



EVPN の概要

イーサネット VPN (EVPN) は、MPLS ネットワークを介してイーサネット マルチポイント サービスを提供する次世代のソリューションです。EVPN は、コアでコントロールプレーン ベースの MAC ラーニングを可能にする既存の仮想プライベート LAN サービス (VPLS) とは 対照的に動作します。EVPN では、EVPN インスタンスに参加している PE が MP-BGP プロト コルを使用してコントロールプレーン内でカスタマー MAC ルートを学習します。コントロ ールプレーン MAC ラーニングは、フローごとのロードバランシングによるマルチホーミングの サポートなど、VPLS の欠点に EVPN で対処できるようにする数多くの利点をもたらします。

EVPN は、ネットワークでの次の新たなニーズに対応するソリューションをネットワーク オペ レータに提供します。

- データセンター相互接続操作 (DCI)
- クラウドおよびサービスの仮想化
- プロトコルの排除とネットワークの簡素化
- 同じ VPN を介した L2 サービスと L3 サービスの統合
- サービスとワークロードの柔軟な配置
- L2 および L3 VPN によるマルチテナント
- 最適な転送とワークロード モビリティ
- 高速コンバージェンス
- 効率的な帯域幅利用

EVPN の利点

EVPN には次の利点があります。

- 統合サービス : L2 および L3 VPN サービスの統合、拡張性と制御における L3VPN のよ うな原則と運用経験、ECMP を使用したオールアクティブマルチホーミングと PE ロードバ ランシング、複数の PE に対してマルチホームである CE との間で発着信するトラフィッ クのロードバランシングが可能。

- ネットワーク効率：フラッドと学習メカニズムの排除、デュアルホーム接続サーバへのリンクでの障害発生時の高速再ルーティング、復元力、および高速な再コンバージェンス、ブロードキャスト、不明ユニキャスト、マルチキャスト（BUM）トラフィック配信の最適化。
- サービスの柔軟性：MPLS データプレーンのカプセル化、既存および新しいサービスタイプのサポート（E-LAN、E-Line）、ピア PE 自動検出、および冗長グループ自動感知。

EVPN のモード

次の EVPN モードがサポートされています。

- シングルホーミング：これにより、カスタマー エッジ（CE）デバイスをプロバイダー エッジ（PE）デバイス 1 台に接続できます。
- マルチホーミング：これにより、カスタマーエッジ（CE）デバイスを複数のプロバイダーエッジ（PE）デバイスに接続できます。マルチホーミングにより、冗長接続が確保されます。冗長 PE デバイスは、ネットワーク障害が発生している場合にトラフィックが中断されないようにします。次にマルチホーミングのタイプを示します。
 - オールアクティブ：オールアクティブモードでは、特定のイーサネットセグメントに接続されているすべての PE が、そのイーサネットセグメントとの間で発着信するトラフィックを転送できます。

- [EVPN の概念 \(2 ページ\)](#)
- [EVPN 動作 \(3 ページ\)](#)
- [EVPN ルートタイプ \(5 ページ\)](#)
- [EVPN L2 ブリッジング サービスの設定 \(6 ページ\)](#)
- [EVPN ソフトウェア MAC ラーニング \(8 ページ\)](#)
- [EVPN アウト オブ サービス \(18 ページ\)](#)
- [EVPN 対応 CFM のサポート \(22 ページ\)](#)
- [イーサネットセグメント単位の EVPN 複数サービス \(22 ページ\)](#)
- [EVPN MPLS と VPLS のシームレスな統合 \(28 ページ\)](#)
- [既存の VPLS ネットワークでの EVPN の設定 \(30 ページ\)](#)
- [L2VPN ブリッジドメインでの EVI の設定 \(32 ページ\)](#)
- [EVPN 設定の確認 \(33 ページ\)](#)
- [エニーキャストゲートウェイ IRB の EVPN シングルアクティブ マルチホーミング \(37 ページ\)](#)
- [EVPN コア分離保護 \(40 ページ\)](#)
- [EVPN ルーティング ポリシー \(43 ページ\)](#)
- [BGP-LU アンダーレイを介した EVPN ブリッジングおよび VPWS サービス \(58 ページ\)](#)

EVPN の概念

EVPN 機能を実装するには、次の概念を理解する必要があります。

- **イーサネットセグメント (ES)** : イーサネットセグメントは、マルチホーム デバイスに接続する一連のイーサネット リンクです。マルチホーム デバイスまたはネットワークが 2 つ以上の PE に一連のイーサネット リンクを通じて接続されている場合に、その一連のリンクをイーサネット セグメントと呼びます。イーサネットセグメント ルートはルートタイプ 4 とも呼びます。このルートは、BUM トラフィックの指定フォワード (DF) の選択に使用されます。
- **イーサネットセグメント識別子 (ESI)** : イーサネットセグメントには一意の非ゼロの識別子が割り当てられます。これをイーサネットセグメント識別子 (ESI) と呼びます。ESI は、ネットワーク全体にわたってイーサネット セグメントを一意に表します。
- **EVI : EVPN インスタンス (EVI)** は仮想ネットワーク識別子 (VNI) で表されます。EVI は、PE ルータ上の VPN を表します。EVI は IP VPN ルーティングおよび転送 (VRF) と同じ役割を果たし、インポート/エクスポートルートターゲット (RT) が割り当てられません。ユーザ ネットワーク インターフェイス (UNI) でのサービス多重化動作に応じて、ポート上のすべてのトラフィック (すべて対 1 のバンドリング)、VLAN 上のトラフィック (1 対 1 のマッピング)、または VLAN のリスト/範囲のトラフィック (選択的バンドリング) をブリッジドメイン (BD) にマップできます。この BD は EVI に関連付けられ、MPLS コアに転送されます。
- **EAD/ES** : ES ごとのイーサネット自動検出ルートはルートタイプ 1 とも呼ばれます。このルートは、アクセス失敗のシナリオ時にトラフィックを早急に収束するために使用されます。このルートにはイーサネット タグ 0xFFFFFFFF が使用されます。
- **EAD/EVI** : EVI ごとのイーサネット自動検出ルートはルートタイプ 1 とも呼ばれます。このルートは、トラフィックはスイッチの 1 つにのみハッシュされる時のエイリアシングとロードバランシングに使用されます。EAD/ES ルートと区別するため、このルートにはイーサネット タグ値 0xFFFFFFFF を使用できません。
- **エイリアシング** : ルートタイプ 1 の EAD/EVI ルートを使用する所定のイーサネットセグメントで接続されているすべてのスイッチへのトラフィックのロードバランシングに使用されます。これはホストを実際に学習するスイッチとは関係なく実行されます。
- **大量撤回** : ルートタイプ 1 の EAD/ES ルートを使用し、アクセス障害シナリオ時に早急に収束するために使用されます。
- **DF の選択** : ループの転送を防ぐために使用されます。カプセル化を解除し、所定のイーサネットセグメントにトラフィックを転送するため、単一のルータのみを使用します。

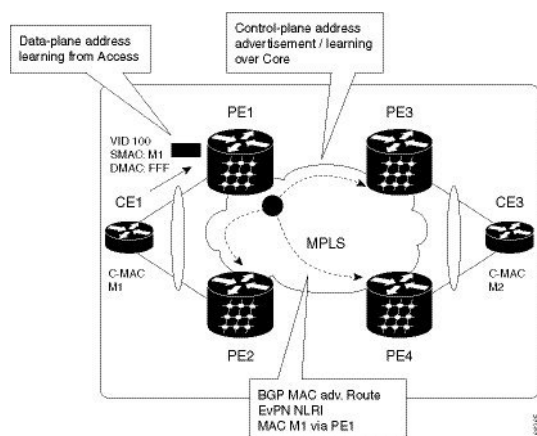
EVPN 動作

以下をアドバタイズするため、PE は起動時に EVPN ルートを交換します。

- **VPN メンバーシップ** : PE は所定のリモート PE のすべてのメンバーを検出します。マルチキャスト入力レプリケーションモデルの場合、EVI に関連付けられている PE フラッドリストの構築にこの情報が使用されます。MAC アドレスを学習した時点で BUM ラベルとユニキャスト ラベルが交換されます。

- **イーサネット セグメント到達可能性**：マルチホーミングのシナリオでは、PE がリモート PE と対応するそれらの冗長モード（オールアクティブまたはシングルアクティブ）を自動的に検出します。セグメント障害が発生した場合、PE はこの段階で使用していたルートを撤回し、リモート PE 上の MAC 大量撤回をシグナリングすることで高速コンバージェンスをトリガーします。
- **冗長グループメンバーシップ**：同じイーサネットセグメントに接続している（マルチホーミング）PE は互いに自動的に検出され、所定の EVI に対するブロードキャスト、不明ユニキャストおよびマルチキャスト（BUM）トラフィックの転送を担う指定フォワーダ（FD）を選択します。

図 1: EVPN 動作



EVPN はシングルホーミングモードまたはデュアルホーミングモードで動作できます。PE 上で EVPN が有効になっており、各 PE が所定の EVPN インスタンスの他のすべてのメンバー PE を検出したときにルートタイプ 3 がアドバタイズされるシングルホーミングのシナリオを考えてみます。不明ユニキャスト（または BUM）MAC を PE で受信すると、EVPN ルートタイプ 2 として他の PE にアドバタイズされます。MAC ルートは EVPN ルートタイプ 2 を使用して他の PE にアドバタイズされます。マルチホーミングのシナリオでは、ルートタイプ 1、3、および 4 がアドバタイズされ、他の PE とそれらの冗長モード（シングルアクティブまたはオールアクティブ）を検出します。ルートタイプ 1 を使用するのには、同じ CE をホストする他の PE を自動検出するためです。この他にも、このルートタイプは CE と PE 間の破損リンクから離れている高速ルートユニキャストトラフィックにも使用されます。ルートタイプ 4 は、指定フォワーダの選択に使用されます。たとえば、カスタマートラフィックが PE に着信し、ローカルイーサネットセグメント上で学習した各カスタマー MAC アドレスの到達可能性情報を EVPN MAC アドバタイズメントルートでコアを介して配布するトポロジを考えてみます。各 EVPN MAC ルートは、カスタマー MAC アドレスと、MAC を学習したポートに関連付けられたイーサネットセグメントおよびその関連付けられた MPLS ラベルをアナウンスします。この EVPN MPLS ラベルは、アドバタイズされた MAC アドレス宛にトラフィックを送信するときにリモート PE によって後で使用されます。

ESI ラベル割り当てによる動作の変更

RFC 7432 の推奨事項に準拠するため、MPLS ラベルの符号化や復号化が、拡張コミュニティで変更されました。これまでは、スプリット ホライズン グループ (SHG) ラベルを符号化するために、拡張コミュニティの下位 20 ビットが使用されていました。今回のリリースから、SHG ラベルの符号化では拡張コミュニティの上位 20 ビットが使用されるようになりました。

この変更により、新旧のソフトウェア リリース バージョンを実行している同じイーサネット セグメント内のルータは、拡張コミュニティを異なる方法で復号化します。この変更により、ピアリング EVPN PE ルータの SHG ラベルで不整合が発生します。ほとんどの場合、ルータは誤った SHG ラベルを持つ BUM パケットをドロップします。ただし、特定の状況では、リモート PE がこのようなパケットを受け入れて CE に転送し、ループが発生する可能性があります。このような状況が発生するのは、ラベルが誤って NULL と読み取られる場合です。

この問題を解決するには、次のことを行うことをお勧めします。

- 両方の PE が異なるソフトウェア リリース バージョンを実行している時間を最小限に抑えます。
- 新しいリリースにアップグレードする前に、アップグレードしたノードを分離し、対応する AC バンドルをシャットダウンします。
- 両方の PE を同じリリースにアップグレードした後、両方のサービスを稼働できます。

同様の推奨事項は、RFC 7432 に準拠していない SHG ラベル割り当てを持つ異なるベンダーとのピアリング PE に適用可能です。

EVPN ルートタイプ

EVPN ネットワーク層到達可能性情報 (NLRI) は、さまざまなルート タイプを提供します。

表 1: EVPN ルートタイプ

ルートタイプ	名前	使用法
1	イーサネット自動検出 (AD) ルート	ES ごとの少数ルートの送信、ES に属する EVI のリストの伝送
2	MAC/IP アドバタイズメント ルート	MAC のアドバタイズ、アドレス到達可能性、IP/MAC バインディングのアドバタイズ
3	包括的なマルチキャスト イーサネット タグ ルート	マルチキャスト トンネル エンドポイントの検出
4	イーサネットセグメント ルート	冗長グループの検出、DF の選択

ルート タイプ	名前	使用法
5	IP プレフィックスルート	IP プレフィックスのアドバタイズ

ルート タイプ 1: イーサネット自動検出 (AD) ルート

イーサネット自動検出 (AD) ルートは、EVI ごとと ESI ごとにアドバタイズされます。これらのルートは、ES ごとに送信されます。これらは ES に属している EVI のリストを伝送します。ESI フィールドは、CE がシングルホームの場合はゼロに設定されます。このルートタイプは、ロードバランシングのための MAC アドレスの大量撤回とエイリアシングに使用されます。

ルート タイプ 2: MAC/IP アドバタイズメント ルート

これらのルートは VLAN ごとのルートであるため、VNI に含まれている PE のみにこれらのルートが必要です。ホストの IP アドレスと MAC アドレスが NRLI 内のピアにアドバタイズされます。MAC アドレスのコントロールプレーン学習は不明ユニキャストのフラッドを削減します。

ルート タイプ 3: 包括的なマルチキャスト イーサネット タグ ルート

このルートは、送信元 PE からリモート PE へのブロードキャスト、不明ユニキャスト、およびマルチキャスト (BUM) トラフィック用の接続を確立します。このルートは、VLAN ごとと ESI ごとにアドバタイズされます。

ルート タイプ 4: イーサネット セグメント ルート

イーサネット セグメント ルートでは CE デバイスを 2 台のデバイスまたは PE デバイスを接続できます。ES ルートでは同じイーサネット セグメントに接続されている PE デバイスを検出できます。

ルート タイプ 5: IP プレフィックス ルート

IP プレフィックスが MAC アドバタイズメント ルートとは関係なくアドバタイズされます。EVPN IRB では、ホストルート /32 は RT-2 を使用してアドバタイズされ、サブネット /24 は RT-5 を使用してアドバタイズされます。



(注) EVPN IRB では、ホストルート /32 は RT-2 を使用してアドバタイズされ、サブネット /24 は RT-5 を使用してアドバタイズされます。

EVPN L2 ブリッジング サービスの設定

EVPN L2 ブリッジング サービスを設定するには、次のステップを実行します。



- (注) 必ず、ラベルモードをプレフィックス単位から VRF ラベルモード単位に変更してください。L2FIB および VPNv4 ルート（ラベル）は同じリソースを共有しているため、リソースを枯渇させると BVI の ping は失敗します。



- (注) EVPN または VPLS ブリッジで直接接続されたネイバーへのトラフィックは、次の場合は機能しません。
- ネイバーに暗黙的ヌルが設定されている場合。
 - インポジションノードにラベルの付いていない（プライマリ）パスとラベルの付いたパス（バックアップ）が混在している場合。



- (注) デバイスには最大 128K の MAC アドレス エントリを含めることができます。デバイス上のブリッジドメインには最大 64K の MAC アドレス エントリを含めることができます。



- (注) フラッドイングの無効化は、EVPN ブリッジドメインではサポートされていません。

```

/* Configure address family session in BGP */
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router#(config)# router bgp 200
RP/0/RSP0/CPU0:router#(config-bgp)# bgp router-id 209.165.200.227
RP/0/RSP0/CPU0:router#(config-bgp)# address-family l2vpn evpn
RP/0/RSP0/CPU0:router#(config-bgp)# neighbor 10.10.10.10
RP/0/RSP0/CPU0:router#(config-bgp-nbr)# remote-as 200
RP/0/RSP0/CPU0:router#(config-bgp-nbr)# description MPLSFACING-PEER
RP/0/RSP0/CPU0:router#(config-bgp-nbr)# update-source Loopback 0
RP/0/RSP0/CPU0:router#(config-bgp-nbr)# address-family l2vpn evpn

/* Configure EVI and define the corresponding BGP route targets */

Router# configure
Router(config)# evpn
Router(config-evpn)# evi 6005
Router(config-evpn-evi)# bgp
Router(config-evpn-evi-bgp)# rd 200:50
Router(config-evpn-evi-bgp)# route-target import 100:6005
Router(config-evpn-evi-bgp)# route-target export 100:6005
Router(config-evpn-evi-bgp)# exit
Router(config-evpn-evi)# advertise-mac

/* Configure a bridge domain */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# bridge group 1
Router(config-l2vpn-bg)# bridge-domain 1-1

```

```
Router(config-l2vpn-bg-bd) # interface GigabitEthernet
Router(config-l2vpn-bg-bd-ac) # evi 6005
Router(config-l2vpn-bg-bd-ac-evi) # commit
Router(config-l2vpnbg-bd-ac-evi) # exit
```

実行コンフィギュレーション

```
router bgp 200 bgp
router-id 209.165.200.227
address-family l2vpn evpn
neighbor 10.10.10.10
  remote-as 200 description MPLS-FACING-PEER
  updatesource Loopback0
  addressfamily l2vpn evpn
!

configure
evpn
evi 6005
  bgp
  rd 200:50
  route-target import 100:6005
  route-target export 100:6005
!
advertise-mac

configure
l2vpn
bridge group 1
  bridge-domain 1-1
  interface GigabitEthernet

  evi 6005
!
```

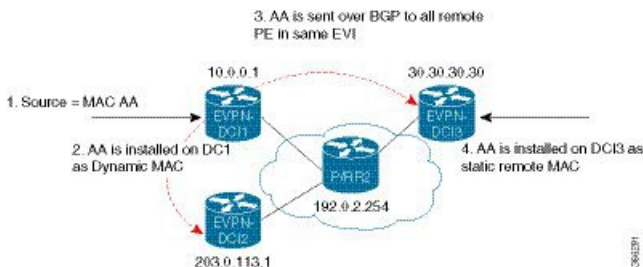
EVPN ソフトウェア MAC ラーニング

あるデバイス上で学習した MAC アドレスは、VLAN 内の別のデバイス上で学習されるか、配布されるようにする必要があります。EVPN ソフトウェア MAC ラーニング機能では、あるデバイス上で学習された MAC アドレスをネットワークに接続された別のデバイスに配布できません。MAC アドレスは、BGP を使用してリモート デバイスから学習されます。



(注) デバイスには最大 128K の MAC アドレス エントリを含めることができます。デバイス上のブリッジドメインには最大 64K の MAC アドレス エントリを含めることができます。

図 2: EVPN ソフトウェア MAC ラーニング



上の図は、ソフトウェア MAC ラーニングのプロセスを示しています。次に、このプロセスに関わるステップを示します。

1. トラフィックは、ブリッジドメイン内の 1 つのポートに着信します。
2. 送信元 MAC アドレス (AA) は PE 上で学習され、ダイナミック MAC エントリとして格納されます。
3. MAC アドレス (AA) がタイプ 2 BGP ルールに変換され、BGP を介して同じ EVI 内のすべてのリモート PE に送信されます。
4. MAC アドレス (AA) は、リモート MAC アドレスとして PE で更新されます。

EVPN ソフトウェア MAC ラーニングの設定

次の項では、EVPN ソフトウェア MAC ラーニングの設定方法について説明します。



(注) ルータは、フロー認識型トランスポート (FAT) 擬似回線をサポートしていません。

```

/* Configure bridge domain. */

RP/0/RSP0/CPU0:router(config)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group EVPN_SH
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain EVPN_2001
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface TenGigE
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# exit
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface BundleEther 20.2001
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# storm-control broadcast pps 10000 ← Enabling
storm-control is optional
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# exit
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# evi 2001
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-evi)# commit

/* Configure address family session in BGP. */

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# router bgp 200
RP/0/RSP0/CPU0:router(config-bgp)# bgp router-id 209.165.200.227
RP/0/RSP0/CPU0:router(config-bgp)# address-family l2vpn evpn

RP/0/RSP0/CPU0:router(config-bgp-nbr)# remote-as 200

```

```
RP/0/RSP0/CPU0:router(config-bgp-nbr)# description MPLSFACINGPEER
RP/0/RSP0/CPU0:router(config-bgp-nbr)# update-source Loopback 0
RP/0/RSP0/CPU0:router(config-bgp-nbr)# address-family l2vpn evpn
```

EVPN ソフトウェア MAC ラーニングでサポートされているモード

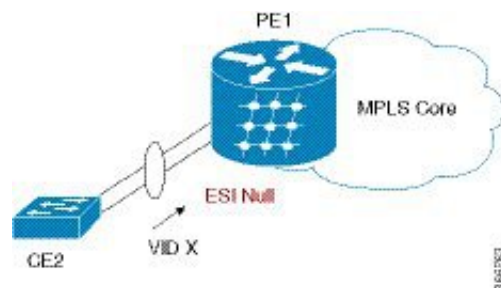
EVPN ソフトウェア MAC ラーニングでサポートされているモードは次のとおりです。

- シングル ホーム デバイス (SHD) またはシングル ホーム ネットワーク (SHN)
- デュアル ホーム デバイス (DHD) : オールアクティブ ロード バランシング

シングル ホーム デバイスまたはシングル ホーム ネットワーク モード

次の項では、EVPN ソフトウェア MAC ラーニング機能をシングル ホーム デバイスまたはシングル ホーム ネットワーク (SHD/SHN) モードで設定する方法について説明します。

図 3: シングル ホーム デバイスまたはシングル ホーム ネットワーク モード



上の図では、PE (PE1) はバンドルインターフェイスまたは物理インターフェイスを使用してイーサネットセグメントに接続されています。SHD/SHN にはヌルイーサネットセグメント識別子 (ESI) を使用します。

シングル ホーム デバイスまたはシングル ホーム ネットワーク モードでの EVPN の設定

この項では、シングル ホーム デバイスまたはシングル ホーム ネットワーク モードで EVPN ソフトウェア MAC ラーニング機能を設定する方法について説明します。

```
/* Configure bridge domain. */

RP/0/RSP0/CPU0:router(config)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group EVPN_ALL_ACTIVE
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain EVPN_2001
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface BundleEther1.2001
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# evi 2001

/* Configure advertisement of MAC routes. */

RP/0/RSP0/CPU0:router(config)# evpn
RP/0/RSP0/CPU0:router(config-evpn)# evi 2001
RP/0/RSP0/CPU0:router(config-evpn-evi)# advertise-mac
```

```

/* Configure address family session in BGP. */

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router#(config)# router bgp 200
RP/0/RSP0/CPU0:router#(config-bgp)# bgp router-id 09.165.200.227
RP/0/RSP0/CPU0:router#(config-bgp)# address-family l2vpn evpn
RP/0/RSP0/CPU0:router#(config-bgp)# neighbor 10.10.10.10
RP/0/RSP0/CPU0:router#(config-bgp-nbr)# remote-as 200
RP/0/RSP0/CPU0:router#(config-bgp-nbr)# description MPLSFACING-PEER
RP/0/RSP0/CPU0:router#(config-bgp-nbr)# update-source Loopback 0
RP/0/RSP0/CPU0:router#(config-bgp-nbr)# address-family l2vpn evpn

```

実行コンフィギュレーション

```

l2vpn
bridge group EVPN_ALL_ACTIVE
bridge-domain EVPN_2001
interface BundleEther1.2001
evi 2001
!
evpn
evi 2001
advertise-mac
!
router bgp 200 bgp
router-id 40.40.40.40
address-family l2vpn evpn
neighbor 10.10.10.10
remote-as 200 description MPLS-FACING-PEER
updatesource Loopback0
addressfamily l2vpn evpn

```

確認

シングルホームデバイスのEVPNを確認します。

```

RP/0/RSP0/CPU0:router# show evpn ethernet-segment interface Te0/4/0/10 detail

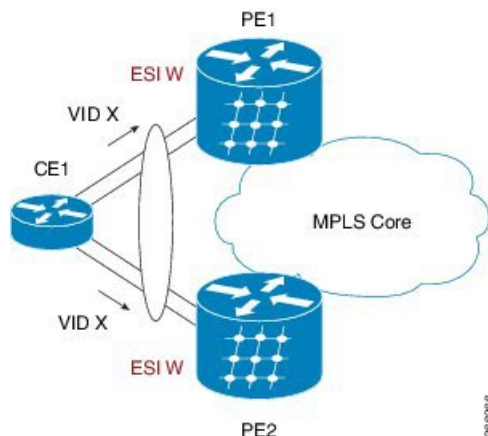
Ethernet Segment Id      Interface      Nexthops
-----
N/A                      Te0/4/0/10   20.20.20.20
.....
Topology :
Operational : SH
Configured : Single-active (AAPS) (default)

```

デュアルホームデバイス：オールアクティブロードバランシングモード

次の項では、デュアルホームデバイス（DHD）にオールアクティブロードバランシングモードでEVPNソフトウェアMACラーニング機能を設定する方法について説明します。

図 4: デュアルホーム デバイス : オールアクティブ ロード バランシング モード



オールアクティブ ロード バランシングはフローごとのアクティブ/アクティブ (AApF) と呼ばれています。上の図では、両方の EVPN PE に同一のイーサネット セグメント識別子を使用しています。PE は、バンドルインターフェイスを使用してイーサネットセグメントに接続されています。CE では、単一のバンドルが2つの EVPN PE に向けて設定されます。このモードでは、学習した MAC アドレスが PE1 と PE2 の両方に格納されます。PE1 と PE2 は両方とも同じ EVI 内でトラフィックを転送できます。

デュアルホーム デバイスでの EVPN ソフトウェア MAC ラーニングの設定 : オールアクティブモード

この項では、オールアクティブモードのデュアルホームデバイスで EVPN ソフトウェア MAC ラーニング機能を設定する方法について説明します。

```

/* Configure bridge domain. */

RP/0/RSP0/CPU0:router(config)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group EVPN_ALL_ACTIVE
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain EVPN_2001
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface BundleEther1.2001
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# evi 2001

/* Configure advertisement of MAC routes. */

RP/0/RSP0/CPU0:router(config)# evpn
RP/0/RSP0/CPU0:router(config-evpn)# evi 2001
RP/0/RSP0/CPU0:router(config-evpn-evi)# advertise-mac
RP/0/RSP0/CPU0:router(config-evpn-evi)# exit
RP/0/RSP0/CPU0:router(config-evpn)# interface bundle-ether1
RP/0/RSP0/CPU0:router(config-evpn-ac)# ethernet-segment
RP/0/RSP0/CPU0:router(config-evpn-ac-es)# identifier type 0 01.11.00.00.00.00.00.01

/* Configure address family session in BGP. */

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router#(config)# router bgp 200
RP/0/RSP0/CPU0:router#(config-bgp)# bgp router-id 209.165.200.227
RP/0/RSP0/CPU0:router#(config-bgp)# address-family l2vpn evpn
RP/0/RSP0/CPU0:router#(config-bgp)# neighbor 10.10.10.10

```

```

RP/0/RSP0/CPU0:router#(config-bgp-nbr)# remote-as 200
RP/0/RSP0/CPU0:router#(config-bgp-nbr)# description MPLSFACING-PEER
RP/0/RSP0/CPU0:router#(config-bgp-nbr)# update-source Loopback 0
RP/0/RSP0/CPU0:router#(config-bgp-nbr)# address-family l2vpn evpn

/* Configure Link Aggregation Control Protocol (LACP) bundle. */

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether1 300
RP/0/RSP0/CPU0:router(config-if)# lacp switchover suppress-flaps 300
RP/0/RSP0/CPU0:router(config-if)# exit

/* Configure VLAN Header Rewrite.*/

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface bundle-Ether1.2001 l2transport
RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1q 10
RP/0/RSP0/CPU0:router(config-if)# rewrite ingress tag pop 1 symmetric

```

実行コンフィギュレーション

```

l2vpn
bridge group EVPN_ALL_ACTIVE
bridge-domain EVPN_2001
interface Bundle-Ether1.2001
!
evi 2001
!
!
evpn
evi 2001
!
advertise-mac
!
interface bundle-ether1
ethernet-segment
identifier type 0 01.11.00.00.00.00.00.01
!
!
router bgp 200
bgp router-id 209.165.200.227
address-family l2vpn evpn
!
neighbor 10.10.10.10
remote-as 200
description MPLS-FACING-PEER
update-source Loopback0
address-family l2vpn evpn
!
interface Bundle-Ether1
lacp switchover suppress-flaps 300
load-interval 30
!
interface bundle-Ether1.2001 l2transport
encapsulation dot1q 2001
rewrite ingress tag pop 1 symmetric
!

```

確認

オールアクティブ モードのデュアルホーム デバイスの EVPN を確認します。

```

RP/0/RSP0/CPU0:router# show evpn ethernet-segment interface bundle-Ether 1 carvin$

Ethernet Segment Id      Interface  Nexthops
-----
0100.211b.fce5.df00.0b00  BE11      10.10.10.10
209.165.201.1
Topology :
Operational : MHN
Configured : All-active (AApF) (default)
Primary Services : Auto-selection
Secondary Services: Auto-selection
Service Carving Results:
Forwarders : 4003
Elected : 2002
EVI E : 2000, 2002, 36002, 36004, 36006, 36008
.....
Not Elected : 2001
EVI NE : 2001, 36001, 36003, 36005, 36007, 36009

MAC Flushing mode : Invalid

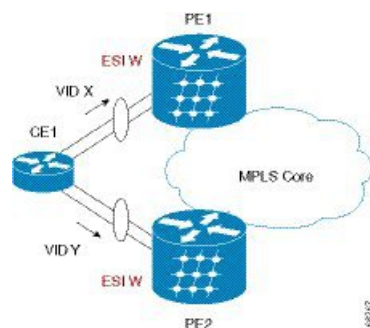
Peering timer : 3 sec [not running]
Recovery timer : 30 sec [not running]
Local SHG label : 34251
Remote SHG labels : 1
38216 : nexthop 209.165.201.1

```

デュアルホームデバイス：シングルアクティブロードバランシングモード

次の項では、デュアルホームデバイス（DHD）にシングルアクティブロードバランシングモードでEVPNソフトウェアMACラーニング機能を設定する方法について説明します。

図 5: デュアルホームデバイス：シングルアクティブロードバランシング



また、シングルアクティブロードバランシングは、サービスごとのオールアクティブ（AApS）とも呼ばれています。

両方のEVPN PEに同一のESIが設定されます。CEでは、2つのEVPN PEへの個別のバンドルまたは独立した物理インターフェイスが設定されます。このモードでは、学習したMACアドレスがPE1とPE2の両方に格納されます。所定の時間に1つのPEのみがEVI内にトラフィックを転送できます。

デュアルホーム デバイスでの EVPN ソフトウェア MAC ラーニングの設定 : シングルアクティブモード

この項では、シングルアクティブモードのデュアルホームで EVPN ソフトウェア MAC ラーニングを設定する方法について説明します。

```
/* Configure bridge domain. */

RP/0/RSP0/CPU0:router(config)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group EVPN_ALL_ACTIVE
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain EVPN_2001
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface BundleEther1.2001
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# evi 2001

/* Configure VLAN Header Rewrite (Single-tagged sub-interface).*/

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface bundle-Ether1.21 l2transport
RP/0/RSP0/CPU0:router(config-if)# lacp switchover suppress-flaps 300
RP/0/RSP0/CPU0:router(config-if)# exit
RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether1.2001 l2transport
RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1q 10
RP/0/RSP0/CPU0:router(config-if)# rewrite ingress tag pop 1 symmetric

/* Configure advertisement of MAC routes. */

RP/0/RSP0/CPU0:router(config)# evpn
RP/0/RSP0/CPU0:router(config-evpn)# evi 2001
RP/0/RSP0/CPU0:router(config-evpn-evi)# advertise-mac

/* Configure load balancing. */

RP/0/RSP0/CPU0:router(config)# evpn
RP/0/RSP0/CPU0:router(config-evpn)# evi 2001
RP/0/RSP0/CPU0:router(config-evpn-evi)# advertise-mac
RP/0/RSP0/CPU0:router(config-evpn-evi)# exit
RP/0/RSP0/CPU0:router(config-evpn)# interface bundle-ether1
RP/0/RSP0/CPU0:router(config-evpn)# ethernet-segment
RP/0/RSP0/CPU0:router(config-evpn-es)# load-balancing-mode single-active
RP/0/RSP0/CPU0:router(config-evpn-es)# identifier type 0 12.12.00.00.00.00.00.02
RP/0/RSP0/CPU0:router(config-evpn-es)# bgp route-target 1212.0000.0002

/* Configure address family session in BGP. */

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router#(config)# router bgp 200
RP/0/RSP0/CPU0:router#(config-bgp)# bgp router-id 209.165.200.227
RP/0/RSP0/CPU0:router#(config-bgp)# address-family l2vpn evpn
RP/0/RSP0/CPU0:router#(config-bgp)# neighbor 10.10.10.10
RP/0/RSP0/CPU0:router#(config-bgp-nbr)# remote-as 200
RP/0/RSP0/CPU0:router#(config-bgp-nbr)# description MPLSFACING-PEER
RP/0/RSP0/CPU0:router#(config-bgp-nbr)# update-source Loopback 0
RP/0/RSP0/CPU0:router#(config-bgp-nbr)# address-family l2vpn evpn
```

確認

シングルアクティブモードのデュアルホームデバイスの EVPN を確認します。

```
RP/0/RSP0/CPU0:router# show evpn ethernet-segment int bundleEther 21 carving detail

...
Ethernet Segment Id      Interface      Nexthops
-----
0012.1200.0000.0000.0002  BE21          10.10.10.10  209.165.201.1

ESI type : 0
Value : 12.1200.0000.0000.0002
ES Import RT : 1212.0000.0000 (from ESI)

Source MAC : 0000.0000.0000 (N/A)
Topology :
Operational : MHN
Configured : Single-active (AAPS)
Primary Services : Auto-selection
Secondary Services: Auto-selection

Service Carving Results:
Forwarders : 2
Elected : 1
EVI E : 500
Not Elected : 1
EVI NE : 501
```

EVPN ソフトウェア MAC ラーニングの確認

パケット ドロップ統計情報を確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain bd-name EVPN_2001 details

Bridge group: EVPN_ALL_ACTIVE, bridge-domain: EVPN_2001, id: 1110,
state: up, ShgId: 0, MSTi: 0
List of EVPNs:
EVPN, state: up
evi: 2001
XC ID 0x80000458
Statistics:
packets: received 28907734874 (unicast 9697466652), sent
76882059953
bytes: received 5550285095808 (unicast 1861913597184), sent
14799781851396
MAC move: 0
List of ACs:
AC: TenGigE, state is up
Type VLAN; Num Ranges: 1
...
Statistics:
packets: received 0 (multicast 0, broadcast 0, unknown
unicast 0, unicast 0), sent 45573594908
bytes: received 0 (multicast 0, broadcast 0, unknown unicast
0, unicast 0), sent 8750130222336
MAC move: 0
.....
```

VPN-ID と MAC アドレス フィルタを使用して EVPN ID を確認します。

```
RP/0/RSP0/CPU0:router# show evpn evi vpn-id 2001 neighbor

Neighbor IP      vpn-id
```



```

-----
209.165.200.225 2001
209.165.201.30 2001

```

BGP L2VPN EVPN の概要を確認します。

```
RP/0/RSP0/CPU0:router# show bgp l2vpn evpn summary
```

```

...
Neighbor      Spk   AS      MsgRcvd  MsgSent  TblVer   InQ   OutQ   Up/Down  St/PfxRcd
209.165.200.225 0     200     216739  229871  200781341 0     0     3d00h   348032
209.165.201.30 0     200     6462962 4208831 200781341 10    0     2d22h   35750

```

ラインカードの L2FIB テーブルへの MAC の更新を確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn mac mac all location 0/6/cPU0
```

```

Topo ID  Producer Next Hop(s)      Mac Address      IP Address
-----  -
1112    0/6/CPU0 Te 00a3.0001.0001

```

ルートスイッチプロセッサ (RSP) の L2FIB テーブルへの MAC の更新を確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn mac mac all location 0/6/cPU0
```

```

Topo ID  Producer Next Hop(s)      Mac Address      IP Address
-----  -
1112    0/6/CPU0 00a3.0001.0001

```

MAC アドレスの概要情報を確認します。

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding bridge-domain EVPN_ALL_ACTIVE:EVPN_2001
mac-address location 0/6/CPU0
```

```

.....
Mac Address      Type      Learned from/Filtered on  LC learned  Resync Age/Last Change
Mapped to
0000.2001.5555  dynamic  Te                        N/A         11 Jan 14:37:22
N/A <-- local dynamic
00bb.2001.0001  dynamic  Te                        N/A         11 Jan 14:37:22
N/A
0000.2001.1111  EVPN     BD id: 1110              N/A         N/A
N/A <-- remote static
00a9.2002.0001  EVPN     BD id: 1110              N/A         N/A
N/A

```

VPN-ID と MAC アドレス フィルタを使用して EVPN ID を確認します。

```
RP/0/RSP0/CPU0:router# show evpn evi vpn-id 2001 mac
```

```

EVI    MAC address      IP address      Nexthop      Label
----  -
2001   00a9.2002.0001  ::              10.10.10.10 34226      <-- Remote MAC
2001   00a9.2002.0001  ::              209.165.201.30 34202
2001   0000.2001.5555  20.1.5.55      TenGigE 34203  <-- local MAC

```

```
RP/0/RSP0/CPU0:router# RP/0/RSP0/CPU0:router# show evpn evi vpn-id 2001 mac 00a9.2002.0001
detail

EVI      MAC address      IP address  Nexthop      Label
----      -
2001     00a9.2002.0001   ::         10.10.10.10  34226

2001     00a9.2002.0001   ::         209.165.201.30  34202

Ethernet Tag : 0
Multi-paths Resolved : True <--- aliasing to two remote PE with All-Active load balancing

Static : No
Local Ethernet Segment : N/A
Remote Ethernet Segment : 0100.211b.fce5.df00.0b00
Local Sequence Number : N/A
Remote Sequence Number : 0
Local Encapsulation : N/A
Remote Encapsulation : MPLS
```

EVPNに関連付けられているBGPルートをブリッジドメインフィルタを使用して確認します。

```
RP/0/RSP0/CPU0:router# show bgp l2vpn evpn bridge-domain EVPN_2001 route-type 2

*> [2][0][48][00bb.2001.0001][0]/104
      0.0.0.0                0 i <----- locally learnt MAC
*>i [2][0][48][00a9.2002.00be][0]/104
      10.10.10.10 100      0 i <----- remotely learnt MAC
* i 209.165.201.30 100 0 i
```

EVPN アウトオブサービス

EVPN アウトオブサービス機能では、Link Aggregation Control Protocol (LACP) を設定したイーサネットセグメントに含まれているバンドルインターフェイスの状態を制御することができます。この機能を使用すると、ノードをアウトオブサービス (OOS) に移行させることができます。プロバイダーエッジ (PE) のすべてのバンドルを手動でシャットダウンする必要はありません。

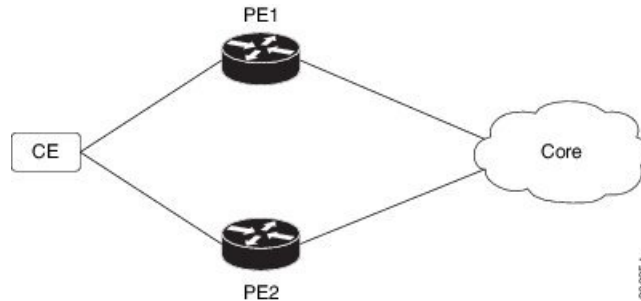
cost-out コマンドを使用してノード上のイーサネットVPN (EVPN) のイーサネットセグメントに属するすべてのバンドルインターフェイスをダウンさせます。イーサネット A-D のイーサネットセグメント (ES EAD) ルートは、バンドルをシャットダウンする前に撤回されます。PEは接続されているカスタマーエッジ (CE) デバイスにシグナリングし、対応するバンドルメンバーをダウンさせます。こうすることで、トラフィックを中断させることなく、トラフィックをこの PE ノードからそらしめます。CEからイーサネットセグメントへのトラフィックは、マルチホーミング環境内のピア PE へと方向付けられます。



(注) EVPN のコストアウトは、手動で設定された ESI でのみサポートされます。

次に、CE が PE1 と PE2 に接続されているトポロジを示します。PE1 に `cost-out` コマンドを設定すると、イーサネットセグメント上のすべてのバンドルインターフェイスがダウンします。また、CE 上の対応するバンドルメンバーもダウンします。したがって、このイーサネットセグメントのトラフィックは CE から PE2 へと送信されるようになります。

図 6: EVPN アウトオブサービス



ノードをサービス状態に戻すには、`no cost-out` コマンドを使用します。これにより、PE 上の EVPN イーサネットセグメントに属するすべてのバンドルインターフェイスと CE 上の対応するバンドルメンバーが起動します。

ノードがコストアウト状態にある場合に新しいバンドルイーサネットセグメントを追加するとそのバンドルがダウンします。同様に、バンドルイーサネットセグメントを削除するとそのバンドルは起動します。

リロード時に指定した時間が経過した後にノードをサービス状態に戻すには、`startup-cost-in` コマンドを使用します。EVPN が初期化された時点でノードがコストアウトになり、設定時間までコストアウト状態が維持されます。タイマー実行中に `evpn no startup-cost-in` コマンドを実行すると、タイマーが停止し、ノードがコストイン状態になります。

「`cost-out`」設定は「`startup-cost-in`」タイマーよりも常に優先されます。そのため、両方の設定でリロードすると、コストアウト状態は「`cost-out`」設定で制御されます。タイマーは関係ありません。同様に、起動タイマーでリロードし、タイマーが実行している間に「`cost-out`」を設定するとタイマーが停止し、OOS 状態は「`cost-out`」設定のみで制御されます。

`startup-cost-in timer` が実行している間に何らかのプロシージャを実行すると、ノードはコストアウト状態を維持し、タイマーが再起動します。

EVPN アウトオブサービスの設定

この項では、EVPN アウトオブサービスを設定する方法について説明します。

```

/* Configuring node cost-out on a PE */

Router# configure
Router(config)# evpn
Router(config-evpn)# cost-out
Router(config-evpn) commit

/* Bringing up the node into service */

Router# configure

```

```

Router(config)# evpn
Router(config-evpn)# no cost-out
Router(config-evpn) commit

/* Configuring the timer to bring up the node into service after the specified time on
reload */

Router# configure
Router(config)# evpn
Router(config-evpn)# startup-cost-in 6000
Router(config-evpn) commit

```

実行コンフィギュレーション

```

configure
evpn
 cost-out
!

configure
evpn
 startup-cost-in 6000
!

```

確認

EVPN アウト オブ サービスの設定を確認します。

```

/* Verify the node cost-out configuration */

Router# show evpn summary
Fri Apr 7 07:45:22.311 IST
Global Information
-----
Number of EVIs : 2
Number of Local EAD Entries : 0
Number of Remote EAD Entries : 0
Number of Local MAC Routes : 0
Number of Local MAC Routes : 5
      MAC : 5
      MAC-IPv4 : 0
      MAC-IPv6 : 0
Number of Local ES:Global MAC : 12
Number of Remote MAC Routes : 7
      MAC : 7
      MAC-IPv4 : 0
      MAC-IPv6 : 0
Number of Local IMCAST Routes : 56
Number of Remote IMCAST Routes: 56
Number of Internal Labels : 5
Number of ES Entries : 9
Number of Neighbor Entries : 1
EVPN Router ID : 192.168.0.1
BGP Router ID : ::
BGP ASN : 100
PBB BSA MAC address : 0207.1fee.be00
Global peering timer : 3 seconds
Global recovery timer : 30 seconds
EVPN cost-out : TRUE
      startup-cost-in timer : Not configured

```

```
/* Verify the no cost-out configuration */

Router# show evpn summary
Fri Apr 7 07:45:22.311 IST
Global Information
-----
Number of EVIs : 2
Number of Local EAD Entries : 0
Number of Remote EAD Entries : 0
Number of Local MAC Routes : 0
Number of Local MAC Routes : 5
      MAC : 5
      MAC-IPv4 : 0
      MAC-IPv6 : 0
Number of Local ES:Global MAC : 12
Number of Remote MAC Routes : 7
      MAC : 7
      MAC-IPv4 : 0
      MAC-IPv6 : 0
Number of Local IMCAST Routes : 56
Number of Remote IMCAST Routes: 56
Number of Internal Labels : 5
Number of ES Entries : 9
Number of Neighbor Entries : 1
EVPN Router ID : 192.168.0.1
BGP Router ID : ::
BGP ASN : 100
PBB BSA MAC address : 0207.1fee.be00
Global peering timer : 3 seconds
Global recovery timer : 30 seconds
EVPN cost-out : FALSE
      startup-cost-in timer : Not configured

/* Verify the startup-cost-in timer configuration */

Router# show evpn summary
Fri Apr 7 07:45:22.311 IST
Global Information
-----
Number of EVIs : 2
Number of Local EAD Entries : 0
Number of Remote EAD Entries : 0
Number of Local MAC Routes : 0
Number of Local MAC Routes : 5
      MAC : 5
      MAC-IPv4 : 0
      MAC-IPv6 : 0
Number of Local ES:Global MAC : 12
Number of Remote MAC Routes : 7
      MAC : 7
      MAC-IPv4 : 0
      MAC-IPv6 : 0
Number of Local IMCAST Routes : 56
Number of Remote IMCAST Routes: 56
Number of Internal Labels : 5
Number of ES Entries : 9
Number of Neighbor Entries : 1
EVPN Router ID : 192.168.0.1
BGP Router ID : ::
BGP ASN : 100
PBB BSA MAC address : 0207.1fee.be00
```

```

Global peering timer      :      3 seconds
Global recovery timer    :      30 seconds
EVPN node cost-out       :      TRUE
startup-cost-in timer   :      6000

```

EVPN 対応 CFM のサポート

イーサネット接続障害管理（CFM）はサービス レベル OAM プロトコルの 1 つで、VLAN ごとにエンドツーエンドのイーサネットサービスをモニタリングおよびトラブルシューティングするためのツールとなります。これには、予防的な接続モニタリング、障害検証、および障害分離の機能が含まれています。CFM は EVPN ネットワークに導入できます。EVPN ネットワークで CFM を使用して、ノード間の接続をモニタできます。

制約事項

EVPN 対応 CFM は、次の制限の下でサポートされています。

- アクティブ-アクティブ マルチホーミングのシナリオでは、マルチホーム CE デバイスとそれらに接続している PE デバイスとの間の接続をモニタする場合、CFM は CE と PE 間の個別のリンク間でのみ使用できます。CE デバイスと PE デバイス間のバンドルで CFM の使用を試みると、シーケンス番号エラーが発生し、統計情報が不正確になります。
- ループバックおよびリンクトレースの結果に副作用が生じる可能性があります。ループバックまたはリンクトレースのいずれかで同じインスタンスに対して複数の結果が報告されたり、同じ 2 つのエンドポイント間にあるループバックとリンクトレースの連続するインスタンスで異なる結果が生じたりする場合があります。

イーサネット セグメント単位の EVPN 複数サービス

イーサネット セグメント単位の EVPN 複数サービス機能を使用すると、単一のイーサネット セグメント（ES）で複数のサービスを設定できます。複数の ES で複数のサービスを設定する代わりに、1 つの ES で複数のサービスを設定できます。

単一のイーサネット バンドルで次のサービスを設定できます。サブインターフェイスごとにサービスを 1 つずつ設定できます。

- フレキシブル クロスコネク ト（FXC） サービス。VLAN 非認識型、VLAN 認識型、およびローカル スイッチング モードをサポートしています。
詳細については、『*L2VPN and Ethernet Services Configuration Guide for Cisco NCS 540 Series Routers*』の「*Configure Point-to-Point Layer 2 Services*」の章を参照してください。
- EVPN-VPWS Xconnect サービス
詳細については、『*L2VPN and Ethernet Services Configuration Guide for Cisco NCS 540 Series Routers*』の「*EVPN Virtual Private Wire Service (VPWS)*」の章を参照してください。
- EVPN Integrated Routing and Bridging（IRB）

詳細については、『*L2VPN and Ethernet Services Configuration Guide for Cisco NCS 540 Series Routers*』の「*Configure EVPN IRB*」の章を参照してください。

- ネイティブ EVPN

詳細については、『*L2VPN and Ethernet Services Configuration Guide for Cisco NCS 540 Series Routers*』の「*EVPN Features*」の章を参照してください。

これらのサービスはすべて、オールアクティブのマルチホーミングのシナリオでのみサポートされます。

イーサネット セグメント単位の EVPN 複数サービスの設定

イーサネット バンドルインターフェイス 22001 を介して 2 つのプロバイダー エッジ (PE) デバイスに接続しているカスタマー エッジ (CE) デバイスを考えてみます。バンドル イーサネット サブインターフェイスで複数のサービスを設定します。

設定例

Bundle-Ether22001 ES を考慮し、サブインターフェイスで複数のサービスを設定します。

```
/* Configure attachment circuits */
Router# configure
Router(config)# interface Bundle-Ether22001.12 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q 12
Router(config-l2vpn-subif)# exit
Router(config-l2vpn)# exit
Router(config)# interface Bundle-Ether22001.13 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q 13
Router(config-l2vpn-subif)# exit
Router(config-l2vpn)# exit
Router(config)# interface Bundle-Ether22001.14 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q 14
Router(config-l2vpn-subif)# exit
Router(config-l2vpn)# exit
Router(config)# interface Bundle-Ether22001.1 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q 1
Router(config-l2vpn-subif)# exit
Router(config-l2vpn)# exit
Router(config)# interface Bundle-Ether22001.2 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q 2
Router(config-l2vpn-subif)# exit
Router(config-l2vpn)# exit
Router(config)# interface Bundle-Ether22001.3 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q 3
Router(config-l2vpn-subif)# exit
Router(config-l2vpn)# exit
Router(config)# interface Bundle-Ether22001.4 l2transport
Router(config-l2vpn-subif)# encapsulation dot1q 1 second-dot1q 4
Router(config-l2vpn-subif)# exit
Router(config-l2vpn)# exit

/*Configure VLAN Unaware FXC Service */
Router(config)# l2vpn
Router(config-l2vpn)# flexible-xconnect-service vlan-unaware fxc_mh1
Router(config-l2vpn-fxs-vu)# interface Bundle-Ether22001.1
```

```

Router(config-l2vpn-fxs-vu) # interface Bundle-Ether22001.2
Router(config-l2vpn-fxs-vu) # interface Bundle-Ether22001.3
Router(config-l2vpn-fxs-vu) # neighbor evpn evi 21006 target 22016
Router(config-l2vpn-fxs-vu) # commit

/* Configure VLAN Aware FXC Service */
Router(config) # l2vpn
Router(config-l2vpn) # flexible-xconnect-service vlan-aware evi 24001
Router(config-l2vpn-fxs-vu) # interface Bundle-Ether22001.12
Router(config-l2vpn-fxs-vu) # interface Bundle-Ether22001.13
Router(config-l2vpn-fxs-vu) # interface Bundle-Ether22001.14
Router(config-l2vpn-fxs-vu) # commit

/* Configure Local Switching - Local switching is supported only on VLAN-aware FXC */
PE1
Router# configure
Router(config) # l2vpn
Router(config-l2vpn) # flexible-xconnect-service vlan-aware evi 31400
Router(config-l2vpn-fxs) # interface Bundle-Ether22001.1400
Router(config-l2vpn-fxs) # interface Bundle-Ether23001.1400
Router(config-l2vpn-fxs) # commit
Router(config-l2vpn-fxs) # exit
PE2
Router# configure
Router(config) # l2vpn
Router(config-l2vpn) # flexible-xconnect-service vlan-aware evi 31401
Router(config-l2vpn-fxs) # interface Bundle-Ether22001.1401
Router(config-l2vpn-fxs) # interface Bundle-Ether23001.1401
Router(config-l2vpn-fxs) # commit
Router(config-l2vpn-fxs) # exit

/* Configure EVPN-VPWS xconnect service and native EVPN with IRB */

Router# configure
Router(config) # interface Bundle-Ether22001.11 l2transport
Router(config-l2vpn-subif) # encapsulation dot1q 1 second-dot1q 11
Router(config-l2vpn-subif) # rewrite ingress tag pop 2 symmetric
Router(config-l2vpn-subif) # commit
Router(config-l2vpn-subif) # exit

Router# configure
Router(config) # interface Bundle-Ether22001.21 l2transport
Router(config-l2vpn-subif) # encapsulation dot1q 1 second-dot1q 21
Router(config-l2vpn-subif) # rewrite ingress tag pop 2 symmetric
Router(config-l2vpn-subif) # commit
Router(config-l2vpn-subif) # exit

Router# configure
Route(config) # l2vpn
Router(config-l2vpn) # xconnect group xg22001
Router(config-l2vpn-xc) # p2p evpn-vpws-mclag-22001
Router(config-l2vpn-xc-p2p) # interface Bundle-Ether22001.11
Router(config-l2vpn-xc-p2p) # neighbor evpn evi 22101 target 220101 source 220301
Router(config-l2vpn-xc-p2p) # commit
Router(config-l2vpn-xc-p2p) # exit

Router # configure
Router (config) # l2vpn
Router (config-l2vpn) # bridge group native_evpn1
Router (config-l2vpn-bg) # bridge-domain bd21
Router (config-l2vpn-bg-bd) # interface Bundle-Ether22001.21
Router (config-l2vpn-bg-bd-ac) # routed interface BVI21
Router (config-l2vpn-bg-bd-bvi) # evi 22021

```



```

Router (config-l2vpn-bg-bd-bvi)# commit
Router (config-l2vpn-bg-bd-bvi)# exit

/* Configure Native EVPN */
Router # configure
Router (config)# evpn
Router (config-evpn)# interface Bundle-Ether22001
Router (config-evpn-ac)# ethernet-segment identifier type 0 ff.ff.ff.ff.ff.ff.ff.00
Router (config-evpn-ac-es)# bgp route-target 2200.0001.0001
Router (config-evpn-ac-es)# exit
Router (config-evpn)# evi 24001
Router (config-evpn-evi)# bgp
Router (config-evpn-evi-bgp)# route-target import 64:24001
Router (config-evpn-evi-bgp)# route-target export 64:24001
Router (config-evpn-evi-bgp)# exit
Router (config-evpn-evi)# exit
Router (config-evpn)# evi 21006
Router (config-evpn-evi)# bgp
Router (config-evpn-evi-bgp)# route-target route-target 64:10000
Router (config-evpn-evi-bgp)# exit
Router (config-evpn-evi)# exit
Router (config-evpn)# evi 22101
Router (config-evpn-evi)# bgp
Router (config-evpn-evi-bgp)# route-target import 64:22101
Router (config-evpn-evi-bgp)# route-target export 64:22101
Router (config-evpn-evi-bgp)# exit
Router (config-evpn-evi)# exit
Router (config-evpn)# evi 22021
Router (config-evpn-evi)# bgp
Router (config-evpn-evi-bgp)# route-target import 64: 22021
Router (config-evpn-evi-bgp)# route-target export 64: 22021
Router (config-evpn-evi-bgp)# exit
Router (config-evpn-evi)# exit
Router (config-evpn-evi)# advertise-mac
Router (config-evpn-evi)# exit
Router (config-evpn)# evi 22022
Router (config-evpn-evi)# bgp
Router (config-evpn-evi-bgp)# route-target import 64: 22022
Router (config-evpn-evi-bgp)# route-target export 64: 22022
Router (config-evpn-evi-bgp)# exit
Router (config-evpn-evi)# advertise-mac
Router (config-evpn-evi)# commit
Router (config-evpn-evi)# exit

```

実行コンフィギュレーション

```

/* Configure attachment circuits */
interface Bundle-Ether22001.12 l2transport
encapsulation dot1q 1 second-dot1q 12
!
interface Bundle-Ether22001.13 l2transport
encapsulation dot1q 1 second-dot1q 13
!
interface Bundle-Ether22001.14 l2transport
encapsulation dot1q 1 second-dot1q 14
!
interface Bundle-Ether22001.1 l2transport
encapsulation dot1q 1 second-dot1q 1
!
interface Bundle-Ether22001.2 l2transport

```

```

encapsulation dot1q 1 second-dot1q 2
!
interface Bundle-Ether22001.3 l2transport
encapsulation dot1q 1 second-dot1q 3
!
interface Bundle-Ether22001.4 l2transport
encapsulation dot1q 1 second-dot1q 4

/*Configure VLAN Unaware FXC Service */
flexible-xconnect-service vlan-unaware fxc_mh1
    interface Bundle-Ether22001.1
    interface Bundle-Ether22001.2
    interface Bundle-Ether22001.3
neighbor evpn evi 21006 target 22016
!
/*Configure VLAN Aware FXC Service */
l2vpn
    flexible-xconnect-service vlan-aware evi 24001
        interface Bundle-Ether22001.12
        interface Bundle-Ether22001.13
        interface Bundle-Ether22001.14

/* Configure Local Switching */
flexible-xconnect-service vlan-aware evi 31400
    interface Bundle-Ether22001.1400
    interface Bundle-Ether23001.1400
!
flexible-xconnect-service vlan-aware evi 31401
    interface Bundle-Ether22001.1401
    interface Bundle-Ether23001.1401
!

/* Configure EVPN-VPWS xconnect service and native EVPN with IRB */
interface Bundle-Ether22001.11 l2transport
    encapsulation dot1q 1 second-dot1q 11
    rewrite ingress tag pop 2 symmetric
!
interface Bundle-Ether22001.21 l2transport
    encapsulation dot1q 1 second-dot1q 21
    rewrite ingress tag pop 2 symmetric
!
!
l2vpn
xconnect group xg22001
p2p evpn-vpws-mclag-22001
    interface Bundle-Ether22001.11
neighbor evpn evi 22101 target 220101 source 220301
!
bridge group native_evpn1
    bridge-domain bd21
    interface Bundle-Ether22001.21
        routed interface BVI21
        evi 22021
!
/* Configure Native EVPN */
Evpn
    interface Bundle-Ether22001
        ethernet-segment identifier type 0 ff.ff.ff.ff.ff.ff.ff.ff.0e
        bgp route-target 2200.0001.0001
!
    evi 24001
        bgp
            route-target import 64:24001
            route-target export 64:24001

```

```

!
evi 21006
  bgp
    route-target 64:100006
!
evi 22101
  bgp
    route-target import 64:22101
    route-target export 64:22101
!
evi 22021
  bgp
    route-target import 64:22021
    route-target export 64:22021
!
  advertise-mac
!
evi 22022
  bgp
    route-target import 64:22022
    route-target export 64:22022
!
  advertise-mac
!

```

確認

各サービスがサブインターフェイスで設定されているかどうかを確認します。

```

Router# show l2vpn xconnect summary
Number of groups: 6
Number of xconnects: 505 Up: 505 Down: 0 Unresolved: 0 Partially-programmed: 0
AC-PW: 505 AC-AC: 0 PW-PW: 0 Monitor-Session-PW: 0
Number of Admin Down segments: 0
Number of MP2MP xconnects: 0
  Up 0 Down 0
Advertised: 0 Non-Advertised: 0

```

```

Router# show l2vpn xconnect-service summary
Number of flexible xconnect services: 74
Up: 74

```

```

Router# show l2vpn flexible-xconnect-service name fxc_mh1
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
SB = Standby, SR = Standby Ready, (PP) = Partially Programmed
Flexible XConnect Service Segment
Name      ST  Type  Description  ST
-----
fxc_mh1  UP  AC:   BE22001.1   UP
          AC:   BE22001.2   UP
          AC:   BE22001.3   UP
-----

```

```

Router# show l2vpn flexible-xconnect-service evi 24001
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
SB = Standby, SR = Standby Ready, (PP) = Partially Programmed
Flexible XConnect Service Segment
Name      ST  Type  Description  ST
-----
evi:24001 UP  AC:   BE22001.11  UP
          AC:   BE22001.12  UP
          AC:   BE22001.13  UP
-----

```

```

AC:   BE22001.14   UP
-----

Router# show l2vpn xconnect group xg22001 xc-name evpn-vpws-mclag-22001
Fri Sep 1 17:28:58.259 UTC
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
SB = Standby, SR = Standby Ready, (PP) = Partially Programmed
XConnect
Group      Name                               ST      Description ST      Segment 2
          Name                               ST      Description ST      Description
-----
          ST
-----
xg22001    evpn-vpws-mclag-22001             UP      BE22001.101 UP      EVPN 22101, 220101,64.1.1.6
          UP
-----

```

関連コマンド

- evpn
- evi
- ethernet-segment
- advertise-mac
- show evpn ethernet-segment
- show evpn evi
- show evpn summary
- show l2vpn xconnect summary
- show l2vpn flexible-xconnect-service
- show l2vpn xconnect group

EVPN MPLS と VPLS のシームレスな統合

EVPN MPLS と VPLS のシームレスな統合により、同じ VPN インスタンスに対して EVPN と VPLS を実行する PE ノードの共存が可能になります。VPLS またはレガシー ネットワークを、サービスの中断なしで次世代の EVPN ネットワークにアップグレードできます。選択したすべての VPLS プロバイダーエッジ (PE) ノードに、EVPN サービスを同時に導入できます。ただし、トラフィックの中断を回避するため、既存の VPLS 対応 PE で EVPN サービスを 1 つずつプロビジョニングします。

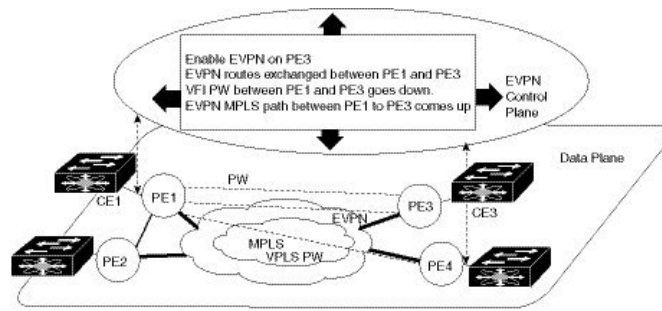
シームレスな統合による VPLS ネットワークの EVPN ネットワークへの移行

EVPN ネットワークでは、VPN インスタンスは EVPN インスタンス ID (EVI) によって識別されます。他の L2VPN テクノロジーと同様に、EVPN インスタンスもルートターゲットおよび

ルート識別子に関連付けられています。MAC をデータプレーンで学習する（「フラッディングと学習の技術」を使用して学習する）従来の VPLS とは異なり、EVPN ではコントロールプレーンを使用して MAC を学習し伝播します。EVPN では、MAC ルートは MP BGP プロトコルによって伝送されます。EVPN 対応 PE では、PE のルートターゲット（RT）が一致した場合にのみ、PE が MAC ルートをラベルとともにそれぞれの EVPN 転送テーブルにインポートします。EVPN PE ルータは、同じ VPN インスタンスで VPLS および EVPN L2 ブリッジングを実行する機能を備えています。EVPN と BGP-AD PW の両方が VPN インスタンスで設定されている場合、EVPN PE は、BGP VPLS 自動検出（AD）ルートと、BGP EVPN 包括マルチキャストルート（タイプ 3）を、特定の VPN インスタンスにアダプタイズします。ルートタイプ 3 は入力複製マルチキャストルートと呼ばれ、ブロードキャスト、未知のユニキャスト、およびマルチキャスト（BUM）トラフィックの送信に使用されます。その他のリモート PE は、送信側の PE RT が設定済みの RT と一致する場合にのみ、同じ VPN インスタンスに対してタイプ 3 ルートをインポートします。したがって、これらのルート交換の最後に、EVPN 対応 PE は、VPN インスタンスにある他のすべての PE とそれらの関連機能を検出します。PE が自身の BUM トラフィックを他の PE に送信するために使用するタイプ 3 ルートでは、同じ RT を持つ PE が BUM トラフィックを受信することが保証されます。EVPN は、タイプ 2 ルートを使用してカスタマー MAC アドレスをアダプタイズします。

EVPN MPLS と VPLS のシームレスな統合により、ネットワーク サービスを中断することなく、VPLS PE ルータを EVPN に 1 つずつアップグレードすることができます。PE1、PE2、PE3、および PE4 が VPLS PW を使用してフルメッシュ ネットワークで相互接続されている次のトポロジを考えてみます。

図 7: EVPN MPLS と VPLS のシームレスな統合



EVPN サービスは、一度に 1 つの PE ノードずつ、ネットワークに導入できます。VPLS サービスの VPN インスタンスで EVPN を有効にすることによって、VPLS から EVPN への移行が PE1 で開始されます。EVPN が有効になるとすぐに、PE1 は他の PE ノードへの EVPN 包括マルチキャストルートのアダプタイズを開始します。PE1 は他の PE ノードからの包含マルチキャストルートを受信しないため、PE1 と他の PE ノード間の VPLS 擬似回線はアクティブなままです。PE1 は、VPLS 擬似回線を使用してトラフィックの転送を維持します。同時に、PE1 は EVPN ルートタイプ 2 を使用して CE1 から学習したすべての MAC アドレスをアダプタイズします。2 番目のステップでは、EVPN が PE3 で有効になっています。PE3 は、他の PE ノードへの包含マルチキャストルートのアダプタイズを開始します。PE1 と PE3 の両方が EVPN ルートを介して互いを検出します。その結果、PE1 と PE3 は両者間の擬似回線をシャットダウンします。EVPN サービスが、PE1 と PE3 の間で VPLS サービスの代わりとなります。この段階では、PE1 は PE2 と PE4 を使用して VPLS サービスを実行し続け、同じ VPN インスタンスで

PE3 を使用して EVPN サービスを開始します。このことを、EVPN と VPLS のシームレスな統合と呼びます。VPLS から EVPN への移行は残りの PE ノードに対して続けられます。最終的に、4 つすべての PE ノードが EVPN サービスで有効になります。VPLS サービスがネットワーク内の EVPN サービスに完全に置き換えられます。すべての VPLS 擬似回線がシャットダウンされます。

既存の VPLS ネットワークでの EVPN の設定

既存の VPLS ネットワークで EVPN を設定するには、次の作業を実行します。

- L2VPN EVPN アドレスファミリの設定
- EVPN コンフィギュレーションモードで、EVI と対応する BGP ルートターゲットを設定します。
- ブリッジドメインでの EVI の設定

さまざまな VPLS ベース ネットワークを EVPN に移行する方法については、[L2VPN ブリッジドメインでの EVI の設定 \(32 ページ\)](#) を参照してください。

L2 EVPN アドレスファミリの設定

BGP と参加ネイバーの両方で EVPN アドレス ファミリを有効にするには、次の作業を実行します。

設定例

```
Router# configure
Router(config)#router bgp 65530
Router(config-bgp)#nsr
Router(config-bgp)#bgp graceful-restart
Router(config-bgp)#bgp router-id 200.0.1.1
Router(config-bgp)#address-family l2vpn evpn
Router(config-bgp-af)#exit
Router(config-bgp)#neighbor 200.0.4.1
Router(config-bgp-nbr)#remote-as 65530
Router(config-bgp-nbr)#update-source Loopback0
Router(config-bgp-nbr)#address-family l2vpn evpn
Router(config-bgp-nbr-af)#commit
```

実行コンフィギュレーション

```
configure
router bgp 65530
  nsr
  bgp graceful-restart
  bgp router-id 200.0.1.1
  address-family l2vpn evpn
  !
  neighbor 200.0.4.1
    remote-as 65530
```

```
update-source Loopback0
address-family l2vpn evpn
!
```

EVPN コンフィギュレーションモードでの EVI と対応する BGP ルートターゲットの設定

EVI を設定し、対応する BGP ルート ターゲットを定義するには、次の作業を実行します。また、`advertise-mac` を設定します。設定しないと MAC ルート（タイプ 2）がアドバタイズされません。

設定例

```
Router# configure
Router (config) #evpn
Router (config-evpn) #evi i
Router (config-evpn-evi-bgp) #bgp
Router (config-evpn-evi-bgp) #table-policy spp-basic-6
Router (config-evpn-evi-bgp) #route-target import 100:6005
Router (config-evpn-evi-bgp) #route-target export 100:6005
Router (config-evpn-evi-bgp) #exit
Router (config-evpn-evi) #advertise-mac
Router (config-evpn-evi) #commit
```

実行コンフィギュレーション

```
configure
evpn
evi
  bgp
    table-policy spp-basic-6
    route-target import 100:6005
    route-target export 100:6005
  !
  advertise-mac
  !
!
```

ブリッジ ドメインでの EVI の設定

対応する L2VPN ブリッジ ドメインで EVI を設定するには、次の作業を実行します。

設定例

```
Router# configure
Router (config) #l2vpn
Router (config-l2vpn) #bridge group bg1
Router (config-l2vpn-bg) #bridge-domain bd1
Router (config-l2vpn-bg-bd) #interface GigabitEthernet
Router (config-l2vpn-bg-bd-ac) #exit
```

```

Router(config-l2vpn-bg-bd) #evi 1
Router(config-l2vpn-bg-bd-evi) #exit
Router(config-l2vpn-bg-bd) #vfi v1
Router(config-l2vpn-bg-bd-vfi) #neighbor 10.1.1.2 pw-id 1000
Router(config-l2vpn-bg-bd-vfi-pw) #mpls static label local 20001 remote 10001
Router(config-l2vpn-bg-bd-vfi-pw) #commit

```

実行コンフィギュレーション

```

configure
l2vpn
bridge group bg1
bridge-domain bd1
interface GigabitEthernet
!
evi 1
!
vfi v1
neighbor 10.1.1.2 pw-id 1000
mpls static label local 20001 remote 10001
!
!
evi 1
!

```

L2VPN ブリッジドメインでの EVI の設定

次の例は、さまざまな VPLS ベース ネットワークの L2VPN ブリッジドメインでの EVI 設定を示しています。

MPLS スタティック ラベルをベースとする VPLS

```

l2vpn
bridge group bg1
bridge-domain bd-1-1
interface GigabitEthernet
!
vfi vfi-1-1
neighbor 200.0.2.1 pw-id 1200001
mpls static label local 20001 remote 10001
!
neighbor 200.0.3.1 pw-id 1300001
mpls static label local 30001 remote 10001
!
neighbor 200.0.4.1 pw-id 1400001
mpls static label local 40001 remote 10001
!
!
evi 1
!

```

自動検出 BGP および BGP シグナリングをベースとする VPLS

```

l2vpn
bridge group bg1

```



```

bridge-domain bd-1-2
  interface GigabitEthernet
  !
  vfi vfi-1-2
    vpn-id 2
    autodiscovery bgp
    rd 101:2
    route-target 65530:200
    signaling-protocol bgp
    ve-id 11
    ve-range 16
  !
  !
  evi 2
  !

```

ターゲット LDP をベースとする VPLS

```

bridge-domain bd-1-4
  interface GigabitEthernet
  !
  vfi vfi-1-4
    neighbor 200.0.2.1 pw-id 1200004
    !
    neighbor 200.0.3.1 pw-id 1300004
    !
    neighbor 200.0.4.1 pw-id 1400004
    !
  evi 3
  !

```

EVPN 設定の確認

EVPN の設定と MAC のアドバタイズメントを確認するには、次のコマンドを使用します。EVPN のステータス、AC のステータス、および VFI のステータスを確認します。

- show l2vpn bridge-domain
- show evpn summary
- show bgp rt l2vpn evpn
- show evpn evi
- show l2route evpn mac all

```

Router#show l2vpn bridge-domain bd-name bd-1-1
Mon Feb 20 21:03:40.244 EST
Legend: pp = Partially Programmed.
Bridge group: bg1, bridge-domain: bd-1-1, id: 0, state: up, ShgId: 0, MSTi: 0
Aging: 300 s, MAC limit: 4000, Action: none, Notification: syslog
Filter MAC addresses: 0
ACs: 1 (1 up), VFIs: 1, PWs: 3 (2 up), PBBs: 0 (0 up), VNIs: 0 (0 up)
List of EVPNs:
  EVPN, state: up
List of ACs:
  Gi0/2/0/0.1, state: up, Static MAC addresses: 0, MSTi: 2
List of Access PWs:

```

```

List of VFIs:
VFI vfi-1-1 (up)
  Neighbor 200.0.2.1 pw-id 1200001, state: up, Static MAC addresses: 0
  Neighbor 200.0.3.1 pw-id 1300001, state: down, Static MAC addresses: 0
  Neighbor 200.0.4.1 pw-id 1400001, state: up, Static MAC addresses: 0
List of Access VFIs:
When PEs are evpn enabled, pseudowires that are associated with that BD will be brought
down. The VPLS BD pseudowires are always up.

```

EVI の設定済みのローカルおよびリモート MAC ルートのうちアドバタイズされたものの数を確認します。

```

Router#show evpn summary
Mon Feb 20 21:05:16.755 EST
-----
Global Information
-----
Number of EVIs                : 6
Number of Local EAD Entries   : 0
Number of Remote EAD Entries  : 0
Number of Local MAC Routes    : 4
    MAC                        : 4
    MAC-IPv4                   : 0
    MAC-IPv6                   : 0
Number of Local ES:Global MAC : 1
Number of Remote MAC Routes   : 0
    MAC                        : 0
    MAC-IPv4                   : 0
    MAC-IPv6                   : 0
Number of Remote SOO MAC Routes : 0
Number of Local IMCAST Routes : 4
Number of Remote IMCAST Routes : 4
Number of Internal Labels     : 0
Number of ES Entries          : 1
Number of Neighbor Entries     : 4
EVPN Router ID                : 200.0.1.1
BGP ASN                       : 65530
PBB BSA MAC address           : 0026.982b.c1e5
Global peering timer          :      3 seconds
Global recovery timer         :     30 seconds

```

EVPN ルートターゲットを確認します。

```

Router#show bgp rt l2vpn evpn
Mon Feb 20 21:06:18.882 EST
EXTCOMM      IMP/EXP
RT:65530:1   1 / 1
RT:65530:2   1 / 1
RT:65530:3   1 / 1
RT:65530:4   1 / 1
Processed 4 entries

```

```

Locally learnt MAC routes can be viewed by forwarding table
show l2vpn forwarding bridge-domain mac-address location 0/0/cpu0
To Resynchronize MAC table from the Network Processors, use the command...
l2vpn resynchronize forwarding mac-address-table location <r/s/i>

```

Mac Address	Type	Learned from/Filtered on	LC learned	Resync Age/Last Change	Mapped to
0033.0000.0001	dynamic	Gi0/2/0/0.1	N/A	20 Feb 21:06:59	N/A

```

0033.0000.0002 dynamic Gi0/2/0/0.2          N/A      20 Feb 21:06:59    N/A
0033.0000.0003 dynamic Gi0/2/0/0.3          N/A      20 Feb 21:04:29    N/A
0033.0000.0004 dynamic Gi0/2/0/0.4          N/A      20 Feb 21:06:59    N/A

```

```

The remote routes learned via evpn enabled BD
show l2vpn forwarding bridge-domain mac-address location 0/0$
To Resynchronize MAC table from the Network Processors, use the command...
l2vpn resynchronize forwarding mac-address-table location <r/s/i>

```

Mac Address	Type	Learned from/Filtered on	LC learned	Resync Age/Last Change	Change
0033.0000.0001	EVPN	BD id: 0	N/A	N/A	N/A
0033.0000.0002	EVPN	BD id: 1	N/A	N/A	N/A
0033.0000.0003	EVPN	BD id: 2	N/A	N/A	N/A
0033.0000.0004	EVPN	BD id: 3	N/A	N/A	N/A

特定の VPN インスタンスに関係のある EVPN MAC ルートを確認します。

```

Router#show evpn evi vpn-id 1 mac
Mon Feb 20 21:36:23.574 EST

```

EVI Label	MAC address	IP address	NextHop
1 45106	0033.0000.0001	::	200.0.1.1

L2 ルーティングを確認します。

```

Router#show l2route evpn mac all
Mon Feb 20 21:39:43.953 EST

```

Topo ID	Mac Address	Prod	Next Hop(s)
0	0033.0000.0001	L2VPN	200.0.1.1/45106/ME
1	0033.0000.0002	L2VPN	200.0.1.1/45108/ME
2	0033.0000.0003	L2VPN	200.0.1.1/45110/ME
3	0033.0000.0004	L2VPN	200.0.1.1/45112/ME

EVPN ルート タイプ 2 ルートを確認します。

```

Router#show bgp l2vpn evpn route-type 2
Mon Feb 20 21:43:23.616 EST
BGP router identifier 200.0.3.1, local AS number 65530
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0 RD version: 0
BGP main routing table version 21
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

```

```

Status codes: s suppressed, d damped, h history, * valid, > best
                i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
      Network      Next Hop      Metric LocPrf Weight Path
Route Distinguisher: 200.0.1.1:1
*>i[2][0][48][0033.0000.0001][0]/104
      200.0.1.1      100      0 i
Route Distinguisher: 200.0.1.1:2
*>i[2][0][48][0033.0000.0002][0]/104
      200.0.1.1      100      0 i
Route Distinguisher: 200.0.1.1:3
*>i[2][0][48][0033.0000.0003][0]/104
      200.0.1.1      100      0 i
Route Distinguisher: 200.0.1.1:4
*>i[2][0][48][0033.0000.0004][0]/104
      200.0.1.1      100      0 i
Route Distinguisher: 200.0.3.1:1 (default for vrf bd-1-1)
*>i[2][0][48][0033.0000.0001][0]/104
      200.0.1.1      100      0 i
Route Distinguisher: 200.0.3.1:2 (default for vrf bd-1-2)
*>i[2][0][48][0033.0000.0002][0]/104
      200.0.1.1      100      0 i
Route Distinguisher: 200.0.3.1:3 (default for vrf bd-1-3)
*>i[2][0][48][0033.0000.0003][0]/104
      200.0.1.1      100      0 i
Route Distinguisher: 200.0.3.1:4 (default for vrf bd-1-4)
*>i[2][0][48][0033.0000.0004][0]/104
      200.0.1.1      100      0 i

```

Processed 8 prefixes, 8 paths

包含マルチキャストルートとルートタイプ3ルートを確認します。

```

Router#show bgp l2vpn evpn route-type 3
Mon Feb 20 21:43:33.970 EST
BGP router identifier 200.0.3.1, local AS number 65530
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 21
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

```

```

Status codes: s suppressed, d damped, h history, * valid, > best
                i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
      Network      Next Hop      Metric LocPrf Weight Path
Route Distinguisher: 200.0.1.1:1
*>i[3][0][32][200.0.1.1]/80
      200.0.1.1      100      0 i
Route Distinguisher: 200.0.1.1:2
*>i[3][0][32][200.0.1.1]/80
      200.0.1.1      100      0 i
Route Distinguisher: 200.0.1.1:3
*>i[3][0][32][200.0.1.1]/80
      200.0.1.1      100      0 i
Route Distinguisher: 200.0.1.1:4
*>i[3][0][32][200.0.1.1]/80
      200.0.1.1      100      0 i
Route Distinguisher: 200.0.3.1:1 (default for vrf bd-1-1)
*>i[3][0][32][200.0.1.1]/80
      200.0.1.1      100      0 i

```

```

*> [3] [0] [32] [200.0.3.1]/80
      0.0.0.0                                     0 i
Route Distinguisher: 200.0.3.1:2 (default for vrf bd-1-2)
*>i[3] [0] [32] [200.0.1.1]/80
      200.0.1.1                                   100 0 i
*> [3] [0] [32] [200.0.3.1]/80
      0.0.0.0                                     0 i
Route Distinguisher: 200.0.3.1:3 (default for vrf bd-1-3)
*>i[3] [0] [32] [200.0.1.1]/80
      200.0.1.1                                   100 0 i
*> [3] [0] [32] [200.0.3.1]/80
      0.0.0.0                                     0 i
Route Distinguisher: 200.0.3.1:4 (default for vrf bd-1-4)
*>i[3] [0] [32] [200.0.1.1]/80
      200.0.1.1                                   100 0 i
*> [3] [0] [32] [200.0.3.1]/80
      0.0.0.0                                     0 i

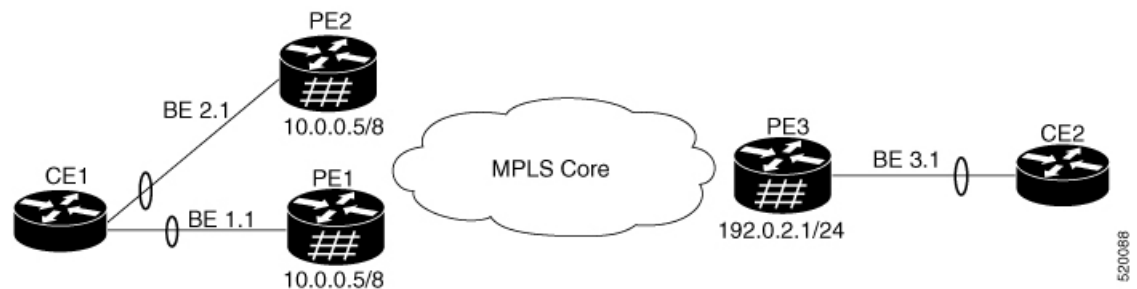
```

エニーキャストゲートウェイ IRB の EVPN シングルアクティブマルチホーミング

エニーキャストゲートウェイ IRB の EVPN シングルアクティブマルチホーミング機能は、シングルアクティブ冗長モードをサポートしています。このモードでは、プロバイダーエッジ (PE) ノードは、EVPN サービスインスタンス (EVI) に基づいて、イーサネットセグメントとの間で発着信するイーサネットセグメントロードバランストラフィックにローカルに接続されます。EVPN サービスインスタンス内では、1つの PE のみがイーサネットセグメント (ES) との間で発着信するトラフィックを転送します。この機能は、サブネット間シナリオのみをサポートします。

図 8: EVPN: エニーキャストゲートウェイ IRB の EVPN シングルアクティブマルチホーミング

Different bundles on CE1



CE1 が PE1 や PE2 にマルチホームされているトポロジについて考えてみます。バンドルイーサネットインターフェイスは BE 1.1、BE 2.1 です。入力インターフェイスは CE1 上の同じスイッチングドメインに属している必要があります。これらのピアリング PE の両方で、ホストルーティングを有効にし、エニーキャストゲートウェイ IP アドレスを設定します。PE1 と PE2 は MPLS コアを通じて PE3 に接続しています。PE3 は、サブネット 10.0.0.5/8 から両方のピアリング PE に到達可能です。ピアリング PE は、PE3 サブネット 192.0.2.1/24 に到達可能です。CE2 はイーサネットインターフェイスバンドルを通じて PE3 に接続されています。PE1 と PE2

はタイプ 4 ルートをアドバタイズしてから、指定フォワーダ（DF）の選択を実行します。非 DF はシングルアクティブ モードの両方向のトラフィックをブロックします。

CE1 から CE2 へのトラフィック フローを考えてみます。CE1 は PE1 と PE2 の両方に Address Resolution Protocol（ARP）ブロードキャスト要求を送信します。ピアリング PE は、共有 ESI に対して指定フォワーダ（DF）の選択を実行します。PE1 が EVI の指定フォワーダである場合、PE1 は CE1 からの ARP 要求に応答します。PE2 は CE1 からのトラフィックをドロップします。その後で、すべてのユニキャストトラフィックが PE1 を通じて送信されます。PE2 は、スタンバイ状態またはブロック状態に設定されており、トラフィックはこのパスを介して送信されません。PE1 は PE3 に MAC をアドバタイズします。PE3 は常に PE1 を通じてトラフィックを送受信します。PE3 はイーサネットインターフェイスバンドルを介してトラフィックを CE2 に送信します。BE1 に障害が発生した場合、PE2 は、PE2 を通過する DF およびトラフィックフローになります。

EVPN シングルアクティブ マルチホーミングの設定

EVPN シングルアクティブマルチホーミング機能を設定するには、PE1 と PE2 上で次のタスクを実行します。

- ホストルーティングを使用した EVPN IRB の設定
- EVPN イーサネットセグメントの設定
- レイヤ 2 インターフェイスの設定
- ブリッジドメインの設定
- VRF の設定

EVPN イーサネットセグメントの設定

EVPN イーサネットセグメントを設定するには、次のタスクを実行します。

```
Router# configure
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether1
Router(config-evpn-ac)# ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 40.00.00.00.00.00.00.01
Router(config-evpn-ac-es)# load-balancing-mode single-active
Router(config-evpn-ac-es)# bgp route-target 4000.0000.0001
Router(config-evpn-ac-es)# comit
```

実行コンフィギュレーション

```
configure
evpn
interface Bundle-Ether1
ethernet-segment
identifier type 0 40.00.00.00.00.00.00.01
load-balancing-mode single-active
bgp route-target 4000.0000.0001
!
```

```
!
```

EVPN サービス インスタンス (EVI) パラメータの設定

EVPN サービス インスタンス (EVI) パラメータを定義するには、このタスクを実行します。

```
Router# configure
Router (config)# evpn
Router (config-evpn)# evi 6005
Router (config-evpn-evi)# bgp
Router (config-evpn-evi-bgp)# rd 200:50
Router (config-evpn-evi-bgp)# route-target import 100:6005
Router (config-evpn-evi-bgp)# route-target export 100:6005
Router (config-evpn-evi-bgp)# commit
```

実行コンフィギュレーション

```
configure
evpn
  evi 6005
  bgp
    rd 200:50
    route-target import 100:6005
    route-target export 100:6005
!
```

レイヤ2 インターフェイスの設定

レイヤ2 インターフェイスを定義するには、次のタスクを実行します。

```
Router# configure
Router (config)# interface bundle-ether2.1 l2transport
Router (config-subif-l2)# no shutdown
Router (config-subif-l2)# encapsulation dot1q 1
Router (config-subif-l2)# rewrite ingress tag pop 1 symmetric
Router (config-subif-l2)#commit
Router (config-subif-l2)#exit
```

実行コンフィギュレーション

この項では、レイヤ2 インターフェイスの実行コンフィギュレーションを示します。

```
configure
interface bundle-ether2.1 l2transport
  no shutdown
  encapsulation dot1q 1
  rewrite ingress tag pop 1 symmetric
!
```

ブリッジ ドメインの設定

次のステップを実行して PE1 と PE2 上にブリッジ ドメインを設定します。

```
Router# configure
```

```

Router(config)# l2vpn
Router(config-l2vpn)# bridge group 6005
Router(config-l2vpn-bg)# bridge-domain 6005
Router(config-l2vpn-bg-bd)# interface Bundle-Ether2.1
Router(config-l2vpn-bg-bd-ac)# evi 6005
Router(config-l2vpnbg-bd-evi)# commit
Router(config-l2vpnbg-bd-evi)# exit

```

実行コンフィギュレーション

この項では、ブリッジドレインの実行コンフィギュレーションを示します。

```

configure
l2vpn
bridge group 6005
  bridge-domain 6005
    interface Bundle-Ether2.1
      evi 6005
!

```

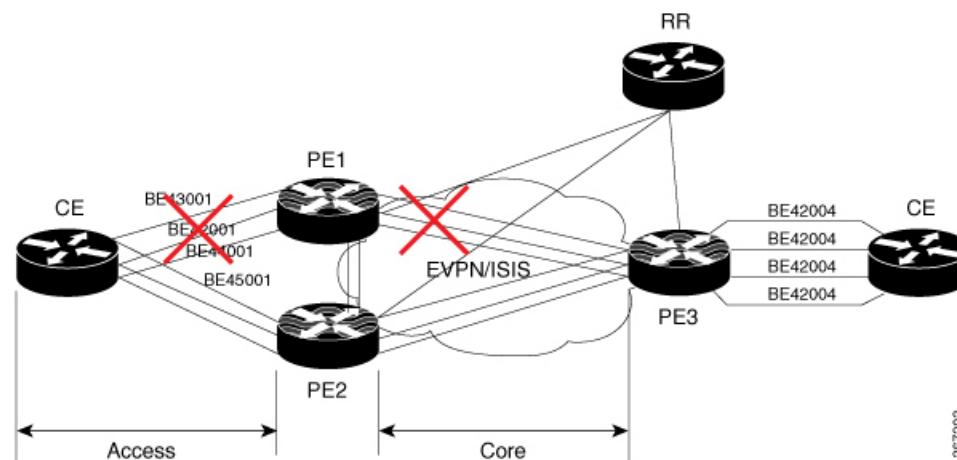
EVPN コア分離保護

EVPN コア分離保護機能を使用すると、コア内のリンク障害をモニタして検出することができます。プロバイダーエッジ (PE) デバイスでコアリンク障害が検出されると、EVPN は、PE のイーサネットセグメント (ES) を停止します。ES は、カスタマーエッジ (CE) デバイスに接続しているアクセスインターフェイスに関連付けられています。

EVPN は、ICCP のコア分離の検出を置き換えるものです。この新機能により、EVPN 環境で ICCP を使用する必要がなくなります。

CE が PE1 および PE2 に接続されているトポロジを考えてみます。PE1、PE2、および PE3 では、MPLS コアネットワーク上で EVPN が実行されています。コアインターフェイスにはギガビットイーサネットまたはバンドルインターフェイスを使用できます。

図 9: EVPN コア分離保護



PE1 のコアリンクがダウンすると、EVPN はリンク障害を検出し、アクセスネットワークをダウンさせてコアネットワークから PE1 ノードを分離します。これにより、CE は PE1 にトラ

フィックを送信できなくなります。BGP セッションもダウンしているため、BGP は、障害が発生した PE によってアドバタイズされたすべてのルートを無効にします。これにより、リモート PE2 および PE3 は、L2FIB 内のネクストホップ パスリストと MAC ルートを更新します。PE2 はすべてのトラフィックの転送者になるため、コアネットワークから PE1 を分離します。

すべてのコア インターフェイスと BGP セッションがアップすると、PE1 はイーサネット A-D イーサネットセグメント (ES-EAD) ルートを再度アドバタイズし、サービスカービングをトリガーして、コア ネットワークの一部になります。

EVPN コア分離保護の設定

EVPN グループの配下にコア インターフェイスを設定し、そのグループを、CE に接続された接続回線 (AC) であるイーサネット セグメントに関連付けます。すべてのコア インターフェイスがダウンすると、EVPN は、関連付けられているアクセス インターフェイスをダウンさせます。これにより、CE デバイスは自身のバンドル内でこれらのリンクを使用できなくなります。グループの一部であるすべてのインターフェイスがダウンすると、EVPN はバンドルをダウンさせ、ES-EAD ルートを取り消します。

制約事項

- EVPN の配下には最大 24 のグループを作成できます。
- グループの下には最大 12 のコア インターフェイスを追加できます。
- コア インターフェイスはグループ間で再利用できます。コア インターフェイスは、バンドル インターフェイスにすることができます。
- EVPN グループにはコア インターフェイスのみを含める必要があります。EVPN グループの配下にアクセス インターフェイスを追加しないでください。
- アクセス インターフェイスは、バンドル インターフェイスにしかできません。
- EVPN コアに面するインターフェイスは、物理インターフェイスまたはバンドル メイン インターフェイスのみにする必要があります。サブインターフェイスはサポートされていません。

```
Router# configure
Router(config)# evpn
Router(config-evpn)# group 42001
Router(config-evpn-group)# core interface GigabitEthernet0/2/0/1
Router(config-evpn-group)# core interface GigabitEthernet0/2/0/3
Router(config-evpn-group)#exit
!
Router(config-evpn)# group 43001
Router(config-evpn-group)# core interface GigabitEthernet0/2/0/2
Router(config-evpn-group)# core interface GigabitEthernet0/2/0/4
Router(config-evpn-group)#exit
!
Router# configure
Router(config)# evpn
Router(config-evpn)# interface bundle-Ether 42001
Router(config-evpn-ac)# core-isolation-group 42001
```

```
Router(config-evpn-ac)# exit
!
Router(config-evpn)# interface bundle-Ether 43001
Router(config-evpn-ac)# core-isolation-group 43001
Router(config-evpn-ac)# commit
```

実行コンフィギュレーション

```
configure
evpn
group 42001
  core interface GigabitEthernet0/2/0/1
  core interface GigabitEthernet0/2/0/3
  !
group 43001
  core interface GigabitEthernet0/2/0/2
  core interface GigabitEthernet0/2/0/4
  !
!
configure
evpn
interface bundle-Ether 42001
  core-isolation-group 42001
  !
interface bundle-Ether 43001
  core-isolation-group 43001
  !
!
```

確認

show evpn group コマンドは、evpn グループの完全なリストと、それらに関連付けられているコア インターフェイスおよびアクセス インターフェイスを表示します。各インターフェイスのステータス（アップまたはダウン）も表示されます。アクセスインターフェイスがアップ状態になるには、コア インターフェイスが少なくとも 1 つアップ状態である必要があります。

```
Router# show evpn group /* Lists specific group with core-interfaces and access interface
status */
EVPN Group: 42001
State: Ready
Core Interfaces:
  Bundle-Ethernet110: down
  Bundle-Ethernet111: down
  GigabethEthernet0/2/0/1: up
  GigabethEthernet0/2/0/3: up
  GigabethEthernet0/4/0/8: up
  GigabethEthernet0/4/0/9: up
  GigabethEthernet0/4/0/10: up
Access Interfaces:
  Bundle-Ether42001: up

EVPN Group: 43001
State: Ready
Core Interfaces:
  Bundle-Ethernet110: down
  GigabethEthernet0/2/0/2: up
  GigabethEthernet0/2/0/4: up
```

```
GigabethEthernet0/4/0/9: up
```

```
Access Interfaces:  
Bundle-Ether43001: up
```

EVPN ルーティング ポリシー

EVPN ルーティング ポリシー機能では、アドレスファミリ L2VPN EVPN のルートポリシー サポートを提供します。この機能は、EVPN ルートフィルタリング機能をルーティングポリシー 言語 (RPL) に追加します。フィルタリングはさまざまな EVPN 属性に基づきます。

ピアから受け入れるか、ピアにアドバタイズされる、または1個のルーティングプロトコルから別のプロトコルへ再配布されるときに、ルートを検査し、フィルタリングして、属性を変更するように、ルーティング ポリシーがルータに指示します。

この機能により、より粒度が高いルートポリシーの定義を提供するルートポリシー一致基準の EVPN ルートタイプ 1 ~ 5 の EVPN ネットワーク層到達可能性情報 (NLRI) 属性を使用して ルートポリシーを設定できます。たとえば、ルートポリシーを特定の EVPN ルートタイプ のみに適用したり、任意の組み合わせの EVPN NLRI 属性に適用できます。この機能は、ルートポリシーを有効にして EVPN NLRI 属性でフィルタリングすることで、ソリューションの設定および展開に柔軟性をもたらします。

この機能を実装するには、次の概念を理解する必要があります。

- ルーティング ポリシー言語
- ルーティング ポリシー言語の構造
- ルーティング ポリシー言語コンポーネント
- ルーティング ポリシー言語使用方法
- ポリシー定義
- パラメータ化
- ポリシー適用のセマンティック
- ポリシー ステートメント
- 接続点

これらの概念については「[ルーティング ポリシーの実装](#)」を参照してください。

現在、この機能は接続ポイント「イン」または「アウト」の BGP ネイバーでのみサポートされています。ルート ポリシーは BGP ネイバーのインバウンドまたはアウトバウンドのみに適用できます。

EVPN ルート タイプ

EVPN NLRI には次のさまざまなルート タイプがあります。

ルートタイプ1：イーサネット自動検出（AD）ルート

イーサネット（AD）ルートは、EVIごととイーサネットセグメント識別子（ESI）ごとにアドバタイズされます。これらのルートは、イーサネットセグメント（ES）ごとに送信されます。これらはESに属しているEVIのリストを伝送します。ESIフィールドは、CEがシングルホームの場合はゼロに設定されます。

イーサネット A-D ルートタイプ固有の EVPN NLRI は次のフィールドで構成されます。

Route Type (1 octet)	*
Length (1 octet)	
Route Distinguisher (RD) (8 octets)	*
Ethernet Segment Identifier (10 octets)	*
Ethernet Tag ID (4 octets)	*
MPLS Label (3 octets)	

NLRI の形式：ルートタイプ1：

[Type] [Len] [RD] [ESI] [ETag] [MPLS Label]

ネット属性：[Type] [RD] [ESI] [ETag]

パス属性：[MPLS Label]

例

```
route-policy evpn-policy
  if rd in (1.1.1.1:0) [and/or evpn-route-type is 1] [and/or esi in
(0a1.a2a3.a4a5.a6a7.a8a9)] [and/or etag is 4294967295] then
    set ..
  endif
end-policy
!
route-policy evpn-policy
  if rd in (1.1.1.2:0) [and/or evpn-route-type is 1] [and/or esi in
(00a1.a2a3.a4a5.a6a7.a8a9)] [and/or etag is 4294967295] then
    set ..
  endif
end-policy
```

ルートタイプ2：MAC/IP アドバタイズメントルート

ホストの IP アドレスと MAC アドレスが NLRI 内のピアにアドバタイズされます。MAC アドレスのコントロールプレーン学習は不明ユニキャストのフラッドを削減します。

MAC/IP アドバタイズメントルートタイプ固有の EVPN NLRI は次のフィールドで構成されま

Route Type (1 octet)	*
Length (1 octet)	
RD (8 octets)	*
Ethernet Segment Identifier (10 octets)	
Ethernet Tag ID (4 octets)	*
MAC Address Length (1 octet)	*
MAC Address (6 octets)	*
IP Address Length (1 octet)	*
IP Address (0, 4, or 16 octets)	*
MPLS Label1 (3 octets)	
MPLS Label2 (0 or 3 octets)	

000306

NLRI の形式 : ルートタイプ 2 :

[Type][Len][RD][ESI][ETag][MAC Addr Len][MAC Addr][IP Addr Len][IP Addr][MPLS Label1][MPLS Label2]

ネット属性 : [Type][RD][ETag][MAC Addr Len][MAC Addr][IP Addr Len][IP Addr]

パス属性 : [ESI], [MPLS Label1], [MPLS Label2]

例

```
route-policy evpn-policy
  if rd in (1.1.1.2:0) [and/or evpn-route-type is 2] [and/or esi in
(0000.0000.0000.0000.0000)] [and/or etag is 0] [and/or macaddress in (0013.aabb.ccdd)]
[and/or destination in (1.2.3.4/32)] then
    set ..
  endif
end-policy
```

ルートタイプ 3 : 包括的なマルチキャストイーサネットタグルート

このルートは、送信元 PE からリモート PE へのブロードキャスト、不明ユニキャスト、およびマルチキャスト (BUM) トラフィック用の接続を確立します。このルートは、VLAN ごとと ESI ごとにアドバタイズされます。

包括的マルチキャストイーサネットタグルートタイプ固有のEVPN NLRIは次のフィールドで構成されます。

Route Type (1 octet)	*
Length (1 octet)	
RD (8 octets)	*
Ethernet Tag ID (4 octets)	*
IP Address Length (1 octet)	*
Originating Router's IP Address (4 or 16 octets)	*

NLRI の形式 : ルートタイプ 3 :

[Type][Len][RD][ETag][IP Addr Len][Originating Router's IP Addr]

ネット属性 : [Type][RD][ETag][IP Addr Len][Originating Router's IP Addr]

例

```
route-policy evpn-policy
  if rd in (1.1.1.1:300) [and/or evpn-route-type is 3] [and/or etag is 0] [and/or
evpn-originator in (1.1.1.1)] then
    set ..
  endif
end-policy
```

ルートタイプ 4 : イーサネットセグメントルート

イーサネットセグメントルートではCEデバイスを2台のデバイスまたはPEデバイスを接続できます。ESルートでは同じイーサネットセグメントに接続されているPEデバイスを検出できます。

イーサネットセグメントルートタイプ固有のEVPN NLRIは次のフィールドで構成されます。

```

+-----+
|Route Type (1 octet)          |*
+-----+
|Length (1 octet)             |
+-----+
|RD (8 octets)                |*
+-----+
|Ethernet Segment Identifier (10 octets)|*
+-----+
|IP Address Length (1 octet)  |*
+-----+
|Originating Router's IP Address |*
|(4 or 16 octets)            |
+-----+

```

3-603-08

NLRI の形式 : ルートタイプ 4 :

[Type][Len][RD][ESI][IP Addr Len][Originating Router's IP Addr]

ネット属性 : [Type][RD][ESI][IP Addr Len][Originating Router's IP Addr]

例

```

route-policy evpn-policy
  if rd in (1.1.1.1:0) [and/or evpn-route-type is 4] [and/or esi in
(00a1.a2a3.a4a5.a6a7.a8a9)] [and/or evpn-originator in (1.1.1.1)] then
    set ..
  endif
end-policy

```

ルートタイプ 5 : IP プレフィックスルート

IP プレフィックスルート タイプ固有の EVPN NLRI は次のフィールドで構成されます。

Route Type (1 octet)	*
Length (1 octet)	
RD (8 octets)	*
Ethernet Segment Identifier (10 octets)	
Ethernet Tag ID (4 octets)	*
IP Address Length (1 octet)	*
IP Address (4 or 16 octets)	*
GW IP Address (4 or 16 octets)	
MPLS Label (3 octets)	

NLRI の形式 : ルートタイプ 5 :

[Type][Len][RD][ESI][ETag][IP Addr Len][IP Addr][GW IP Addr][Label]

ネット属性 : [Type][RD][ETag][IP Addr Len][IP Addr]

パス属性 : [ESI], [GW IP Addr], [Label]

例

```
route-policy evpn-policy
  if rd in (30.30.30.30:1) [and/or evpn-route-type is 5] [and/or esi in
(0000.0000.0000.0000.0000)] [and/or etag is 0] [and/or destination in (12.2.0.0/16)]
[and/or evpn-gateway in (0.0.0.0)] then
    set ..
  endif
end-policy
```

EVPN RPL 属性

ルート識別子

ルート識別子 (rd) 属性は、8 オクテットで構成されます。rd は EVPN ルートのタイプそれぞれに指定できます。この属性は、ルートポリシーでは必須ではありません。

例

```
rd in (1.2.3.4:0)
```


EVPN ルート タイプ

EVPN ルートタイプ属性は、1 オクテットで構成されます。これによって EVPN ルートタイプが指定されます。EVPN ルートタイプ属性は、特定の EVPN NLRI プレフィックス形式を識別するために使用されます。これは、すべての EVPN ルートタイプのネット属性の 1 つです。

例

```
evpn-route-type is 3
```

The following are the various EVPN route types that can be used:

- 1 - ethernet-ad
- 2 - mac-advertisement
- 3 - inclusive-multicast
- 4 - ethernet-segment
- 5 - ip-advertisement

IP プレフィックス

IP プレフィックス属性は、それぞれ 4 つの部分（アドレス、マスク長、最小一致長、最大一致長）がある IPv4 または IPv6 プレフィックス一致指定を保持しています。アドレスは必須ですが、他の 3 つの部分は任意です。EVPN ルートタイプ 2 での IP プレフィックスの指定により、IPv4 または IPv6 のいずれかのホスト IP アドレスを表します（/32 または /128）。EVPN ルートタイプ 5 の IP プレフィックスでの指定により、IPv4 または IPv6 のサブネットを表します。これは、EVPN ルート 2 と 5 のネット属性の 1 つです。

例

```
destination in (128.47.10.2/32)
destination in (128.47.0.0/16)
destination in (128:47::1/128)
destination in (128:47::0/112)
```

esi

イーサネットセグメント識別子（ESI）属性は、10 オクテットで構成されます。これは EVPN ルートタイプ 1 と 4 のネット属性であり、EVPN ルートタイプ 2 と 5 のパス属性です。

例

```
esi in (ffff.ffff.ffff.ffff.fff0)
```

etag

イーサネットタグ属性は 4 オクテットで構成されます。イーサネットタグは、特定のブロードキャストドメイン（VLAN など）を識別します。EVPN インスタンスは 1 つまたは複数のブ

ロードキャスト ドメインで構成されます。これは EVPN ルート タイプ 1、2、3、および 5 の ネット属性です。

例

```
etag in (10000)
```

mac

MAC 属性は 6 オクテットで構成されます。これは、EVPN ルート 2 の ネット属性です。

例

```
mac in (0206.acb1.e806)
```

evpn-originator

evpn-originator 属性は、発信元ルータの IP アドレス（4 または 16 オクテット）を指定します。これは、EVPN ルート 3 と 4 の ネット属性です。

例

```
evpn-originator in (1.2.3.4)
```

evpn-gateway

evpn-gateway 属性は、ゲートウェイの IP アドレスを指定します。ゲートウェイ IP アドレスは 32 ビットまたは 128 ビットのフィールド（IPv4 または IPv6）であり、IP プレフィックスに応じてオーバーレイ ネクストホップをエンコードします。ゲートウェイ IP アドレス フィールドは、オーバーレイ ネクストホップとして使用しない場合はゼロに設定できます。これは、EVPN ルート 5 の パス属性です。

例

```
evpn-gateway in (1.2.3.4)
```

EVPN RPL 属性セット

このコンテキストでは、セットという用語を、順序付けのない固有のエレメントの集合を意味する数学的な概念で使用されます。ポリシー言語は、セットをマッチング用の値のグループに対するコンテナとして提供します。セットは、条件式で使用されます。セットの要素はカンマで区切ります。ヌル（空）のセットは許可されます。

prefix-set

prefix-set は、それぞれ 4 つの部分（アドレス、マスク長、最小一致長、最大一致長）がある IPv4 または IPv6 プレフィックス一致指定を保持しています。アドレスは必須ですが、他の 3 つの部分は任意です。prefix-set は 1 つまたは複数の IP プレフィックスを指定します。

例

```
prefix-set ip_prefix_set
14.2.0.0/16,
54.0.0.0/16,
12.12.12.0/24,
50:50::1:0/112
end-set
```

mac-set

mac-set は 1 つまたは複数の MAC プレフィックスを指定します。

例

```
mac-set mac_address_set
1234.2345.6789,
2345.3456.7890
end-set
```

esi-set

esi-set は、1 つまたは複数の ESI を指定します。

例

```
esi-set evpn_esi_set
1234.2345.3456.4567.5678,
1234.2345.3456.4567.5670
end-set
```

etag-set

etag-set は、1 つまたは複数のイーサネット タグを指定します。

例

```
etag-set evpn_etag_set
10000,
20000
end-set
```

EVPN RPL 機能の設定

次の項では、mac-set、esi-set、evpn-gateway、および evpn-originator を設定する方法について説明します。

```

/* Configuring a mac-set and referring it in a route-policy (Attach point - neighbor-in)
*/
Router# configure
Router(config)# mac-set demo_mac_set
Router(config-mac)# 1234.ffff.aaa3,
Router(config-mac)# 2323.4444.ffff
Router(config-mac)# end-set
Router(config)# !
Router(config)# route-policy policy_use_pass_mac_set
Router(config-rpl)# if mac in demo_mac_set then
Router(config-rpl-if)# set med 200
Router(config-rpl-if)# else
Router(config-rpl-else)# set med 1000
Router(config-rpl-else)# endif
Router(config-rpl)# end-policy
Router(config)# commit
Router(config)# router bgp 100
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# !
Router(config-bgp-af)# neighbor 10.0.0.10
Router(config-bgp-nbr)# remote-as 8
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy policy_use_pass_mac_set in
Router(config-bgp-nbr-af)# commit

/* Configuring a esi-set and referring it in a route-policy (Attach point - neighbor-in)
*/
Router# configure
Router(config)# esi-set demo_esi
Router(config-esi)# ad34.1233.1222.ffff.44ff,
Router(config-esi)# ad34.1233.1222.ffff.6666
Router(config-esi)# end-set
Router(config)# !
Router(config)# route-policy use_esi
Router(config-rpl)# if esi in demo_esi then
Router(config-rpl-if)# set local-preference 100
Router(config-rpl-if)# else
Router(config-rpl-else)# set local-preference 300
Router(config-rpl-else)# endif
Router(config-rpl)# end-policy
Router(config)# commit

/* Configuring evpn-gateway/evpn-originator in a route-policy (Attach point - neighbor-in
and out) */
Router# configure
Router(config)# route-policy gateway_demo
Router(config-rpl)# if evpn-gateway in (10.0.0.0/32) then
Router(config-rpl-if)# pass
Router(config-rpl-if)# endif
Router(config-rpl)# end-policy
Router(config)# commit
Router(config)# route-policy originator_demo
Router(config-rpl)# if evpn-originator in (10.0.0.1/32) then
Router(config-rpl-if)# set local-preference 100
Router(config-rpl-if)# else

```

```
Router(config-rpl-else)# set med 200
Router(config-rpl-else)# endif
Router(config-rpl)# end-policy
Router(config)# commit
Router(config)# router bgp 100
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# !
Router(config-bgp-af)# neighbor 10.0.0.10
Router(config-bgp-nbr)# remote-as 8
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy gateway_demo in
Router(config-bgp-nbr-af)# route-policy originator_demo out
Router(config-bgp-nbr-af)# commit
```

実行コンフィギュレーション

```
/* Configuring a mac-set and refering it in a route-policy (Attach point - neighbor-in)
*/
mac-set demo_mac_set
  1234.ffff.aaa3,
  2323.4444.ffff
end-set
!
route-policy policy_use_pass_mac_set
  if mac in demo_mac_set then
    set med 200
  else
    set med 1000
  endif
end-policy
!
router bgp 100
  address-family ipv4 unicast
  !
  neighbor 10.0.0.10
    remote-as 8
    address-family ipv4 unicast
    route-policy policy_use_pass_mac_set in
  !
  !
end

/* Configuring a esi-set and refering it in a route-policy (Attach point - neighbor-in)
*/
Wed Oct 26 11:52:23.720 IST
esi-set demo_esi
  ad34.1233.1222.ffff.44ff,
  ad34.1233.1222.ffff.6666
end-set
!
route-policy use_esi
  if esi in demo_esi then
    set local-preference 100
  else
    set local-preference 300
  endif
end-policy
```

EVPN ルート ポリシーの例

```
route-policy ex_2
  if rd in (2.2.18.2:1004) and evpn-route-type is 1 then
    drop
  elseif rd in (2.2.18.2:1009) and evpn-route-type is 1 then
    drop
  else
    pass
  endif
end-policy
!
route-policy ex_3
  if evpn-route-type is 5 then
    set extcommunity bandwidth (100:9999)
  else
    pass
  endif
end-policy
!
route-policy samp
end-policy
!
route-policy samp1
  if rd in (30.0.101.2:0) then
    pass
  endif
end-policy
!
route-policy samp2
  if rd in (30.0.101.2:0, 1:1) then
    pass
  endif
end-policy
!
route-policy samp3
  if rd in (*:*) then
    pass
  endif
end-policy
!
route-policy samp4
  if rd in (30.0.101.2:*) then
    pass
  endif
end-policy
!
route-policy samp5
  if evpn-route-type is 1 then
    pass
  endif
end-policy
!
route-policy samp6
  if evpn-route-type is 2 or evpn-route-type is 5 then
    pass
  endif
end-policy
!
route-policy samp7
  if evpn-route-type is 4 or evpn-route-type is 3 then
    pass
```

```
endif
end-policy
!
route-policy samp8
  if evpn-route-type is 1 or evpn-route-type is 2 or evpn-route-type is 3 then
    pass
  endif
end-policy
!
route-policy samp9
  if evpn-route-type is 1 or evpn-route-type is 2 or evpn-route-type is 3 or
  evpn-route-type is 4 then
    pass
  endif
end-policy
!
route-policy test1
  if evpn-route-type is 2 then
    set next-hop 10.2.3.4
  else
    pass
  endif
end-policy
!
route-policy test2
  if evpn-route-type is 2 then
    set next-hop 10.10.10.10
  else
    drop
  endif
end-policy
!
route-policy test3
  if evpn-route-type is 1 then
    set tag 9988
  else
    pass
  endif
end-policy
!
route-policy samp21
  if mac in (6000.6000.6000) then
    pass
  endif
end-policy
!
route-policy samp22
  if extcommunity rt matches-any (100:1001) then
    pass
  else
    drop
  endif
end-policy
!
route-policy samp23
  if evpn-route-type is 1 and esi in (aaaa.bbbb.cccc.dddd.eeee) then
    pass
  else
    drop
  endif
end-policy
!
route-policy samp24
  if evpn-route-type is 5 and extcommunity rt matches-any (100:1001) then
```

```
        pass
      else
        drop
      endif
    end-policy
  !
  route-policy samp25
    if evpn-route-type is 2 and esi in (1234.1234.1234.1234.1236) then
      pass
    else
      drop
    endif
  end-policy
  !
  route-policy samp26
    if etag in (20000) then
      pass
    else
      drop
    endif
  end-policy
  !
  route-policy samp27
    if destination in (99.99.99.1) and etag in (20000) then
      pass
    else
      drop
    endif
  end-policy
  !
  route-policy samp31
    if evpn-route-type is 1 or evpn-route-type is 2 or evpn-route-type is 3 or
    evpn-route-type is 4 or evpn-route-type is 5 then
      pass
    else
      drop
    endif
  end-policy
  !
  route-policy samp33
    if esi in evpn_esi_set1 then
      pass
    else
      drop
    endif
  end-policy
  !
  route-policy samp34
    if destination in (90:1:1::9/128) then
      pass
    else
      drop
    endif
  end-policy
  !
  route-policy samp35
    if destination in evpn_prefix_set1 then
      pass
    else
      drop
    endif
  end-policy
  !
  route-policy samp36
```



```
    if evpn-route-type is 3 and evpn-originator in (80:1:1::3) then
        pass
    else
        drop
    endif
end-policy
!
route-policy samp37
    if evpn-gateway in (10:10::10) then
        pass
    else
        drop
    endif
end-policy
!
route-policy samp38
    if mac in evpn_mac_set1 then
        pass
    else
        drop
    endif
end-policy
!
route-policy samp39
    if mac in (6000.6000.6002) then
        pass
    else
        drop
    endif
end-policy
!
route-policy samp41
    if evpn-gateway in (10.10.10.10, 10:10::10) then
        pass
    else
        drop
    endif
end-policy
!
route-policy samp42
    if evpn-originator in (24.162.160.1/32, 70:1:1::1/128) then
        pass
    else
        drop
    endif
end-policy
!
route-policy example
    if rd in (62300:1903) and evpn-route-type is 1 then
        drop
    elseif rd in (62300:19032) and evpn-route-type is 1 then
        drop
    else
        pass
    endif
end-policy
!
route-policy samp100
    if evpn-route-type is 4 or evpn-route-type is 5 then
        drop
    else
        pass
    endif
end-policy
```

```
!  
route-policy samp101  
  if evpn-route-type is 4 then  
    drop  
  else  
    pass  
  endif  
end-policy  
!  
route-policy samp102  
  if evpn-route-type is 4 then  
    drop  
  elseif evpn-route-type is 5 then  
    drop  
  else  
    pass  
  endif  
end-policy  
!  
route-policy samp103  
  if evpn-route-type is 2 and destination in evpn_prefix_set1 then  
    drop  
  else  
    pass  
  endif  
end-policy  
!  
route-policy samp104  
  if evpn-route-type is 1 and etag in evpn_etag_set1 then  
    drop  
  elseif evpn-route-type is 2 and mac in evpn_mac_set1 then  
    drop  
  elseif evpn-route-type is 5 and esi in evpn_esi_set1 then  
    drop  
  else  
    pass  
  endif  
end-policy  
!
```

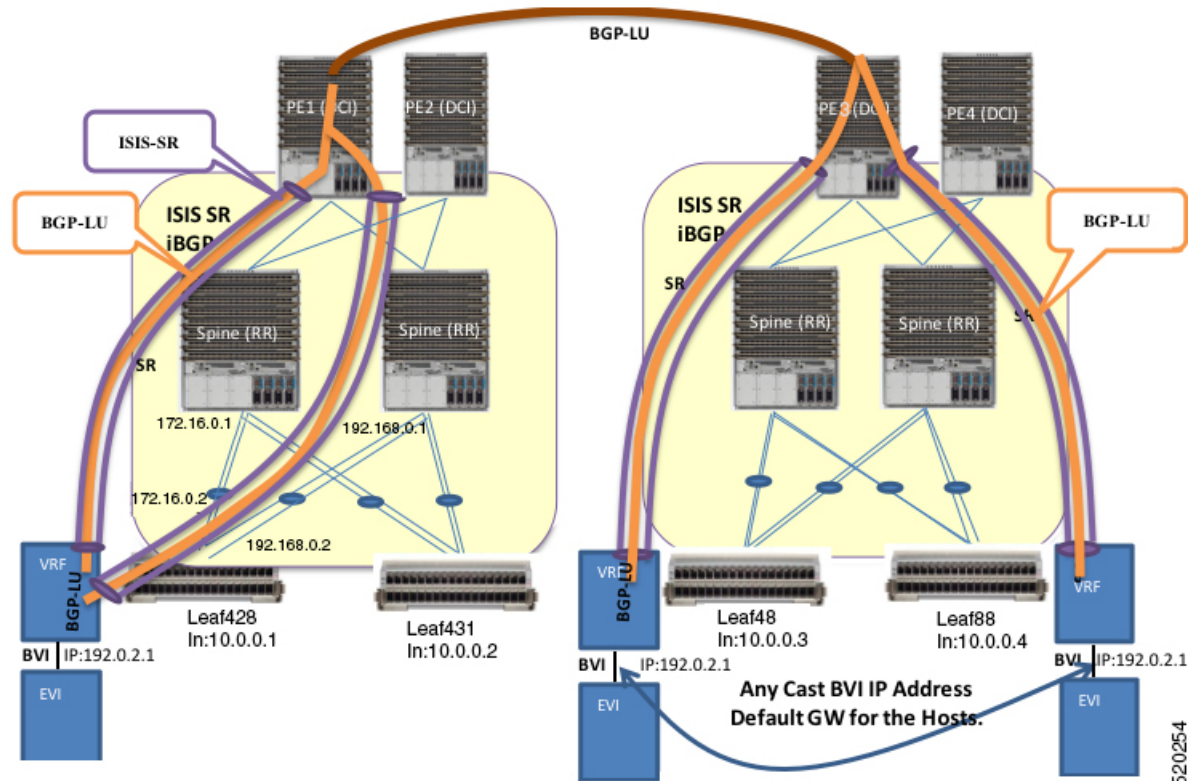
BGP-LU アンダーレイを介した EVPN ブリッジングおよび VPWS サービス

BGP-LU アンダーレイを介した EVPN ブリッジングおよび VPWS サービスでは、データセンター（DC）間のエンドツーエンド EVPN サービスを設定できます。この機能を使用すると、トランスポート、BGP-LU、サービスレベルの 3 レベルで ECMP を実行できます。

この機能は次のサービスをサポートしています。

- IGP を使用して BGP-LU を介した IRB VRF（SR または非 SR : LDP、IGP）
- IGP を使用して BGP-LU を介した EVPN のエイリアシング（SR または非 SR : LDP、IGP）
- IGP を使用して BGP-LU を介した VPWS

図 10: BGP-LU アンダーレイを介した EVPN ブリッジングおよび VPWS サービス



この項では、BGP-LUアンダーレイ機能を使用したEVPNブリッジングおよびVPWSサービスのトポロジについて説明します。

- DCIを介して接続されている2つのデータセンターについて考えてみます。リーフノードでのブリッジングおよびサブネット間ルーティングにより、EVPNを設定します。
- BVI接続回線を備えたEVPNインスタンスをVRFを実装したインターフェイスに設定します。
- 同じMACアドレスを持つエニーキャストIPアドレスを使用して、BVIインターフェイスを設定します。これは、同じEVPNブリッジドメイン全体で、すべてのホストのデフォルトゲートウェイになります。
- リーフは、ローカルホストのデフォルトゲートウェイとして機能します。
- リーフノードにホストを接続します。リーフノードはスパインを介してルーティングされます。DC相互接続の場合、スパインはプロバイダーエッジ (PE) デバイスとデータセンター相互接続 (DCI) を介して接続されます。
- IGPやI-BGPというラベルが付けられたIS-ISは、リーフノード、スパイン、DCIの内部で有効になります。スパインは、ルートリフレクタ (RR) として機能します。
- リーフノード、スパイン、DCIの間で、IS-IS SRポリシーを設定します。
- DC間でBGP-LUを設定します。

- ラベル付けされたユニキャスト BGP ルータは、リーフノードとトンネリング全体で IGP ラベル付きパス (IS-IS SR) を介して学習されます。
たとえば、Leaf428 では、BGP-LU ルートがリモートループバック 10.0.0.3 と 10.0.0.4 用に学習されます。
- IRB (BVI) インターフェイスルートは EVPN インスタンス全体で学習され、トンネリングされたラベル付きルートとして BGP-LU を介してプログラミングされます。
たとえば Leaf428 では、192.0.2.1 は 10.0.0.3 と 10.0.0.4 の 2 つの BGP-LU パスを使用して到達できます。

BGP-LU アンダーレイを介した EVPN ブリッジングおよび VPWS サービスの設定

BGP-LU アンダーレイ機能を介して EVPN ブリッジングおよび VPWS サービスを設定するには、次のタスクを実行します。

- IGP の設定
- BGP の設定
- EVPN インスタンスと ESI の設定
- BVI (IRB) インターフェイスの設定
- VRF の設定
- VRF を使用した BVI の設定
- BGP での VRF の設定
- ブリッジドメインの設定と接続回線および EVPN インスタンスとの関連付け
- ブリッジドメインの設定と接続回線、EVPN インスタンス、および BVI との関連付け
- EVPN VPWS の設定

設定例

```
/* Configure IGP */
IGP configuration is a pre-requisite to configure EVPN. IGP can be OSPF or ISIS.
Router# configure
Router(config)#router ospf 1
Router(config-ospf)#router-id 209.165.201.1
Router(config-ospf)#area 10
Router(config-ospf-ar)#interface loopback0\
Router(config-ospf-ar-if)#exit
Router(config-ospf-ar)#interface TenGigE0/0/0/1\
Router(config-ospf-ar-if)#exit
Router(config-ospf-ar)#interface TenGigE0/0/0/17\
Router(config-ospf-ar-if)#commit
```

```
/* Configure BGP */
Router# configure
Router(config)#router bgp 100
Router(config-bgp)#router-id 209.165.201.1
Router(config-bgp)#bgp graceful-restart
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#redistribute connected
Router(config-bgp-af)#network 209.165.200.225/27
Router(config-bgp-af)#allocate-label all
Router(config-bgp-af)#exit
Router(config-bgp)#address-family ipv6 unicast
Router(config-bgp-af)#allocate-label all
Router(config-bgp-af)#exit
Router(config-bgp)#neighbor-group spines
Router(config-bgp-nbrgrp)#remote-as 100
Router(config-bgp-nbrgrp)#update-source loopback0
Router(config-bgp-nbrgrp)#address-family ipv4 labeled-unicast multipath
Router(config-bgp-nbrgrp-af)#exit
Router(config-bgp-nbrgrp)#address-family ipv6 labeled-unicast multipath
Router(config-bgp-nbrgrp-af)#exit
Router(config-bgp-nbrgrp)#address-family l2vpn evpn
Router(config-bgp-nbrgrp-af)#advertise vpv4 unicast re-originated
Router(config-bgp-nbrgrp-af)#advertise vpv6 unicast re-originated
Router(config-bgp-nbrgrp-af)#exit
Router(config-bgp-nbrgrp)#exit
Router(config-bgp)#neighbor 209.165.200.225
Router(config-bgp-nbr)#use neighbor-group spines
Router(config-bgp-nbr)#commit

/* Configure VPN4 address-family */
Router(config)#router bgp 100
Router(config-bgp)#router-id 209.165.201.1
Router(config-bgp)#ibgp policy out enforce-modifications
Router(config-bgp)#address-family vpv4 unicast
Router(config-bgp-af)#commit

/* Configure EVPN instance and ESI */
Router#configure
Router(config)#evpn
Router(config-evpn)#evi 100
Router(config-evpn-instance)#advertise-mac
Router(config-evpn-instance-mac)#exit
Router(config-evpn-instance)#exit
Router(config-evpn)#interface Bundle-Ether1
Router(config-evpn-ac)#ethernet-segment identifier type 0 aa.aa.aa.aa.aa.aa.aa.aa.ac
Router(config-evpn-ac-es)#bgp route-target 0011.0011.0012
Router(config-evpn-ac)#commit

/* Configure BVI (IRB) Interface */
Router#configure
Router(config)#interface BVI200
Router(config-if)#ipv4 address 192.0.2.1 255.255.255.0
Router(config-if)#commit

/* Configure VRF */
Router# configure
Router(config)# vrf vpn2
Router(config-vrf)# address-family ipv4 unicast
Router(config-vrf-af)# import route-target 81:2
Router(config-vrf-af)# exit
Router(config-vrf)# address-family ipv6 unicast
Router(config-vrf-af)# import route-target 81:2
```

```

Router(config-vrf-af)# commit

/* Configure BVI with VRF */
Router(config)# interface BVI200
Router(config-if)# host-routing
Router(config-if)# vrf vpn72
Router(config-if-vrf)# ipv4 address ipv4 address 192.0.2.1 255.255.255.0
Router(config-if-vrf)# mac-address 10.1111.1
Router(config-if)# commit

/* Configure VRF under BGP */
Router(config)# router bgp 100
Router(config-bgp)# vrf vpn2
Router(config-bgp-vrf)# rd 102:2
Router(config-bgp-vrf)# address-family ipv4 unicast
Router(config-bgp-vrf-af)# label mode per-vrf
Router(config-bgp-vrf-af)# maximum-paths ibgp 8
Router(config-bgp-vrf-af)# redistribute connected
Router(config-bgp-vrf-af)# exit
Router(config-bgp-vrf)# address-family ipv6 unicast
Router(config-bgp-vrf-af)# label mode per-vrf
Router(config-bgp-vrf-af)# maximum-paths ibgp 8
Router(config-bgp-vrf-af)# redistribute connected
Router(config-bgp-vrf-af)# commit

/* Configure bridge domain and associate with attachment circuits and EVPN instance */
Router(config)#l2vpn
Router(config-l2vpn)#bridge group bg1
Router(config-l2vpn-bg)#bridge-domain bd1
Router(config-l2vpn-bg-bd)#interface BundleEther1.100
Router(config-l2vpn-bg-bd-ac)#evi 100
Router(config-l2vpn-bg-bd-evi)#commit

/* Configure bridge domain and associate with attachment circuits, EVPN instance and BVI
*/
Router(config)#l2vpn
Router(config-l2vpn)#bridge group bg2
Router(config-l2vpn-bg)#bridge-domain bd2
Router(config-l2vpn-bg-bd)#interface TenGigE0/0/0/38.200
Router(config-l2vpn-bg-bd-ac)#routed interface BVI200
Router(config-l2vpn-bg-bd-bvi)#evi 200
Router(config-l2vpn-bg-bd-bvi)#commit
Router(config-l2vpn-bg-bd-bvi)#exit

Router(config)#l2vpn
Router(config-l2vpn)#bridge group bg3
Router(config-l2vpn-bg)#bridge-domain bd3
Router(config-l2vpn-bg-bd)#interface TenGigE0/0/0/38.202
Router(config-l2vpn-bg-bd-ac)#routed interface BVI202
Router(config-l2vpn-bg-bd-bvi)#evi 202
Router(config-l2vpn-bg-bd-bvi)#commit

/* Configure EVPN VPWS */
Router#configure
Router(config)#router bgp 100
Router(config-bgp)#neighbor-group spines
Router(config-bgp-nbrgrp)#remote-as 100
Router(config-bgp-nbrgrp)#update-source loopback0
Router(config-bgp-nbrgrp)#address-family ipv4 labeled-unicast multipath
Router(config-bgp-nbrgrp-af)#exit
Router(config-bgp-nbrgrp)#address-family ipv6 labeled-unicast multipath
Router(config-bgp-nbrgrp-af)#exit
Router(config-bgp-nbrgrp)#address-family l2vpn evpn

```

```
Router(config-bgp-nbrgrp-af) #exit
Router(config-bgp-nbrgrp) #exit
Router(config-bgp) #neighbor 209.165.200.225
Router(config-bgp-nbr) #use neighbor-group spines
Router(config-bgp-nbr) #commit
Router(config-bgp-af) #exit
Router(config-bgp) #exit
Router(config) #l2vpn
Router(config-l2vpn) #xconnect group aa-evpn-vpws
Router(config-l2vpn-xc) #p2p vpws_513
Router(config-l2vpn-xc-p2p) #interface Bundle-Ether1.513
Router(config-l2vpn-xc-p2p) #neighbor evpn evi 513 target 513 source 513
Router(config-l2vpn-xc-p2p) # commit
```

実行コンフィギュレーション

この項では、フラッディング無効化の実行コンフィギュレーションを示します。

```
/* Configure IGP */
router ospf 1
  router-id 209.165.201.1
  area 10
    interface Loopback0
    !
    interface TenGigE0/0/0/1
    !
    interface TenGigE0/0/0/17
    !
  !
/* Configure BGP */
router bgp 100
  router-id 209.165.201.1
  bgp graceful-restart
  address-family ipv4 unicast
    redistribute connected
    network 209.165.200.225/27
    allocate-label all
  address-family ipv6 unicast
    allocate-label all
  neighbor-group spines
    remote-as 100
    update-source loopback0
    address-family ipv4 labeled-unicast multipath
    !
    address-family ipv6 labeled-unicast multipath
    !
    address-family l2vpn evpn
      advertise vpv4 unicast re-originated
      advertise vpv6 unicast re-originated
    !
  neighbor 209.165.200.225
    use neighbor-group spines
  !

/* Configure VPN4 address-family */
router bgp 100
  router-id 209.165.201.1
  ibgp policy out enforce-modifications
  address-family vpv4 unicast
  !

/* Configure EVPN instance and ESI */
```

```
evpn
evi 100
  advertise-mac
  !
interface Bundle-Ether1
  ethernet-segment
  identifier type 0 aa.aa.aa.aa.aa.aa.aa.aa.ac
  bgp route-target 0011.0011.0012
  !
  !
  !

/* Configuring BVI (IRB) Interface */
configure
interface BVI200
  ipv4 address 192.0.2.1 255.255.255.0

/* Configure VRF */
vrf vpn2
  address-family ipv4 unicast
  import route-target 81:2
  !
  !
  !
  address-family ipv6 unicast
  import route-target 81:2
  !
  !
  !

/* Configure BVI with VRF */
interface BVI200
  host-routing
  vrf vpn72
  ipv4 address ipv4 address ipv4 address 192.0.2.1 255.255.255.0
  mac-address 10.1111.1
  !

/* Configure VRF under BGP */
router bgp 100
vrf vpn2
  rd 102:2
  address-family ipv4 unicast
  label mode per-vrf
  maximum-paths ibgp 8
  redistribute connected
  !
  address-family ipv6 unicast
  label mode per-vrf
  maximum-paths ibgp 8
  redistribute connected
  !
  !

/* Configure bridge domain and associate with attachment circuits and EVPN instance */
l2vpn
bridge group bg1
  bridge-domain b1
  interface Bundle-Ether1.100
  !
  evi 100

/*
bridge group bg2
```



```

bridge-domain bd2
 interface TenGigE0/0/0/38.200
 !
 routed interface BVI200
 !
 evi 200
 !
 !

/* Configurige bridge domain and associate with attachment circuits, EVPN instance and
BVI */
bridge group bg3
 bridge-domain bd3
 interface TenGigE0/0/0/38.202
 !
 routed interface BVI202
 !
 evi 202
 !
 !
 !

/* Configure EVPN VPWS */
configure
router bgp 100
 neighbor-group spines
 remote-as 100
 update-source Loopback0
 address-family ipv4 labeled-unicast multipath
 !
 address-family ipv6 labeled-unicast multipath
 !
 address-family l2vpn evpn

neighbor 209.165.200.225
 use neighbor-group spines
 !
 !
l2vpn
 xconnect group aa-evpn-vpws
 p2p vpws_513
 interface Bundle-Ether1.513
 neighbor evpn evi 513 target 513 source 513

```

確認

BGP-LU アンダーレイ機能により EVPN ブリッジングと VPWS サービスが正しく設定されていることを確認します。

```

Router#show cef vrf AIM9 10.0.0.1
Tue Jan 20 22:00:56.233 UTC
10.0.0.1/8, version 4, internal 0x5000001 0x0 (ptr 0x97d34b44) [1], 0x0 (0x0), 0x208
(0x98bef0f0)
Updated Mar 18 06:01:46.175
Prefix Len 32, traffic index 0, precedence n/a, priority 3
via 10.0.0.3/8, 7 dependencies, recursive, bgp-multipath [flags 0x6080]
 path-idx 0 NHID 0x0 [0x972c6f08 0x0]
 recursion-via-/32
 next hop VRF - 'default', table - 0xe0000000
 next hop 10.0.0.3/8 via 16448/0/21
 next hop 192.0.2.1/24 BE128 labels imposed {16111 64013 80002}

```

```

via 100.0.0.88/32, 7 dependencies, recursive, bgp-multipath [flags 0x6080]
path-idx 1 NHID 0x0 [0x972c6d68 0x0]
recursion-via-/32
next hop VRF - 'default', table - 0xe0000000
next hop 10.0.0.4/8 via 16488/0/21
next hop 192.0.2.1/24 BE128 labels imposed {16111 64009 80002}

```

```

Router#show l2vpn xconnect group aa-evpn-vpws xc-name vpws_513 detail
Wed Jan 22 13:14:05.878 GMT+4

```

```

Group aa-evpn-vpws, XC vpws_513, state is up; Interworking none
AC: Bundle-Ether1.513, state is up
Type VLAN; Num Ranges: 1
Rewrite Tags: []
VLAN ranges: [513, 513]
MTU 1500; XC ID 0xa00005f7; interworking none
Statistics:
  packets: received 0, sent 0
  bytes: received 0, sent 0
  drops: illegal VLAN 0, illegal length 0
EVPN: neighbor 24000, PW ID: evi 513, ac-id 513, state is up ( established )
XC ID 0xc0000001
Encapsulation MPLS
Source address 209.165.200.225
Encap type Ethernet, control word enabled
Sequencing not set
LSP : Up

```

EVPN	Local	Remote
Label	29045	1048577
MTU	1500	1500
Control word	enabled	enabled
AC ID	513	513
EVPN type	Ethernet	Ethernet

```

Router# show evpn internal-label vpn-id 513 detail
Tue Jan 28 13:22:19.110 GMT+4

```

VPN-ID	Encap	Ethernet Segment Id	EtherTag	Label
513	MPLS	0099.9900.0000.0000.9999	0	None
Multi-paths resolved: FALSE (Remote all-active)				
Multi-paths Internal label: None				
EAD/ES 10.0.0.5 0				
513	MPLS	0099.9900.0000.0000.9999	513	24000
Multi-paths resolved: TRUE (Remote all-active)				
Multi-paths Internal label: 24000				
EAD/ES 10.0.0.5 0				
EAD/EVI (P) 10.0.0.5 29104				
Summary pathlist:				
0xffffffff (P) 10.0.0.5 29104				

```

Router# show mpls forwarding labels 24000 hardware egress detail location 0/0/CPU0

```

```

Tue Jan 28 13:22:19.110 GMT+4
Label Label or ID Interface Switched
-----
24000 29104 EVPN:513 10.0.0.5 N/A
Updated: Oct 18 13:14:02.193
Version: 137839, Priority: 3
Label Stack (Top -> Bottom): { 29104 }

```

```

NHID: 0x0, Encap-ID: 0x140ea00000002, Path idx: 0, Backup path idx: 0, Weight: 0
MAC/Encaps: 0/4, MTU: 0
Packets Switched: 0

LEAF - HAL pd context :
sub-type : MPLS, ecd_marked:0, has_collapsed_ldi:0
collapse_bwalk_required:0, ecdv2_marked:0,
HW Walk:
LEAF:
  PI:0x308de88fb8 PD:0x308de89058 rev:5554240 type: MPLS (2)
  LEAF location: LEM
  FEC key: 0x23e0220000d71
  label action: MPLS_NOP
LWLDI:
  PI:0x309faa82c8 PD:0x309faa8308 rev:5554239 p-rev:5459825 5459825 ldi
type:EOS0_EOS1
  FEC key: 0x23e0220000d71 fec index: 0x0(0) num paths:2, bkup paths: 0
  Collpased IMP LDI: ECD_MARKED
  IMP pattern:3
  PI:0x309faa82c8 PD:0x309faa8308 rev:5554239 p-rev:5459825 5459825
  FEC key: 0x257c720000d71 fec index: 0x20000003(3) num paths:2
  Path:0 fec index: 0x20018f14(102164) DSP fec index: 0x200001f8(504),
    MPLS encap key: 0xf1b00000400140ea MPLS encap id: 0x400140ea Remote: 0
    Label Stack: 29104 16012 dpa-rev:55458217
  Path:1 fec index: 0x20018f15(102165) DSP fec index: 0x200001f9(505),
    MPLS encap key: 0xf1b00000400140eb MPLS encap id: 0x400140eb Remote: 0
    Label Stack: 29104 16012 dpa-rev:55458218

REC-SHLDI HAL PD context :
ecd_marked:10, collapse_bwalk_required:0, load_shared_lb:0

RSHLDI:
  PI:0x3093d16af8 PD:0x3093d16bc8 rev:5494421 dpa-rev:36033167 flag:0x1
  FEC key: 0x249e440000d71 fec index: 0x2001c169(115049) num paths: 1
  p-rev:5459825
  Path:0 fec index: 0x2001c169(115049) DSP fec index: 0x200001f8(504),

LEAF - HAL pd context :
sub-type : MPLS, ecd_marked:1, has_collapsed_ldi:0
collapse_bwalk_required:0, ecdv2_marked:0,
HW Walk:
LEAF:
  PI:0x308de433b8 PD:0x308de43458 rev:5459864 type: MPLS (2)
  LEAF location: LEM
  FEC key: 0
LWLDI:
  PI:0x309ffe9798 PD:0x309ffe97d8 rev:5459825 p-rev:4927729 4927729 ldi
type:IMP_EOS0_EOS1
  FEC key: 0x1a1c740000d71 fec index: 0x0(0) num paths:2, bkup paths: 0
  IMP LDI: ECD_MARKED SERVICE_MARKED
  IMP pattern:3
  PI:0x309ffe9798 PD:0x309ffe97d8 rev:5459825 p-rev:4927729 4927729
  FEC key: 0x23e0220000d71 fec index: 0x20000002(2) num paths:2
  Path:0 fec index: 0x2001f8b4(129204) DSP fec index: 0x200001f8(504),
    MPLS encap key: 0xf1b0000040013ef0 MPLS encap id: 0x40013ef0 Remote: 0
    Label Stack: 16012 dpa-rev:35993054. <<< LU Label>>>>
  Path:1 fec index: 0x2001f8b5(129205) DSP fec index: 0x200001f9(505),
    MPLS encap key: 0xf1b0000040013ef2 MPLS encap id: 0x40013ef2 Remote: 0
    Label Stack: 16012 dpa-rev:35993055 <<< LU Label>>>>

```

```

REC-SHLDI HAL PD context :
ecd_marked:10, collapse_bwalk_required:0, load_shared_lb:0

RSHLDI:
  PI:0x308dd32c38 PD:0x308dd32d08 rev:4927729 dpa-rev:35005343 flag:0x3
  FEC key: 0x1alc740000d71 fec index: 0x20000813(2067) num paths: 2
  p-rev:4926086
  Path:0 fec index: 0x2001eefd(126717) DSP fec index: 0x200001f8(504),
  Path:1 fec index: 0x2001eefe(126718) DSP fec index: 0x200001f9(505),
LEAF - HAL pd context :
  sub-type : MPLS, ecd_marked:1, has_collapsed_ldi:0
  collapse_bwalk_required:0, ecdv2_marked:0,
HW Walk:
LEAF:
  PI:0x308dde33b8 PD:0x308dde3458 rev:4924403 type: MPLS (2)
  LEAF location: LEM
  FEC key: 0

LWLDI:
  PI:0x308b04ea58 PD:0x308b04ea98 rev:4924400 p-rev:4924389 4924389 4924389 4924389
ldi type:IMP_EOS0_EOS1
  FEC key: 0x1a75340000d71 fec index: 0x0(0) num paths:4, bkup paths: 0
  IMP LDI: ECD_MARKED
  IMP pattern:3
  PI:0x308b04ea58 PD:0x308b04ea98 rev:4924400 p-rev:4924389 4924389 4924389 4924389

  FEC key: 0x1a74720000d71 fec index: 0x200001f8(504) num paths:4
  Path:0 fec index: 0x2001ee86(126598) DSP:0x21
    MPLS encap key: 0xf1b0000040015878 MPLS encap id: 0x40015878 Remote: 0
    Label Stack: 16005 dpa-rev:34999715
  Path:1 fec index: 0x2001ee87(126599) DSP:0x22
    MPLS encap key: 0xf1b000004001587a MPLS encap id: 0x4001587a Remote: 0
    Label Stack: 16005 dpa-rev:34999716
  Path:2 fec index: 0x2001ee88(126600) DSP:0xc000002
    MPLS encap key: 0xf1b0000040016980 MPLS encap id: 0x40016980 Remote: 0
    Label Stack: 16005 dpa-rev:34989935
  Path:3 fec index: 0x2001ee89(126601) DSP:0xc000003
    MPLS encap key: 0xf1b00000400157fc MPLS encap id: 0x400157fc Remote: 0
    Label Stack: 16005 dpa-rev:34989936

SHLDI:
  PI:0x30927740c8 PD:0x3092774198 rev:4924389 dpa-rev:34999705 flag:0x0
  FEC key: 0x1a75340000d71 fec index: 0x200001ff(511) num paths: 4 bkup paths:
  0
  p-rev:4924311 4924329 8779 4920854
  Path:0 fec index: 0x2001ee8f(126607) DSP:0x21 Dest fec index: 0x0(0)
  Path:1 fec index: 0x2001ee90(126608) DSP:0x22 Dest fec index: 0x0(0)
  Path:2 fec index: 0x2001ee91(126609) DSP:0xc000002 Dest fec index: 0x0(0)
  Path:3 fec index: 0x2001ee92(126610) DSP:0xc000003 Dest fec index: 0x0(0)
TX-NHINFO:
  PI: 0x308dc51298 PD: 0x308dc51318 rev:4924311 dpa-rev:34994174 Encap
hdl: 0x3091632e98
  Encap id: 0x40010003 Remote: 0 L3 int: 1670 flags: 0x3
  npu_mask: 0x1 DMAC: 84:78:ac:2d:f8:1f

  TX-NHINFO:
  PI: 0x308dc51c20 PD: 0x308dc51ca0 rev:4924329 dpa-rev:34994264 Encap
hdl: 0x30916332c8
  Encap id: 0x40010001 Remote: 0 L3 int: 1679 flags: 0x3
  npu_mask: 0x1 DMAC: d4:6d:50:7c:f9:4d

  TX-NHINFO:
  PI: 0x308dc51ff0 PD: 0x308dc52070 rev:8779 dpa-rev:61964 Encap hdl:
0x308e9f4980

```

```

Encap id: 0x40010007 Remote: 0 L3 int: 1728 flags: 0x807
npu_mask: 0x1 DMAC: 84:78:ac:2d:f8:22

TX-NHINFO:
  PI: 0x308dc51480 PD: 0x308dc51500 rev:4920854 dpa-rev:34989846 Encap
hdl: 0x308e9f4db0
  Encap id: 0x40010005 Remote: 0 L3 int: 1727 flags: 0x807
  npu_mask: 0x1 DMAC: 40:55:39:11:37:39

LEAF - HAL pd context :
sub-type : MPLS, ecd_marked:1, has_collapsed_ldi:0
collapse_bwalk_required:0, ecdv2_marked:0,
HW Walk:
LEAF:
  PI:0x308dde35b8 PD:0x308dde3658 rev:4926089 type: MPLS (2)
  LEAF location: LEM
  FEC key: 0

LWLDI:
  PI:0x308b04eb48 PD:0x308b04eb88 rev:4926086 p-rev:4924389 4924389 4924389 4924389
ldi type:IMP_EOS0_EOS1
  FEC key: 0x1a75340000d71 fec index: 0x0(0) num paths:4, bkup paths: 0
  IMP LDI: ECD_MARKED
  IMP pattern:3
  PI:0x308b04eb48 PD:0x308b04eb88 rev:4926086 p-rev:4924389 4924389 4924389 4924389

  FEC key: 0x1a74820000d71 fec index: 0x200001f9(505) num paths:4
  Path:0 fec index: 0x2001ee81(126593) DSP:0x21
    MPLS encap key: 0xf1b000004001587c MPLS encap id: 0x4001587c Remote: 0
    Label Stack: 16006 dpa-rev:35002526
  Path:1 fec index: 0x2001ee82(126594) DSP:0x22
    MPLS encap key: 0xf1b000004001588a MPLS encap id: 0x4001588a Remote: 0
    Label Stack: 16006 dpa-rev:35002527
  Path:2 fec index: 0x2001ee83(126595) DSP:0xc000002
    MPLS encap key: 0xf1b0000040016964 MPLS encap id: 0x40016964 Remote: 0
    Label Stack: 16006 dpa-rev:34991843
  Path:3 fec index: 0x2001ee84(126596) DSP:0xc000003
    MPLS encap key: 0xf1b00000400157fe MPLS encap id: 0x400157fe Remote: 0
    Label Stack: 16006 dpa-rev:34991844

SHLDI:
  PI:0x30927740c8 PD:0x3092774198 rev:4924389 dpa-rev:34999705 flag:0x0
  FEC key: 0x1a75340000d71 fec index: 0x200001ff(511) num paths: 4 bkup paths:
0
  p-rev:4924311 4924329 8779 4920854
  Path:0 fec index: 0x2001ee8f(126607) DSP:0x21 Dest fec index: 0x0(0)
  Path:1 fec index: 0x2001ee90(126608) DSP:0x22 Dest fec index: 0x0(0)
  Path:2 fec index: 0x2001ee91(126609) DSP:0xc000002 Dest fec index: 0x0(0)
  Path:3 fec index: 0x2001ee92(126610) DSP:0xc000003 Dest fec index: 0x0(0)

TX-NHINFO:
  PI: 0x308dc51298 PD: 0x308dc51318 rev:4924311 dpa-rev:34994174 Encap
hdl: 0x3091632e98
  Encap id: 0x40010003 Remote: 0 L3 int: 1670 flags: 0x3
  npu_mask: 0x1 DMAC: 84:78:ac:2d:f8:1f

TX-NHINFO:
  PI: 0x308dc51c20 PD: 0x308dc51ca0 rev:4924329 dpa-rev:34994264 Encap
hdl: 0x30916332c8
  Encap id: 0x40010001 Remote: 0 L3 int: 1679 flags: 0x3
  npu_mask: 0x1 DMAC: d4:6d:50:7c:f9:4d

TX-NHINFO:
  PI: 0x308dc51ff0 PD: 0x308dc52070 rev:8779 dpa-rev:61964 Encap hdl:
0x308e9f4980

```

```
Encap id: 0x40010007 Remote: 0 L3 int: 1728 flags: 0x807  
npu_mask: 0x1 DMAC: 84:78:ac:2d:f8:22
```

TX-NHINFO:

```
PI: 0x308dc51480 PD: 0x308dc51500 rev:4920854 dpa-rev:34989846 Encap  
hdl: 0x308e9f4db0
```

```
Encap id: 0x40010005 Remote: 0 L3 int: 1727 flags: 0x807  
npu_mask: 0x1 DMAC: 40:55:39:11:37:39
```

関連項目

[BGP-LU アンダーレイを介した EVPN ブリッジングおよび VPWS サービス \(58 ページ\)](#)

関連コマンド

- show l2vpn bridge-domain
- show bgp l2vpn evpn neighbors
- show cef vrf