットのメモリダンプが生成されますが、デバッグ結果のこの部分についての注釈は、このドキュメントでは扱いません。

DHCPスヌーピングデバッグ(CGW)

DHCP DISCOVER

パケットがCGW（ファイアウォールでヘアピニングされている）でどのように送受信されるため、デバッグは2回発生します

ファブリックからトンネルインターフェイスに到達し、Tw 1/0/1からファブリックVLAN 202のファイアウォールに送信

<#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

VLAN 2021のTw 1/0/2のファイアウォールからSVIとヘルパーに送信されてDHCPサーバに到着する

<#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

DHCPサーバからSVI 2021に戻り、そこでヘルパーが設定され、ファイアウォールに転送される

<#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

ファブリックVLANのファイアウォールから着信し、CGWからファブリックにリーフに向けて送信される

<#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

組み込みキャプチャ

EPCを使用して、DHCPパケット交換とパラメータが正しいことを確認します。

- これはCGWの観点から示されますが、パケット交換を確認するためにリーフでこのプロセスを繰り返すことができます
- 次の例は、他のDHCPパケットのプロセスと分析が同じであるため、ディスカバリを示しています

リーフループバックへのルートの確認

<#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
```

Leaf01に面したリンクで実行するようにキャプチャを設定します

```
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
```

キャプチャを開始し、ホストにDHCP IPアドレスを要求するようにトリガーし、キャプチャを停止します。

<#root>

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

DHCP Discoverで始まるキャプチャ結果を表示します（これがすべて同じDORAイベントである

ことを確認するには、トランザクションIDに注意してください）。

<#root>

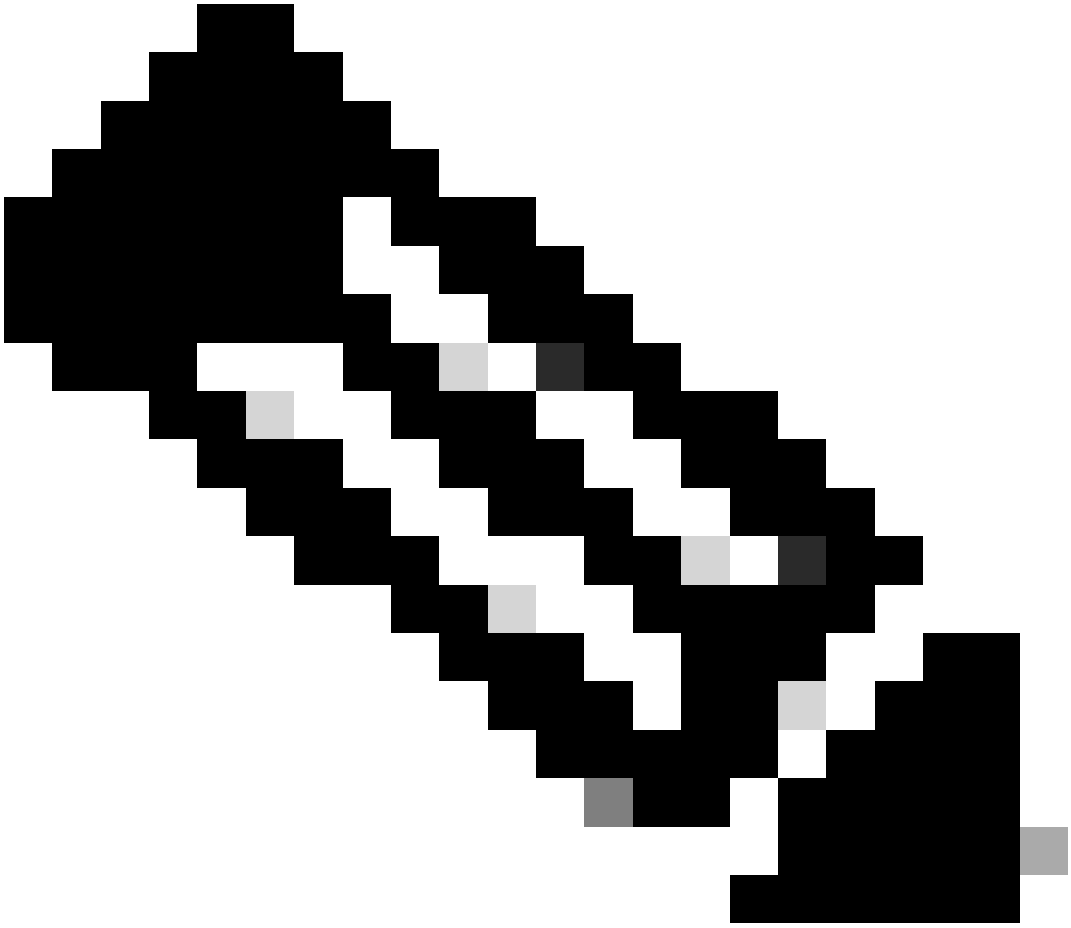
```
CGW#
show monitor cap 1 buff brief | i DHCP


 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
```



注：キャプチャツールは、任意のリーフまたはCGWで、DHCP DORA交換の一部で障害が発生している疑いのある最後のポイントを判別するために使用できます。

エラーートのスヌーピング統計情報の確認

<#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スヌーピングのパントパスの確認

- CoPPは、パントパスでパケットをドロップする主要コンポーネントです

<#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
```

パケットフラッディングが発生している可能性のある場所を特定するのにも非常に便利なコマンドが、「show platform software fed switch active punt rates interfaces」です。

- これは、フラッディングが発生している送信元インターフェイスを見つけるのに非常に役立ちます。フラッディングはパントパスをいっぱいにし、正当なCPUに送られるトラフィックに影響を与えます

<#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スヌーピングクライアント統計情報

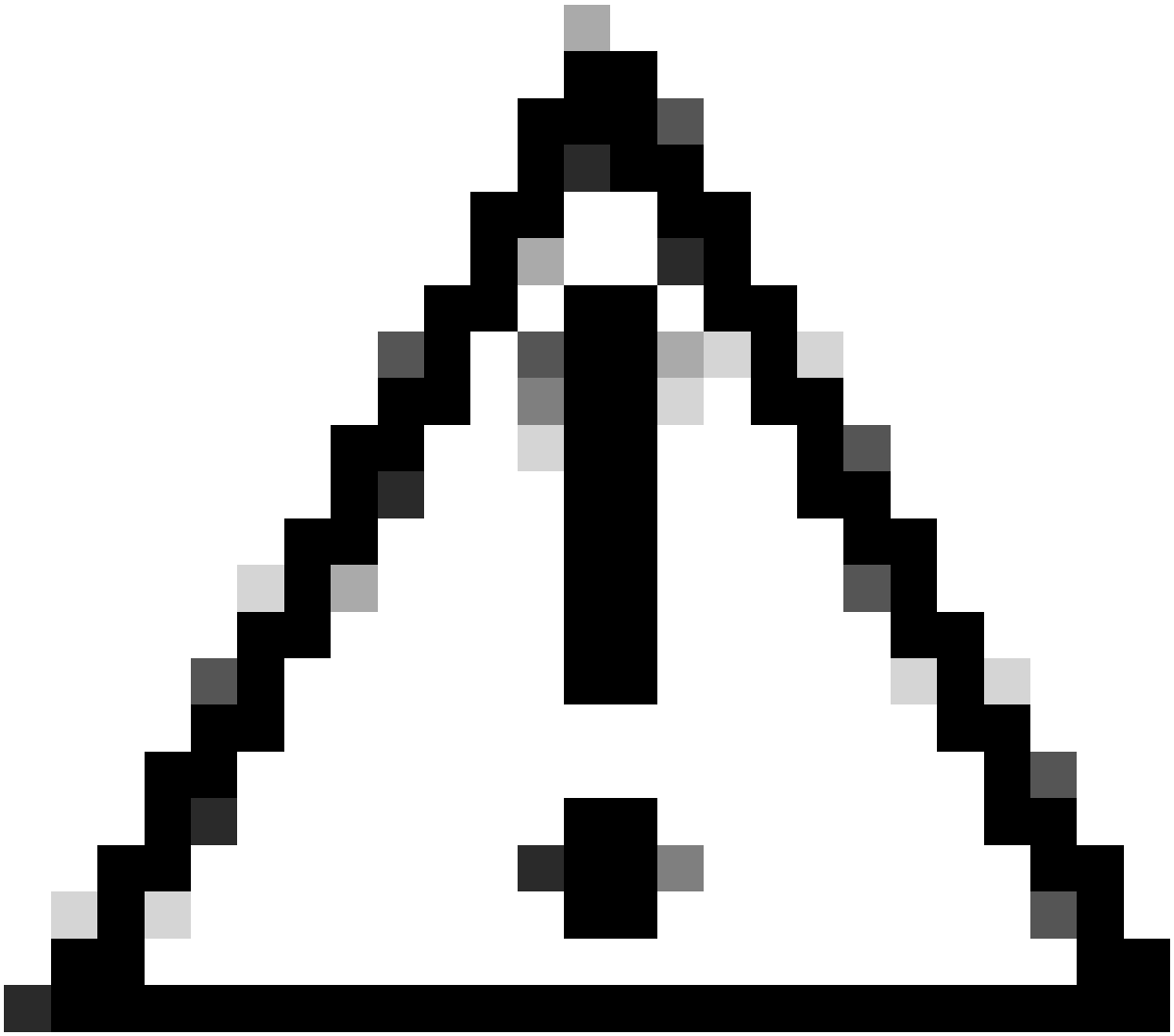
次のコマンドを使用して、DHCPメッセージ交換を確認します。これは、イベントトレースを確認するために、リーフまたはCGWの両方で実行できます

<#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
```

その他のデバッグ

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



注意：デバッグを実行する際には注意が必要です。

関連情報

- Catalyst 9000シリーズスイッチでのBGP EVPNルーティングポリシーの実装
- Catalyst 9000シリーズスイッチでのBGP EVPN保護オーバーレイセグメンテーションの実装
- Catalyst 9000スイッチでのDHCPスヌーピングの操作とトラブルシューティング

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。