# Cisco ASR 9000 シリーズ アグリゲーション サービス ルータ リリース 6.1.x L2VPN およびイーサネット サービス コンフィギュレーション ガイド

**初版**：2016 年 11 月 1 日


**シスコシステムズ合同会社**
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

# 目 次

# はじめに

リリース 6.1.2 以降、シスコは 64 ビット Linux ベースの IOS XR オペレーティング システムのサポートを導入しています。32 ビット環境と 64 ビット環境の間で、広範な機能パリティが維持されます。特に明記されていない限り、このドキュメントの内容は両方の環境に適用されます。Cisco IOS XR 64 ビットの詳細については、ドキュメント『Release Notes for Cisco ASR 9000 シリーズルータ, Release 6.1.2』を参照してください。

このガイドでは、Cisco ASR 9000 シリーズ ルータの設定について説明します。『 *L2VPN and Ethernet Services Configuration Guide for Cisco ASR 9000 Series RoutersL2VPN and Ethernet Services Configuration Guide for Cisco NCS 560 Series Routers*』の「はじめに」には、次の項が含まれています。



## マニュアルの変更履歴

次の表に、初版後このマニュアルに加えられた技術的な変更の履歴を示します。

| 日付 | 変更点 |
|---|---|
| 2016 年 11 月 | このマニュアルの初版 |

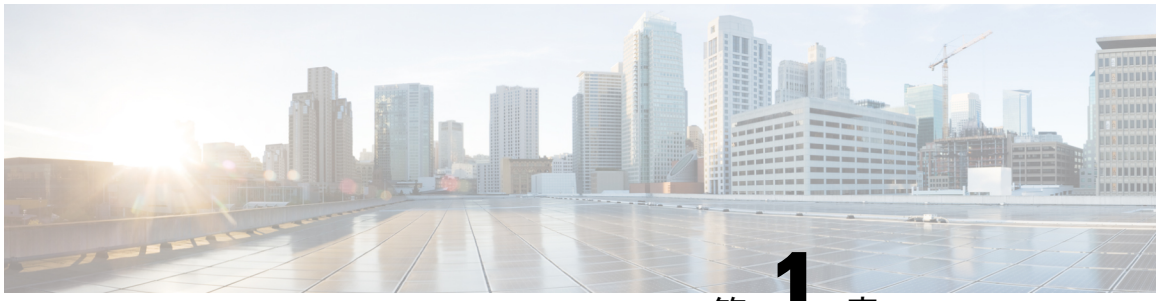
## マニュアルの入手方法およびテクニカル サポート

マニュアルの入手方法、テクニカル サポート、その他の有用な情報について、次の URL で、毎月更新される『What's New in Cisco Product Documentation』を参照してください。シスコの新規および改訂版の技術マニュアルの一覧も示されています。

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

『What's New in Cisco Product Documentation』は RSS フィードとして購読できます。また、リーダーアプリケーションを使用してコンテンツがデスクトップに直接配信されるように設定する

こともできます。RSS フィードは無料のサービスです。シスコは現在、RSS バージョン 2.0 をサポートしています。



第 1 章

# 新規および変更された VPN 機能

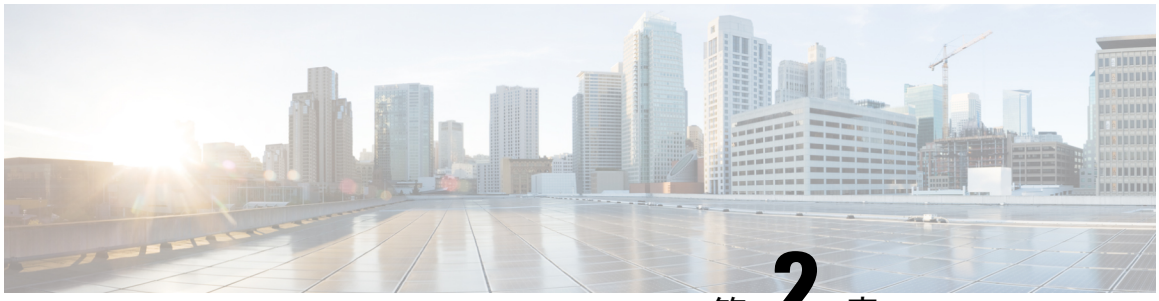
この表では、『L2VPN and Ethernet Services Configuration Guide for Cisco ASR 9000 Series Routers』の新機能および変更された機能の情報の概要と、その機能がどこでドキュメント化されているかを示します。


- 新機能および変更された機能に関する情報 （1 ページ）


## 新機能および変更された機能に関する情報

| 機能 | 説明 | 変更が行われたリリース | 参照先 |
|---|---|---|---|
| EVPN VXLAN レイヤ 2 Data Center Interconnect ゲートウェイ | Cisco ASR 9000 シリーズルータはデータセンター相互接続（DCI）レイヤ 2 ゲートウェイとして機能し、MPLS ベースの L2VPN ネットワークを介して EVPN VXLAN ベースのデータセンター間にレイヤ 2 接続を提供します。 | リリース 6.1.2 | EVPN VXLAN レイヤ 2 Data Center Interconnect ゲートウェイ （535 ページ） |

| 機能 | 説明 | 変更が行われたリリース | 参照先 |
|---|---|---|---|
| EVPN ESI マルチパス | EVPN イーサネットセグメント識別子（ESI）マルチパス機能は、アクティブ-アクティブのデュアルホーム接続 ToR と DCI へのマルチパス トラフィックをサポートし、データセンター内に冗長接続を実現します。 | リリース 6.1.2 | VXLAN の EVPN ESI マルチパス：EVI ベースのロードバランシング （536 ページ） |
| L2TPv3 over IPv4 | この機能が導入されます。 | リリース 6.1.2 | L2TPv3 over IPv4 （102 ページ） |
| L2VPN のサービスパス設定 | この機能が導入されます。 | リリース 6.1.2 | L2VPN のサービスパス設定 （233 ページ） |
| L2VPN ルートポリシー | この機能が導入されます。 | リリース 6.1.2 | L2VPN ルートポリシー （234 ページ） |
| EVPN VPWS ブリッジドメイン（BD）アクセス<br><br>PWHE への EVPN VPWS | EVPN-VPWS 機能が、ブリッジドメインアクセスおよび PWHE インターフェイスの設定をサポートするように拡張されています。 | リリース 6.1.2 | EVPN 仮想プライベート ワイヤ サービス（VPWS） （111 ページ） |



第 2 章

# キャリアイーサネットモデル

この章では、レイヤ2（L2）の機能および規格について紹介します。この章では、L2VPN機能を設定する方法についても説明します。

分散ギガビットイーサネットおよび10ギガビットイーサネットのアーキテクチャと機能により、サービスプロバイダーは、ルータとPOP内の他のシステム（コアルータ、エッジルータ、L2スイッチ、レイヤ3（L3）スイッチなど）を相互接続するために設計された、高密度、高帯域幅のネットワーキングソリューションを提供でき、その一方でネットワークのスケーラビリティおよびパフォーマンスも提供されます。

（注） 管理イーサネット インターフェイスの設定情報については説明しません。管理イーサネット インターフェイスを設定し、Telnet サーバを有効にする場合は、『Cisco ASR 9000 Series Aggregation Services Routers Getting Started GuideCisco NCS 6000 Series Routers Getting Started Guide』を参照してください。ルーティングのために管理イーサネット インターフェイスを設定する場合、または管理イーサネットインターフェイスの設定を変更する場合は、『*Interface and Hardware Component Configuration Guide for Cisco ASR 9000 Series Routers*』の「Advanced Configuration and Modification of the Management Ethernet Interface on the Cisco ASR 9000 Series Router」の章を参照してください。

### イーサネット インターフェイス設定の機能履歴

| リリース | 変更内容 |
|---|---|
| リリース 3.7.2 | この機能は、Cisco ASR 9000 シリーズ ルータで導入されました。 |
| リリース 4.1.1 | バンドルインターフェイスの EFP のスケーラビリティが導入されました。 |

• 次の作業 （42 ページ）

# レイヤ 2 イーサネット インターフェイスを設定するための前提条件

イーサネットインターフェイスを設定する前に、次のタスクと条件が満たされていることを確認してください。

- 適切なタスク ID を含むタスクグループに関連付けられているユーザグループに属している必要があります。このコマンドリファレンスには、各コマンドに必要なタスク ID が含まれます。

  ユーザグループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

- 次のラインカードの少なくとも 1 つがルータに取り付けられていることを確認してください。
  - 4 ポート 10 ギガビットイーサネット（4 x 10 GE）ラインカード
  - 8 ポート 10 ギガビットイーサネット（4 x 10 GE）ラインカード
  - 40 ポート 1 ギガビット イーサネット ラインカード
- インターフェイスの IP アドレスがわかっていること。
- 汎用インターフェイス名に汎用表記法の *rack/slot/module/port* を適用する方法を理解しています。

# レイヤ 2 理論と規格準拠

イーサネット インターフェイスを設定するには、次の概念を理解している必要があります。

## イーサネット テクノロジーの概要

イーサネットはIEEE 802.3国際規格によって定義されています。イーサネットによって、同軸ケーブル、ツイストペアケーブル、または光ファイバケーブルで、最大 1024 ノードの接続が可能になります。

Cisco ASR 9000 シリーズルータは、ギガビットイーサネット（1000 Mbps）インターフェイスおよび 10 ギガビットイーサネット（10 Gbps）インターフェイスをサポートしています。

## キャリア イーサネット サービス

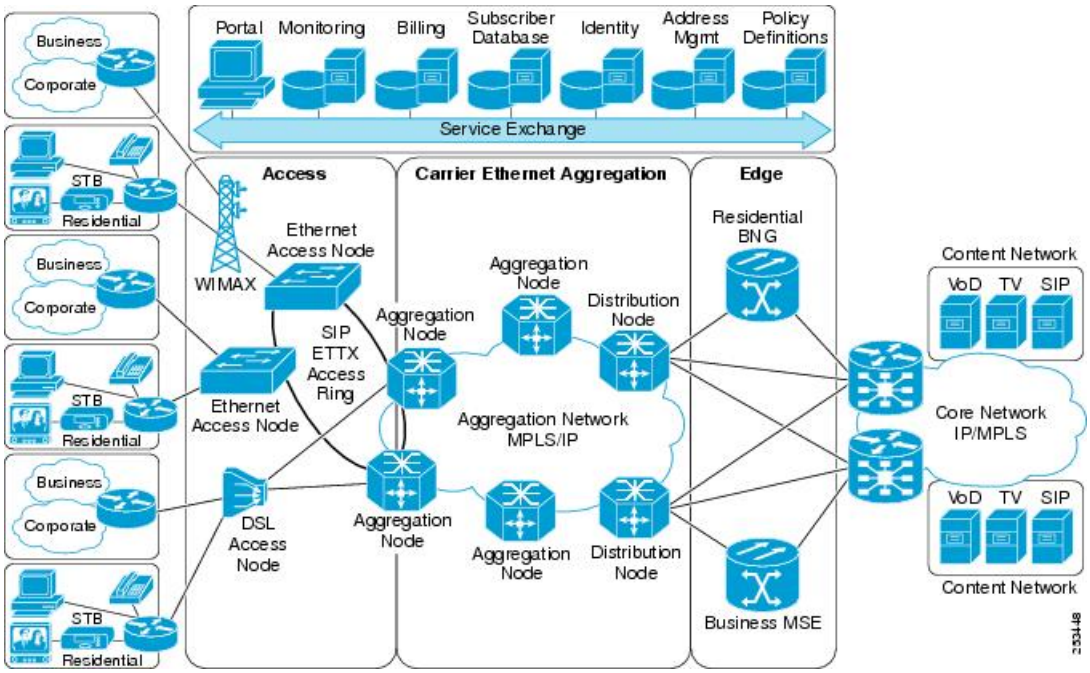
シスコが推奨するメトロ イーサネット フォーラム（MEF）キャリアイーサネット（CE）2.0 サービス：

- E-LINE：E-Line は 2 つの UNI のみを接続するポイントツーポイント イーサネット サービスです。
- E-LAN：E-LAN は多数（2 つ以上）の UNI を接続するマルチポイントツーマルチポイント サービスで、サイトにフルメッシュ接続を提供します。
- E-TREE：E-Tree は多数の UNI を接続するルーテッド マルチポイント サービスで、サイトにハブアンドスポーク マルチポイント接続を提供します。
- E-ACCESS：E-Access サービスは、少なくとも 1 つの UNI OVC エンドポイントと 1 つの ENNI エンドポイントを持つ、OVC ベースのサービスです。

イーサネット WAN（EWAN）について説明する際に、次の用語を使用します。

- CE（カスタマーエッジ）：サービスプロバイダーに接続するカスタマーデバイス
- PE（プロバイダーエッジ）：カスタマーに接続するサービス プロバイダー デバイス
- UNI：CE と PE 間の接続
- AC：CE を PE に接続する物理または仮想回線
- 多重化 UNI：複数の VLAN フローをサポートする UNI
- 疑似回線：サービス プロバイダー ネットワーク内のエンドツーエンド パスを示すために使用する用語

**図 *1 : EWAN* の用語**




## イーサネットワイヤサービス

イーサネットワイヤサービスは、ポイントツーポイントのイーサネットセグメントをエミュレートするサービスです。これは、プロバイダーエッジがレイヤ 2 で動作し、通常レイヤ 2

ネットワークで実行される以外、イーサネット専用回線（EPL）、レイヤ 1 ポイントツーポイント サービスに似ています。EWS は特定の UNI で受信されたすべてのフレームをカプセル化し、フレームに含まれる内容を参照せずに、これらのフレームを単一出力UNIに転送します。このサービスの動作は EWS を VLAN タグ付きフレームで使用できることを示します。VLAN タグは、一部の例外を除いて EWS（ブリッジ プロトコル データ ユニット（BPDU））に対して透過的です。これらの例外には、IEEE 802.1x、IEEE 802.2ad、および IEEE 802.3x が含まれます。これは、これらのフレームがローカルで意味を持ち、カスタマーとサービスプロバイダーの両方がそれらのフレームをローカルで終了できるよう支援されるためです。

サービスプロバイダーはインターフェイスでフレームを単純に受け取り、実際のフレームを参照せずにこれらを送信するため（ただし、形式と長さが特定のインターフェイスに適合していることは確認します）、EWS はカスタマーのイーサネットフレーム内にある VLAN タグに関与しません。

EWS は all-to-one バンドリングの概念に対応しています。つまり、EWS はポイントツーポイント回線の一方のエンドのポートと他方のエンドのポートをマッピングします。EWS はポート間サービスです。したがって、カスタマーが 1 つのスイッチまたはルータを n 個のスイッチまたはルータに接続する必要がある場合は、n 個のポートおよび n 個の疑似回線または論理回線が必要になります。

考慮すべき 1 つの重要なポイントは、EWS はイーサネットレイヤ 1 接続を広範にエミュレートするにもかかわらず、サービスは共有インフラストラクチャで提供され、したがって、すべてのインターフェイス帯域幅を常に使用できる可能性は低く、またそのようにする必要もないということです。EWS は、通常、多くのユーザが伝送パスのどこかで回線を共有する、サブラインレートサービスです。その結果、コストが EPL のコストよりも、ほとんどの場合、小さくなります。SP は、レイヤ 1 EPL とは異なり、特定契約の特定目的を達成するために、QoS およびトラフィック エンジニアリングを実装する必要があります。ただし、カスタマーアプリケーションに本当の意味でのワイヤレート透過サービスが必要な場合、DWDM（高密度波長分割多重）、CDWM（低密度波長分割多重）、SONET/SDH などの光送信デバイスを使用して提供される EPL サービスを検討する必要があります。

## イーサネット仮想専用回線

イーサネット仮想専用回線（EVPL）は、ポイントツーポイント接続を提供する点でEWSに似ています。EWS と EVPL の主な違いは、EVPL は、VLAN タグを使用して、宛先の異なる複数の疑似回線を 1 つのポートとの間で多重化する点です。つまり、EPL および EWS とは異なり、EVPL は、1 対多の多重化サービスです。サービス多重化は、複数の疑似回線が 1 つのアクセスインターフェイスまたは UNI を使用することを意味します。これらの回線は L2VPN 内、たとえばインターネットゲートウェイにおいて終端可能です。サービスユーザの観点からは、このサービス多重化機能により、インターフェイス使用の効率化、ケーブル設備の単純化、および追加インターフェイスに関連するメンテナンスコストの削減が実現します。

1 つのルータが他の n 個のルータに接続する上記と同じ例を使用した場合、送信元ルータには、EWS の場合と同様に、サービス用ポートは n 個ではなく 1 個のみ必要です。サービスは、ポート間で提供する必要はなく、論理的疑似回線間でも提供できます。EVPL の場合、各回線は、別のリモートロケーションで終端可能です。一方、EWS を使用した場合、すべてのフレームが 1 つの回線にマッピングされます。したがって、1 つの出力ポイントにマッピングされることとなります。

図 *2 : EVPL* サービス多重化の例：*1* ポート（左）をすべての宛先（右）に対し使用可能


EVPL では、フレームリレーと同様に、顧客のデバイスはサービス プロバイダー ネットワークに接続されている単一の物理ポートを介して複数の接続にアクセスできます。EVPL で提供されるサービスは、VLAN 番号が、フレームリレーのデータリンク接続識別子（DLCI）と同様の方法で、仮想回線識別子として使用される点において、フレームリレーと概念が類似していると考えることができます。EWS とは異なり、EVPL は BPDU を転送しません。これは、IEEE 802.1Q（VLAN タギング）がデフォルト VLAN で BPDU だけを送信するためです。ハブアンドスポーク ネットワークでは、最大で 1 つのスポークしか BPDU を受信しないため、ネットワークの残りの部分ではスパニングツリーは中断されます。したがって、EVPL は、BPDU を一切送信せず、イーサネット スパニング ツリーの代わりにルーティングプロトコルを実行します。こうしたルーティングプロトコルは、カスタマーおよびプロバイダーに対し、より優れた柔軟性、トラフィック決定特性、および付加価値サービスを提供します。

# イーサネット LAN サービス

イーサネット LAN サービス（E-LAN）は、マルチポイント接続モデルを提供する点において EWS および ERS と異なります。E-LAN サービスの定義は、IETF マルチポイントレイヤ 2 サービスワーキンググループ内でまだ検討中ですので注意してください。E-LAN はマルチポイントモデルを使用しますが、1 つの宛先へユニキャストパケットを転送できます。つまり、ポイントツーポイント接続をサポートします。エンドユーザには、ネットワークは、エンドツーエンド疑似回線リンクではなく、各カスタマーが独自の VLAN またはブロードキャストドメインを使用する巨大イーサネットスイッチのように見えます。

## E-LAN の例

E-LAN は特定のポイントツーポイント疑似回線にインターフェイスまたは VLAN をマッピングしません。代わりに、仮想イーサネットスイッチの動作を模倣します。つまり、E-LAN は顧客の MAC アドレスを使用して、サービスプロバイダーのネットワーク内の適切な出力 UNI にフレームを転送します。E-LAN は、イーサネットスイッチのサービス属性のエミュレートとインターフェイス アソシエーションのための送信元 MAC の学習、不明ブロードキャストおよびマルチキャストフレームのフラッディング、およびサービスユーザのスパニングツリープロトコルのモニタニング（オプション）を実行します。注意する 1 つの重要なポイントは、サービスプロバイダーは転送ネットワーク内でスパニングツリーを使用する場合があるにもかかわらず、サービスユーザのスパニングツリーとの相互動作がないことです。

このサービスは、L3 ではなく L2 で動作することを除き、MPLS VPN に動作が似ています。VPLS E-LAN は実行可能なソリューションですが、これのスケーラビリティと QoS 制御は、MPLS VPN のスケーラビリティと QoS 制御に比べると低品質です。さらに、サービスプロバイダーが付加価値レイヤ 3 サービスを提供することは、はるかに困難であり、不可能な場合もあります（これは本マニュアルで後述しています）。

## イーサネットフローポイント

イーサネットフローポイント（EFP）はメインインターフェイスのサブストリーム パーティションです。Cisco ASR 9000 シリーズルータでは、EFP はカプセル化ステートメントにより L2 サブインターフェイスとして実装されます。

## イーサネット仮想回線

イーサネット仮想回線（EVC）はポイントツーポイント トンネルです。Cisco ASR 9000 シリーズルータでは、EVC は疑似回線（PW）として実装されます。

## イーサネット OAM プロトコル

メトロエリアネットワーク（MAN）またはワイドエリアネットワーク（WAN）テクノロジーとしてのイーサネットでは、運用管理および保守（OAM）機能の実装によって大きな恩恵が得られます。OAM 機能により、サービスプロバイダーは MAN や WAN で接続の品質をモニタできます。サービスプロバイダーは、特定のイベントをモニタし、イベントに対しアクションを実行すること、および必要に応じて、トラブルシューティングのために特定のインターフェイスをループバックモードにすることができます。リンクの片側または両側をモニタするようにイーサネット OAM 機能を設定できます。

イーサネット OAM プロトコルの詳細については、『*Interface and Hardware Component Configuration Guide for Cisco ASR 9000 Series Routers*』の「Configuring Ethernet Interfaces」の章を参照してください。

# イーサネット インターフェイスでのレイヤ 2 VPN

L2VPN 接続は、IP または MPLS 対応 IP ネットワーク間の LAN の動作をエミュレートすることで、イーサネットデバイス間が共通の LAN セグメントに接続した場合と同様に通信できるようになります。

L2VPN の機能によって、サービスプロバイダー（SP）は地理的に離れたカスタマーサイトにも L2 サービスを提供できるようになります。通常、SP はアクセスネットワークを使用して、カスタマーをコアネットワークに接続します。このアクセスネットワークでは、イーサネット、フレームリレーなどの L2 テクノロジーが併用される場合があります。カスタマーサイトと近接した SP エッジルータ間の接続は、接続回線（AC）と呼ばれます。カスタマーからのトラフィックは、このリンク上で SP コアネットワークのエッジへ伝送されます。次に、SP コアネットワーク上の疑似接続のトンネルを介して、別のエッジルータへ伝送されます。このトラフィックはエッジルータによって別の AC へと伝送され、そこからカスタマーのリモートサイトへ伝送されます。

L2VPN の機能によって、異なる種類の L2 接続回線と疑似回線間の接続が可能になります。その結果、ユーザはさまざまなエンドツーエンドサービスを実装できるようになります。

Cisco IOS XR ソフトウェアは、2 つのイーサネット回線が接続されている、ポイントツーポイントおよびエンドツーエンドサービスをサポートしています。L2VPNイーサネットポートは、次の 2つの モードのいずれかで動作します。

- ポートモード：このモードでは、ポートに到達するすべてのパケットは、パケットに指定されている VLAN タグに関係なく、疑似回線上で送信されます。VLAN モードでは、l2transport コンフィギュレーション モードで設定が実行されます。
- VLAN モード：CE（カスタマーエッジ）の各 VLAN または PE（プロバイダーエッジ）リンクへのアクセスネットワークは個別の L2VPN 接続として設定できます（VC タイプ 4 または VC タイプ 5 を使用します）。VLAN 上で L2VPN を設定する方法については、このマニュアルの「キャリアイーサネットモデル」の章を参照してください。VLAN モードでは、個別のサブインターフェイスで設定を実行します。

スイッチングは次の 3 つの方法で実行できます。

- AC-to-PW：PE に到達したトラフィックは PW（疑似回線）を介してトンネリングされます（反対に、PW を介して到達したトラフィックは AC を介して送信されます）。これが最も一般的なシナリオです。
- ローカルの切り替え：1 つの AC 上で到達するトラフィックは、疑似接続を介さずに別の AC へ送出されます。
- PW の切り替え：PW に到達するトラフィックは AC へ送信されませんが、別の PW 上でコアに返信されます。

イーサネット インターフェイスで L2VPN を設定する場合、次の点に気を付けてください。

- L2VPN リンクは QoS（Quality of Service）および MTU（最大伝送単位）の設定をサポートしています。
- ネットワークの要件として、パケットを透過的に伝送することが必須の場合は、サービスプロバイダー（SP）ネットワークのエッジにおいてパケットの宛先 MAC（メディア アクセス コントロール）アドレスを変更することが必要になる可能性があります。こうすることで、SP ネットワークのデバイスによるパケットの消費が回避されます。

AC と疑似回線の情報を表示するには、**show interfaces** コマンドを使用します。

# ギガビット イーサネット プロトコル規格の概要

ギガビット イーサネット インターフェイスは、次のプロトコル規格をサポートしています。

- IEEE 802.3 物理イーサネット インフラストラクチャ
- IEEE 802.3ab 1000BASE-T ギガビットイーサネット
- IEEE 802.3z 1000 Mbps ギガビットイーサネット
- IEEE 802.3ae 10 Gbps イーサネット

各規格の詳細については、このマニュアルで後述します。

## IEEE 802.3 物理イーサネット インフラストラクチャ

IEEE 802.3 プロトコル規格では、接続するイーサネットの物理層とデータリンク層の MAC 下位層が定義されています。IEEE 802.3 では、多様な物理メディアで、また多様な速度でキャリア検知多重アクセス/衝突検出（CSMA/CD）アクセスを使用します。IEEE 802.3 規格は 10 Mbps イーサネットに対応します。IEEE 802.3 規格の拡張では、ギガビットイーサネット、10 ギガビットイーサネット、およびファストイーサネットの実装を規定しています。

## IEEE 802.3ab 1000BASE-T ギガビットイーサネット

IEEE 802.3abプロトコル規格、つまり銅線上のギガビットイーサネット（別名 1000BaseT）は、既存のファストイーサネット規格の拡張です。この拡張は、すでに設置されているカテゴリ 5e/6 ケーブル配線システム上のギガビットイーサネットの動作を規定しており、費用有効性の高いソリューションを実現できます。結果として、ファストイーサネットを実行する銅線ベースの環境では既存のインフラストラクチャ上でギガビットイーサネットも実行できるため、要求の厳しいアプリケーションでもネットワークのパフォーマンスが大幅に向上します。

## IEEE 802.3z 1000 Mbps ギガビットイーサネット

ギガビットイーサネットはイーサネットプロトコルの上で構築されますが、速度はファストイーサネットの 10 倍で、1000 Mbps（1 Gbps）に上がります。ギガビットイーサネットを使用すると、デスクトップで 10 Mbps または 100 Mbps、データセンターで最高 1000 Mbps までイーサネットを拡張できます。ギガビットイーサネットは IEEE 802.3z プロトコル規格に準拠します。

ネットワーク管理者は、現在のイーサネット規格と、すでに設置されているイーサネットおよびファストイーサネットのスイッチおよびルータのベースを利用することで、ギガビットイーサネットをサポートするために新しいテクノロジーのトレーニングや学習をし直す必要はなくなります。

## IEEE 802.3ae 10 Gbps イーサネット

国際標準化組織の開放型システム間相互接続（OSI）モデルでは、イーサネットは基本的に L2 プロトコルです。10 ギガビットイーサネットでは、IEEE 802.3 イーサネット MAC プロトコル、IEEE 802.3 イーサネットフレーム形式、およびIEEE 802.3 の最小および最大フレームサイズを使用します。10 Gbps イーサネットは IEEE 802.3ae プロトコル規格に準拠します。

イーサネットモデルに忠実だった 1000BASE-X と 1000BASE-T（ギガビットイーサネット）と同様に、10 ギガビットイーサネットも速度と距離の点でイーサネットが自然に発展した結果です。10 ギガビットイーサネットは全二重方式でファイバのみのテクノロジーなので、低速で半二重方式のイーサネットテクノロジーを定義する CSMA/CD プロトコルを使用した、通信事業者に影響される多重アクセスは必要ありません。他のどの点でも、10 ギガビットイーサネットは元のイーサネットモデルに忠実です。

## 一般的なイーサネット規格

- イーサネット II フレーム構成（別名 DIX）。
- IEEE 802.3 フレーム構成には、LLC および LLC/SNAP プロトコルフレーム形式も含まれます。
- IEEE 802.1d MAC ブリッジおよびスパニングツリー：この規格は、ブリッジング環境での MAC ラーニングと MAC エージングを指定します。また、元のスパニングツリープロトコルを定義します。MSTP も IEEE 802.1s および IEEE 802.1q で定義されています。
- IEEE 802.1q VLAN タギング：この規格は、VLAN タギングを定義し、またスイッチ間の従来の VLAN トランキングも定義します。技術的には、QinQ タギングおよび MSTP も定義します。Cisco ASR 9000 シリーズルータは ISL をサポートしません。

- IEEE 802.1ad プロバイダーブリッジ：この規格は 802.1q のサブセットであり、多くの場合 802.1ad と呼ばれます。Cisco ASR 9000 シリーズルータは、規格全体には準拠していませんが、規格の機能の大部分がサポートされます。

# MAC アドレス

MAC アドレスは、L2 でインターフェイスを識別する一意の 6 バイトアドレスです。

# イーサネット MTU

イーサネットの最大伝送ユニット（MTU）は、最大フレームのサイズから 4 バイトのフレームチェックシーケンス（FCS）を引いた値です。この MTU がイーサネットネットワークで伝送できるサイズです。パケットの宛先に到達するまでに経由する各物理ネットワークは、MTU が異なる可能性があります。

Cisco IOS XR ソフトウェアは、次の 2 つのタイプのフレーム転送プロセスをサポートします。

- IPV4 パケットのフラグメンテーション：このプロセスでは、ネクストホップの物理ネットワークの MTU 内に収まるように、必要に応じて IPv4 パケットが分割されます。

（注） IPv6 はフラグメンテーションをサポートしません。

- MTU の検出プロセスによる最大パケットサイズの決定：このプロセスは、すべての IPv6 デバイスと発信側の IPv4 デバイスに使用できます。このプロセスでは、分割せずに送信できる IPv6 または IPv4 パケットの最大サイズを、発信側の IP デバイスが決定します。最大パケットは、IP 発信元デバイスおよび IP 宛先デバイス間にあるすべてのネットワークの中で、最小 MTU と等値です。このパス内にあるすべてのネットワークの最小 MTU よりもパケットが大きい場合、そのパケットは必要に応じて分割されます。このプロセスによって、発信側のデバイスから大きすぎる IP パケットが送信されなくなります。

標準フレームサイズを超えるフレームの場合、ジャンボフレームのサポートが自動的にイネーブルになります。デフォルト値は標準フレームの場合は 1514、802.1Q タグ付きフレームの場合は 1518 です。これらの数値には、4 バイトの FCS は含まれません。

# イーサネット インターフェイスでのフロー制御

10 ギガビット イーサネット インターフェイスでのフロー制御は、フロー制御ポーズフレームを定期的に送信する処理で構成されます。この処理は、標準の管理インターフェイスで使用される通常の全二重および半二重のフロー制御とは根本的に異なります。Cisco ASR 9000 シリーズルータでは、入力および出力の両方でフロー制御はデフォルトではオフになっています。

# VRRP

仮想ルータ冗長プロトコル（VRRP）によって、静的なデフォルトのルーティング環境に固有の単一障害点が除外されます。VRRP は、仮想ルータの役割を LAN 上の VPN コンセントレータの1つに動的に割り当てるという、選択プロトコルを規定します。仮想ルータに割り当てる IP アドレスを制御する VRRP VPN コンセントレータはマスターと呼ばれ、送信されたパケットをその IP アドレスに転送します。マスターが使用不可になると、バックアップ VPN コンセントレータがマスターの役割を引き継ぎます。

VRRP の詳細については、『Cisco ASR 9000 Series Routers IP Addresses and Services Configuration Guide』の「Implementing VRRP」の章を参照してください。

# HSRP

Hot Standby Routing Protocol（HSRP）はシスコの独自プロトコルです。HSRP は障害の発生時にルータのバックアップを用意するルーティングプロトコルです。複数のルータが同じセグメントのイーサネット、FDDI、またはトークンリング ネットワークに接続されて連携し、LAN 上にある単一の仮想ルータを表わします。これらのルータは同じ IP アドレスおよび MAC アドレスを共有するため、ルータのいずれかに障害が発生した場合でも、LAN 上のホストはそのまま同じ IP アドレスおよび MAC アドレスにパケットを転送できます。ルーティングの担当デバイスの切り替えは、ユーザには検知されません。

HSRP は、特定の状況で IP トラフィックを中断しないフェールオーバーをサポートし、ホストからは単一のルータを使用しているように見え、使用している第1ホップのルータに障害が発生した場合でも接続を維持できるように設計されています。つまり、HSRP は、発信元のホストが第 1 ホップのルータの IP アドレスを動的に取得できない場合でも、第 1 ホップのルータの障害に対処できます。複数のルータが HSRP に参加し、連携して単一の仮想ルータであるように見せます。HSRP によって、確実に単一のルータが仮想ルータの代わりにパケットを転送します。エンドホストがそのパケットを仮想ルータに転送します。

パケットを転送するルータは、アクティブルータと呼ばれます。アクティブルータに障害が発生した場合、代わりになるスタンバイルータが選択されます。HSRP には、参加するルータの IP アドレスを使用して、アクティブルータとスタンバイルータを決定するメカニズムがあります。アクティブルータに障害が発生した場合、スタンバイルータが引き継ぐことができます。ホストの接続が長く切断することはありません。

HSRP はユーザ データグラム プロトコル（UDP）上で実行され、ポート番号 1985 を使用します。ルータは、プロトコルパケットの発信元アドレスとして仮想アドレスではなく実際の IP アドレスを使用するため、HSRP ルータは相互を識別できます。

HSRP の詳細については、『Cisco ASR 9000 Series Routers IP Addresses and Services Configuration Guide』の「Implementing HSRP」の章を参照してください。

# イーサネット インターフェイスのリンクのオートネゴシエーション

リンクのオートネゴシエーションによって、リンクセグメントを共有するデバイスは、最高のパフォーマンスモードの相互運用で自動的に設定されます。イーサネット インターフェイスで

リンクのオートネゴシエーションをイネーブルにするには、インターフェイス コンフィギュレーション モードで **negotiation auto** コマンドを使用します。ラインカードのイーサネット インターフェイスで、リンクのオートネゴシエーションはデフォルトでディセーブルです。

（注） **negotiation auto** コマンドは、ギガビットイーサネットインターフェイスだけで使用できます。

# イーサネットフローポイントとは

イーサネットフローポイント（EFP）とは、物理またはバンドルインターフェイスにおいて、トラフィックの分類に使用されるレイヤ 2 の論理サブインターフェイスです。

物理インターフェイスは、ギガビットイーサネット 0/0/0/1 または 10 ギガビットイーサネット 0/0/0/0 インターフェイスの場合があり、ラインカードのポートがあります。バンドルインターフェイスは、物理インターフェイスをグループ化することにより作成される仮想インターフェイスです。

たとえば、ギガビットイーサネット 0/0/0/1、10 ギガビットイーサネット 0/0/0/0 などの物理インターフェイスは、バンドルインターフェイスのメンバーとして設定できます。

物理インターフェイスをグループ化すると、以下が可能になります。

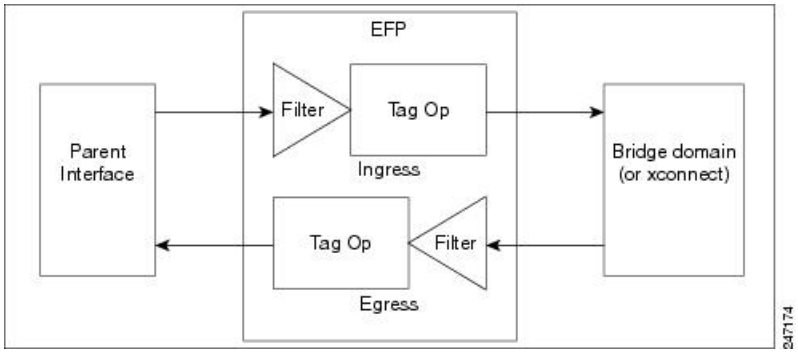
- ルーティングエントリの削減
- バンドルインターフェイスの帯域幅の増加
- バンドルメンバー間でのトラフィックのバランシング

EFP の特徴は、次のとおりです。

- EFP は、インターフェイスで Ethernet Virtual Connection（EVC; イーサネット仮想コネクション）の論理的な境界ポイントを表します。2 つ以上の UNI を関連付ける EVC では、EVC が通過するすべてのデバイスの各インターフェイスにフローポイントがあります。
- EFP は、特定のサービスのインスタンス化と見なすことができます。EFP は、一連のフィルタによって定義されます。これらのフィルタは、特定の EFP に属するフレームを分類するために、すべての入力トラフィックに適用されます。EFP フィルタは一連のエントリであり、各エントリはパケットの先頭部分に類似しています（送信元/宛先 MAC アドレスは無視します）。各エントリには、通常、0、1、または 2 つの VLAN タグが含まれます。パケットが、フィルタのエントリと同じタグで始まる場合、そのパケットはフィルタに一致することになります。パケットの先頭部分がフィルタのエントリに対応しない場合、パケットはフィルタに一致しません。
- EFP は次の 4 つの役割を果たします。
  - 特定のインターフェイスで特定のフローに属するすべてのフレームを識別します。
  - 入力および出力イーサネットヘッダー処理を実行します。
  - 識別されたフレームに機能を追加します。
  - オプションで、データパスでのフレームの転送方法を定義します。

ルータの各種インターフェイスにEFPが設定されている場合、トラフィックフローに対しさまざまな操作を実行できます。また、ルータの1つまたは複数の入力EFPから1つまたは複数の出力EFPに対しさまざまな方法でトラフィックをブリッジングまたはトンネリングできます。このトラフィックでは、VLAN ID、シングルまたはダブル（QinQ）カプセル化、およびEthertypeが併用されます。

**図 *3 : EFP* モデル**




入力のどのトラフィックをそのEFPに向けるか指定するために、EFPのサブインターフェイスを設定します。これは、入力で照合するVLAN、VLANの範囲、またはQinQタギングを指定することで行います。入力のすべてのトラフィックは、各EFPの一致条件と比較され、一致した場合には、そのEFPによって処理されます。EFPによって実行される処理では、VLAN IDを変更すること、VLANタグを追加または削除することや、Ethertypeを変更することができます。

## バンドルインターフェイスでの EFP のスケーラビリティの改善

次の2通りの方法でバンドルインターフェイスのEFPのスケーラビリティを改善できます。

- シャーシあたりのEFPの数を32000から64000に増やします。
- 単一ノードポイントで、ラインカードあたりのEFPの数を、物理インターフェイス スケーリングと同じスケールに増やします。

次に、ラインカードあたりのEFPのスケーラビリティを改善する例を示します。

バンドルインターフェイス スケーリングが4000、物理インターフェイス スケーリングが16000のBモジュール ライン カード タイプ[1]があるとします。Bモジュールの EFP のスケーラビリティは、バンドルあたり 4000 EFP のバンドルを 3 つ追加することで改善されます。

（注） バンドルインターフェイスに追加できる EFP の最大数は 4000 です。

ラインカードあたりの EFP の数は、16000 またはそれぞれ 4000 EFP の 4 つのバンドルに現在拡張されています。

## EFP CLI の概要

Cisco IOS XR は、EFP および EVC 設定のための構造化 CLI を実装しています。EFP を設定するために、通常、次のコマンドが使用されます。

- **l2transport** コマンド：このコマンドは、サブインターフェイス（または物理ポート、バンドルポートの親インターフェイス）を EFP として指定します。
- **encapsulation** コマンド：このコマンドは、一致基準を指定するために使用されます。
- **rewrite** コマンド：このコマンドは、VLAN タグの書き換え条件を指定するために使用されます。

# EFP 出力フィルタリング

EFP 出力フィルタリング機能は、EFP 出力トラフィックをフィルタリングする方法を提供し、指定するすべての EFP の出力トラフィックが入力一致条件に準拠するようにします。

入力 EFP は出力 EFP に似ています。ルータは、EFP の入力一致条件に一致するトラフィックを、その EFP のトラフィックとして送信するように設定されます。これが実行されないようにルータを設定することができます。不一致の出力 EFP トラフィックがルータを出ることを防ぐための予防手段はありません。

Cisco ASR 9000 シリーズルータでは、同じブリッジドメイン内の異なるポートで異なる VLAN を使用できます。これにより、ブリッジは、パケットの VLAN タグが設定されていないポートからパケットを転送できます。EFP 出力フィルタリングは、これを確認し、出力ポートで無効なパケットを廃棄します。

## EFP のフレームの識別

EFP は、イーサネットカプセル化に関係なく、指定ポートで特定フローに属するフレームを識別します。EFP は、フレームヘッダー内のフィールドに基づいてフローまたは EFP に柔軟にフレームをマッピングできます。

以下を使用して、フレームと EFP を照合できます。

- VLAN タグ
- MAC アドレス（送信元アドレス、宛先アドレス、または両方）
- 802.1p CoS ビット
- 上の複数の項目の論理的な組み合わせ：VLAN、MAC および CoS
- デフォルトの一致（つまり、特定の EFP に一致しない他のトラフィック）
- プロトコル Ethertype

次の項目を使用して、フレームと EFP を照合することはできません。

- 以下のような、最も外側のイーサネット フレーム ヘッダーおよび関連するタグの外部の情報
  - IPv4、IPv6、または MPLS のタグヘッダーのデータ
  - C-DMAC、C-SMAC、または C-VLAN
- 上の有効なフレーム一致の論理和：VLAN、MAC、および CoS

特定の一致条件について、以降の各項で詳しく説明します。

### VLAN タグの一致

次の表では、さまざまなカプセル化タイプとそれぞれに対応する EFP 識別子について説明します。

| カプセル化タイプ | EFP 識別子 |
|---|---|
| タグなし | **encapsulation** コマンドで **untagged** キーワードを使用する、入力物理インターフェイスまたはサブインターフェイスの静的設定。タグなしサブインターフェイスは 1 つのみ使用できます。タグなしサブインターフェイスが作成されると、トラフィックは、メインインターフェイスではなく、このインターフェイスに送られます。 |
| プライオリティタグ付きイーサネットフレーム | プライオリティタグ付きフレームは、VLAN ID がゼロの、単一 802.1Q VLAN ヘッダーを持つフレームとして定義されます。 |
| ネイティブ VLAN | Cisco ASR 9000 シリーズルータはネイティブ VLAN をサポートしていません。<br>使用するコマンド<br>**encapsulation dot1q** <*vlan-id*>, **untagged** |
| 単一タグ付きフレーム | 802.1Q カスタマータグ付きイーサネットフレーム |
| 二重タグ付きフレーム | 802.1Q（ethertype 0x8100）二重タグ付きフレーム<br>802.1ad 二重タグ付きフレーム<br>レガシー 0x9100 および 0x9200 二重タグ付きフレーム |

| カプセル化タイプ | EFP 識別子 |
|---|---|
| デフォルトのタギング | 最大一致のワイルドカードが設定されたEFP。目的は、同じ物理インターフェイスで他のEFPに一致しないトラフィックを受信することです。 |

特定の EFP にマッピングするフレームを定義するときに、ワイルドカードおよび VLAN の範囲を使用できます。EFP は、単一の VLAN タグ、VLAN タグの範囲、VLAN タグのスタック、または両方の組み合わせ（VLAN スタックとワイルドカード）に基づいてフローを区別できます。EFP は、EFP モデル、カプセル化非依存にする柔軟性を提供しています。また、新しいタギングまたはトンネリング方式を追加することで、EFP を拡張できるようになっています。

### MAC アドレスの一致

送信元 MAC アドレス、宛先 MAC アドレス、または両方を照合できます。いずれの場合も、MAC アドレスは完全に一致する必要があります。ワイルドカード一致または部分一致では不十分な場合があります。

### 802.1p CoS ビットの一致

1 つ以上の精確な CoS 一致が指定されます。CoS は 3 ビットのみであるため、8 種類の選択に制限されます。

### 論理結合

上記の一致基準はすべて、個別の条件すべてを満たすフレームを選択的に組み合わせることができます。

### デフォルトの一致

特定の EFP に一致していない他のすべてのトラフィックと一致する単一 EFP を定義できます。

### 照合順序と設定の検証

照合に使用する EFP の順序を決定できる、重複 EFP を設定できます。ただし、他の EFP または親トランクインターフェイスのサブインターフェイスと競合する EFP は、設定の検証でブロックする必要があります。

優先順位は、ハードウェアでの EFP 照合の適用方法に対し使用されます。このモデルは、あいまいな一致の前に、より精度の高い一致を処理するためのモデルです。

### 出力の動作

EFP 一致基準は出力でも使用でき、プラットフォームサポートに基づいて、EFP から出力できるフレームをポリシングできます。条件（送信元/宛先 MAC 一致基準は入れ替わります）に一致しないフレームはドロップされます。

## 機能の適用

フレームが特定の EFP に一致した後、適切な機能を適用できます。このコンテキストでは、「機能」とは、設定や QoS、ACL などによって指定されたフレーム操作を意味します。イーサネット インフラストラクチャは、機能オーナーが EFP に機能を適用できるように適切なインターフェイスを提供しています。そのため、EFP を表すために IM インターフェイスハンドルが使用され、これにより機能オーナーは、通常のインターフェイスまたはサブインターフェイスで機能が管理されるのと同じ方法で、EFP で機能を管理できます。

イーサネットインフラストラクチャの一部である EFP で適用できる唯一の L2 機能は、L2 ヘッダーのカプセル化の変更です。この L2 機能については、次の項で説明します。

### カプセル化の変更

EFP は、入力と出力の両方で、次の L2 ヘッダーのカプセル化の変更をサポートしています。

- 1 つまたは 2 つの VLAN タグのプッシュ処理
- 1 つまたは 2 つの VLAN タグのポップ処理

（注） この変更では、EFP に部分一致するタグのポップ処理のみ実行できます。

- 1 つまたは 2 つの VLAN タグの書き換え
  - 外部タグの書き換え
  - 2 つの外部タグの書き換え
  - 外部タグの書き換え、および追加タグのプッシュ処理
  - 外部タグの削除、および内部タグの書き換え

  各 VLAN ID 操作に対して、以下を指定できます。
- VLAN タグタイプ、つまり、C-VLAN、S-VLAN、または I-TAG。802.1Q C-VLAN タグの Ethertype は、dot1q tunneling type コマンドで定義されます。
- VLAN ID。0 は、プライオリティタグ付きフレームを生成するために、外部 VLAN タグに対し指定できます。

（注） タグの書き換えでは、以前のタグの CoS ビットを、802.1ad カプセル化フレームの DEI ビットと同じ方法で維持する必要があります。

## データ転送動作の定義

データパスで転送される特定のイーサネットフローに属するフレームを指定するために、EFP を使用できます。次の転送ケースが、Cisco IOS XR ソフトウェアでの EFP に対しサポートされます。

- L2 スイッチドサービス（ブリッジング）：EFP はブリッジドメインにマッピングされ、そこでフレームは宛先 MAC アドレスに基づいてスイッチングされます。これには、マルチポイントサービスが含まれます。
  - イーサネットとイーサネットのブリッジング
  - マルチポイントレイヤ 2 サービス
- L2 スイッチドサービス（AC と AC の xconnect）：これは、静的に確立されるポイントツーポイント L2 アソシエーションに対応し、MAC アドレスルックアップを必要としません。
  - イーサネットとイーサネットのローカルスイッチング：EFP は同じポートまたは別のポートの S-VLAN にマッピングされます。S-VLAN は同一にすること、または別にすることができます。
- トンネル型サービス（xconnect）：EFP はレイヤ 3 トンネルにマッピングされます。これは、ポイントツーポイント サービスのみに対応します。
  - Ethernet over MPLS（EoMPLS）
  - L2TPv3
- L2 終端サービス（レイヤ 3 サービスへのイーサネットアクセス）：EFP は、グローバルアドレスを持つ IP インターフェイス、または VRF に属する IP インターフェイスにマッピングされます（IP および MPLS レイヤ 3 VPN の両方が含まれます）。

# 802.1Q VLAN

VLAN とは、実際は異なる LAN セグメント上のデバイスでも、同じセグメントで接続している場合と同様に通信できるように設定された、1 つまたは複数の LAN 上にあるデバイスのグループです。VLAN は、物理接続ではなく論理接続に基づいているため、ユーザ管理、ホスト管理、帯域割り当て、およびリソースの最適化がとても柔軟です。

IEEE の 802.1Q プロトコル規格では、ブロードキャストおよびマルチキャストのトラフィックが必要以上の帯域を消費しないように、大規模なネットワークを小規模なパーツに分割することで問題に対処しています。また、内部ネットワークのセグメント間に、より高レベルのセキュリティを実現できます。

802.1Q 仕様は、イーサネット フレームに VLAN メンバーシップ情報を挿入する標準方式を確立します。

Cisco IOS XR ソフトウェアは、ギガビットイーサネットおよび 10 ギガビット イーサネット インターフェイスでの VLAN サブインターフェイスの設定をサポートしています。

## 802.1Q タグ付きフレーム

IEEE 802.1Q タグ ベースの VLAN は、MAC ヘッダーの特別なタグを使用し、ブリッジでのフレームの VLAN メンバーシップを識別できます。このタグは、VLAN および Quality of Service（QoS）のプライオリティの識別に使用されます。VLAN は、手動での入力によってスタティックに作成することも、Generic Attribute Registration Protocol（GARP）VLAN Registration プロトコル（GVRP）を介してダイナミックに作成することもできます。VLAN ID は、フレームを特定の VLAN に関連付けて、スイッチがネットワークでフレームを処理する必要があるという

情報を提供します。タグ付きフレームは、タグなしフレームよりも4バイト長く、イーサネットフレームのTypeおよびLengthフィールドにある2バイトのTag Protocol Identifier（TPID）フィールドと、イーサネットフレームのSource Addressフィールドの後ろから始まる2バイトのTag Control Information（TCI）が含まれます。

## サブインターフェイス

サブインターフェイスは、ハードウェアインターフェイス上に作成される論理インターフェイスです。これらのソフトウェア定義のインターフェイスにより、単一のハードウェアインターフェイス上でトラフィックを論理チャネルに分割することができ、また、物理インターフェイス上で帯域幅を効率的に利用することができます。

サブインターフェイスは、インターフェイス名の末尾に拡張を追加することで、他のインターフェイスと区別されます。たとえば、物理インターフェイスTenGigE 0/1/0/0上のイーサネットサブインターフェイス23は、TenGigE 0/1/0/0.23となります。

サブインターフェイスがトラフィックを渡すことができるようにするには、有効なタグ付きプロトコルのカプセル化とVLAN識別子の割り当てが必要です。すべてのイーサネットサブインターフェイスは常に、デフォルトで802.1Q VLANでカプセル化されます。ただし、VLAN識別子は明示的に定義する必要があります。

## サブインターフェイス MTU

サブインターフェイスの最大伝送ユニット（MTU）は、物理インターフェイスから継承されます。これには、802.1Q VLANタグに許可されている追加の4バイトも含まれます。

## イーサネットバンドルでの VLAN サブインターフェイス

イーサネットバンドルは、1つ以上のイーサネットポートのグループを集約し、1つのリンクとして扱うようにしたものです。単一のイーサネットバンドルに複数のVLANサブインターフェイスを追加できます。

イーサネットバンドルの設定方法については、このマニュアルの「リンクバンドルの設定」の章を参照してください。イーサネットバンドルにVLANサブインターフェイスを作成する手順は、物理イーサネットインターフェイスにVLANサブインターフェイスを作成する手順とまったく同じです。

イーサネットバンドルにVLANサブインターフェイスを作成する場合は、このマニュアルで後述する「802.1Q VLANインターフェイスの設定」セクションを参照してください。

## VLAN でのレイヤ 2 VPN

レイヤ2 VPN（L2VPN）機能によって、サービスプロバイダー（SP）は地理的に離れた顧客サイトにもL2サービスを提供できるようになります。詳細は、キャリアイーサネットモデル（3ページ）の章の「イーサネットインターフェイスでのレイヤ2 VPN」の項を参照してください。

VLAN接続回線（AC）を設定するための設定モデルは、基本のVLANの設定に使用するモデルに類似しています。ユーザはまずVLANサブインターフェイスを作成し、次にサブインター

フェイス コンフィギュレーション モードで VLAN を設定します。接続回路を作成するには、**interface** コマンド文字列に **l2transport** キーワードを含めて、そのインターフェイスが L2 インターフェイスであることを指定する必要があります。

VLAN AC は、L2VPN 操作の 3 つのモードをサポートします。

- 基本の Dot1Q 接続回線：この接続回線は、特定の VLAN タグで送受信されるすべてのフレームに対応します。
- QinQ 接続回線：この接続回線は、特定の外部 VLAN タグおよび特定の内部 VLAN タグで送受信されるすべてのフレームに対応します。QinQ は、2 つのタグのスタックを使用する Dot1Q の拡張です。
- Q-in-Any 接続回線：この接続回線は、内部 VLAN タグが L3 終端でない限り、特定の外部 VLAN タグおよび任意の内部 VLAN タグで送受信されるすべてのフレームに対応します。Q-in-Any は、ワイルドカード化を使用して任意の 2 番目のタグに一致させる QinQ の拡張です。

（注） Q-in-Any モードは、基本の Dot1Q モードを変化させたものです。Q-in-Any モードではフレームは基本の QinQ カプセル化が行われていますが、Q-in-Any モードでは内部タグは無関係です。ただし、いくつかの特定の内部 VLAN タグが特定のサービス用に使用される場合を除きます。たとえば、一般的なインターネットアクセスに L3 サービスを提供するために、あるタグが使用されることがあります。

CE-to-PE リンクの各 VLAN は、（VC タイプ 4 または VC タイプ 5 を使用する）独立した L2VPN 接続として設定できます。

VLAN に L2VPN を設定する場合は、次の事項に注意する必要があります。

- Cisco IOS XR ソフトウェアは、ラインカードごとに最大 4000 の接続回線をサポートしています。
- ポイントツーポイント接続では、2 つの接続回線を同じタイプにするべきではありません。たとえば、ポートモードのイーサネット接続回線は Dot1Q イーサネット接続回線に接続できます。
- 疑似回線は、VLAN モードまたはポートモードで実行できます。VLAN モードで実行される疑似接続に単一の Dot1Q タグを設定することができますが、ポートモードで実行される疑似接続にタグを設定することはできません。これらの異なるタイプの回路を接続するには、インターワーキングが必要です。この場合のインターワーキングは、タグのポップ、プッシュ、書き換えの形を取ります。L2VPN を使用するメリットは、まったく異なるタイプのメディアを接続するのに必要なインターワーキングを簡素化できることにあります。
- MPLS 疑似回線の両側にある接続回線は異なるタイプでもかまいません。この場合、接続回線の一方または両方のエンドで、疑似回線を行うための適切な変換が行われます。

接続回線と疑似回線の情報を表示するには、**show interfaces** コマンドを使用します。

（注） **show interfaces** コマンドの詳細については、『 *Interface and Hardware Component Configuration Guide for Cisco ASR 9000 Series Routers*』を参照してください。

# イーサネット インターフェイスでのレイヤ 2 機能の設定方法

（注） インターフェイスの設定の詳細については、『 *Interface and Hardware Component Configuration Guide for Cisco ASR 9000 Series Routers*』を参照してください。

## ギガビットイーサネットおよび10ギガビットイーサネットのデフォルト設定値

次の表は、ギガビットイーサネットまたは10ギガビットイーサネットのモジュラサービスカードおよびPCの脅威対策 PLIM でインターフェイスをイネーブルにしたときに表示される、デフォルトのインターフェイス設定パラメータを示します。

（注） インターフェイスを管理上のダウン状態にするには、**shutdown**コマンドを使用する必要があります。インターフェイスのデフォルトは no shutdown です。ルータにモジュラサービスカードを初めて挿入したときに、プリコンフィギュレーションが行われていない場合、設定マネージャによってshutdown項目が設定に追加されます。このshutdownを削除するには、noshutdownコマンドを入力します。

**表 *1:* ギガビットイーサネットおよび *10* ギガビット イーサネット モジュラ サービス カードのデフォルト設定値**

| パラメータ | 設定ファイルのエントリ | デフォルト値 | 制約事項 |
|---|---|---|---|
| フロー制御 | **flow-control** | 出力オン、入力オフ | なし |

| パラメータ | 設定ファイルのエントリ | デフォルト値 | 制約事項 |
|---|---|---|---|
| MTU | **mtu** | 1514バイト（通常のフレーム）<br>1518バイト（802.1Qタグ付きフレーム）<br>1522バイト（QinQフレーム） | なし |
| MACアドレス | **mac address** | ハードウェアバーンドイン アドレス（BIA[2]） | L3のみ |
| L2ポート | **l2transport** | off/L3 | L2サブインターフェイスにはL3メイン親インターフェイスが必要です。 |
| 出力フィルタリング | **Ethernet egress-filter** | off | なし |
| リンクネゴシエーション | **negotiation** | off | 物理メインインターフェイスのみ |
| Tunneling Ethertype | **tunneling ethertype** | 0X8100 | メインインターフェイスのみで設定されます。サブインターフェイスのみに適用されます。 |
| VLANタグの一致 | **encapsulation** | メインインターフェイスではすべてのフレーム。サブインターフェイスでは指定されたフレームのみ | encapsulationコマンドはサブインターフェイスのみ |

1. 制約事項はL2メインインターフェイス、L2サブインターフェイス、L3メインインターフェイス、インターフレックスL2インターフェイスなどに適用されます。
2. 組み込みのアドレス

# イーサネット インターフェイスの設定

イーサネット インターフェイスの設定については、『*Interface and Hardware Component Configuration Guide for Cisco ASR 9000 Series Routers*』を参照してください。

# 10 ギガビット イーサネット インターフェイスの設定

イーサネット インターフェイスを設定するには、次の作業を行います。

## 手順の概要

1. **configure interface TenGigE** [ *instance* ]
2. **l2transport**
3. **mtu** *bytes*
4. **no shutdown**
5. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure interface TenGigE** [ *instance* ]

例：

```
RP/0/RSP0/cpu 0: router# configure
RP/0/RSP0/cpu 0: router# interface TenGigE 0/0/0/1
```

10 ギガビット イーサネット インターフェイスのインターフェイス コンフィギュレーション モードを開始します。

**ステップ 2** **l2transport**

例：

```
RP/0/RSP0/cpu 0: router(config-if)# l2transport
```

ポートでレイヤ 2 トランスポート モードをイネーブルにし、レイヤ 2 トランスポート コンフィギュレーション モードを開始します。

**ステップ 3** **mtu** *bytes*

例：

```
RP/0/RSP0/cpu 0: router(config-if-l2)# mtu 1448
```

ブリッジドメインの最大パケットサイズまたは最大伝送ユニット（MTU）サイズを調整します。

- バイト単位で MTU サイズを指定するには、bytes 引数を使用します。範囲は 64 ～ 65535 です。

**ステップ 4** **no shutdown**

例：

```
RP/0/RSP0/cpu 0: router(config-if-l2)#  no shutdown
```

shutdown設定を削除します。こうすることでインターフェイスが強制的に管理上のダウン状態になります。

**ステップ 5** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## ギガビット イーサネット インターフェイスの設定

基本的なギガビットイーサネットまたは10ギガビットイーサネットインターフェイスを設定するには、次の作業を行います。

### 手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ipv4 address** *ip-address mask*
4. **flow-control** { **bidirectional** | **egress** | **ingress** }
5. **mtu** *bytes*
6. **mac-address** *value1.value2.value3*
7. **negotiation auto** （ギガビット イーサネット インターフェイスでのみ）
8. **no shutdown**
9. **commit** コマンドまたは **end** コマンドを使用します。
10. **show interfaces** [ **GigabitEthernet** | **TenGigE** ] *instance*

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure terminal
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface** *type interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config)# interface GigabitEthernet 0/1/0/0
```

インターフェイス コンフィギュレーション モードを開始し、イーサネット インターフェイス名と *rack/slot/module/port* 表記を指定します。

**ステップ 3** **ipv4 address** *ip-address mask*

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
```

IP アドレスとサブネット マスクをインターフェイスに割り当てます。

- *ip-address* をインターフェイスのプライマリ IPv4 アドレスに置き換えます。
- *mask* を関連付けられた IP サブネットのマスクに置き換えます。ネットワーク マスクは、次のいずれかの方法で指定できます。
  - 4 分割ドット付き 10 進表記のアドレスでネットワーク マスクを指定します。たとえば、255.0.0.0 は、値が 1 の各ビットは、対応するアドレスのビットがそのネットワーク アドレスに属することを示します。
  - ネットワーク マスクは、スラッシュ（/）と数字で示すことができます。たとえば、/8 は、マスクの最初の 8 ビットが 1 で、対応するアドレスのビットがネットワーク アドレスであることを示します。

**ステップ 4** **flow-control** { **bidirectional** | **egress** | **ingress** }

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# flow control ingress
```

（任意）フロー制御のポーズ フレームの送信および処理をイネーブルにします。

- **egress**：出力でフロー制御の一時停止フレームの送信を有効にします。
- **ingress**：入力で受信した一時停止フレームの処理を有効にします。
- **bidirectional**：出力でフロー制御の一時停止フレームの送信を有効にし、入力で受信した一時停止フレームの処理を有効にします。

**ステップ 5** **mtu** *bytes*

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# mtu 1448
```

（任意）インターフェイスの MTU 値を設定します。

- 通常フレームのデフォルトは 1514 バイト、802.1Q タグ付きフレームのデフォルトは 1518 バイトです。
- ギガビットイーサネットおよび 10 ギガビットイーサネットの mtu 値の範囲は 64 ～ 65535 バイトです。

**ステップ 6** **mac-address** *value1.value2.value3*

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# mac address 0001.2468.ABCD
```

（任意）[Management Ethernet] インターフェイスの MAC 層アドレスを設定します。

- 値は、それぞれMAC アドレスの上位、中間、および下位の 2 バイト（16 進）です。各 2 バイト値の範囲は 0 ～ ffff です。

**ステップ 7** **negotiation auto** （ギガビット イーサネット インターフェイスでのみ）

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# negotiation auto
```

（任意）ギガビットイーサネットインターフェイスのオートネゴシエーションをイネーブルにします。

- オートネゴシエーションは接続の両エンドで明示的にイネーブルにするか、接続の両エンドで速度とデュプレックス設定を手動設定する必要があります。
- オートネゴシエーションがイネーブルの場合、手動で設定した速度またはデュプレックス モードの設定の方が優先されます。

（注） **negotiation auto** コマンドは、ギガビットイーサネットインターフェイスだけで使用できます。

**ステップ 8** **no shutdown**

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# no shutdown
```

shutdown 設定を削除します。こうすることでインターフェイスが強制的に管理上のダウン状態になります。

**ステップ 9** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

**ステップ 10** **show interfaces** [ **GigabitEthernet** | **TenGigE** ] *instance*

**例**：

```
RP/0/RSP0/cpu 0: router #show interfaces TenGigE 0/3/0/0
```

（任意）ルータ上のインターフェイスに関する統計情報を表示します。

## 次の作業

- イーサネット インターフェイスで 802.1Q VLAN サブインターフェイスを設定する方法については、このマニュアルで後述する「キャリアイーサネットモデル」の章を参照してください。
- L2VPN 実装のイーサネットポートで AC を設定する方法については、この章で後述する「イーサネットポートでの接続回路の設定」を参照してください。

# イーサネットポートでの接続回路の設定

ギガビット イーサネットまたは 10 ギガビット イーサネット ポートで接続回路を設定するには、次の手順を実行します。接続回線設定の詳細については、『*Interface and Hardware Component Configuration Guide for Cisco ASR 9000 Series Routers*』を参照してください。

（注） この手順の各操作では、EFP モードで操作する L2VPN イーサネット ポートを設定します。

### 手順の概要

1. **configure**
2. **interface [GigabitEthernet | TenGigE]** *instance.subinterface* **l2transport**
3. **encapsulation dot1q** *vlan-id*
4. **interface [GigabitEthernet | TenGigE]** *instance.subinterface* **l2transport**
5. **encapsulation dot1q** *vlan-id*
6. **l2vpn**
7. **bridge group** *bridge-group-name*
8. **bridge-domain** *domain-name*
9. **interface [GigabitEthernet | TenGigE]** *instance.subinterface*
10. **interface [GigabitEthernet | TenGigE]** *instance.subinterface*
11. **commit** コマンドまたは **end** コマンドを使用します。
12. **show run interface [GigabitEthernet | TenGigE]** *instance.subinterface*

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface [GigabitEthernet | TenGigE]** *instance.subinterface* **l2transport**

**例**：

```
RP/0/RSP0/cpu 0: router(config)# interface GigabitEthernet0/5/0/0.20
l2transport
```

サブインターフェイス コンフィギュレーション モードを開始し、インターフェイス タイプ、ロケーション、サブインターフェイス番号を指定します。

- instance 引数を次のインスタンスのいずれかに置換します。
  - 物理イーサネット インターフェイス インスタンスまたはイーサネット バンドル インスタンス。名前表記は rack/slot/module/port です。値の間に表記の一部としてスラッシュが必要です。
  - イーサネット バンドル インスタンス。範囲は 1 ～ 65535 です。
- subinterface 引数をサブインターフェイスの値に置き換えます。範囲は 0 ～ 4095 です。
- 名前の表記は instance.subinterface の形式で、表記の一部として引数をピリオドで区切る必要があります。

**ステップ 3** **encapsulation dot1q** *vlan-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-subif)#encapsulation dot1q 50
```

一致する VLAN ID および EtherType をインターフェイスに割り当てます。

**ステップ 4** **interface [GigabitEthernet | TenGigE]** *instance.subinterface* **l2transport**

**例**：

```
RP/0/RSP0/cpu 0: router(config)#  interface GigabitEthernet0/5/0/0.20
l2transport
```

サブインターフェイス コンフィギュレーション モードを開始し、インターフェイス タイプ、ロケーション、サブインターフェイス番号を指定します。

- instance 引数を次のインスタンスのいずれかに置換します。
  - 物理イーサネット インターフェイス インスタンスまたはイーサネット バンドル インスタンス。名前表記は rack/slot/module/port です。値の間に表記の一部としてスラッシュが必要です。
  - イーサネット バンドル インスタンス。範囲は 1 ～ 65535 です。
- subinterface 引数をサブインターフェイスの値に置き換えます。範囲は 0 ～ 4095 です。
- 名前の表記は instance.subinterface の形式で、表記の一部として引数をピリオドで区切る必要があります。

**ステップ 5** **encapsulation dot1q** *vlan-id*

例：

```
RP/0/RSP0/cpu 0: router(config-subif)# encapsulation dot1q 50
```

一致する VLAN ID および EtherType をインターフェイスに割り当てます。

**ステップ 6** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config-subif)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 7** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group ce-doc-examples
```

名前付きブリッジグループのコンフィギュレーションモードを開始します。このコマンドは、新しいブリッジグループを作成するか、既存のブリッジグループを変更します（ブリッジグループが存在する場合）。ブリッジ グループは、ブリッジ ドメインを整理します。

**ステップ 8** **bridge-domain** *domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-
domain ac-example
```

名前付きブリッジドメインのコンフィギュレーションモードを開始します。このコマンドは、新しいブリッジドメインを作成するか、既存のブリッジドメインを変更します（ブリッジドメインが存在する場合）。

**ステップ 9** **interface [GigabitEthernet | TenGigE]** *instance.subinterface*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# interface GigabitEthernet0/5/0/0.20
```

ブリッジドメインにインターフェイスを追加し、パケットの転送と、同じブリッジドメイン内の他のインターフェイスからのパケットの受信を可能にします。これで、インターフェイス EFP は、このブリッジ ドメイン上の接続回線になります。

**ステップ 10** **interface [GigabitEthernet | TenGigE]** *instance.subinterface*

例：

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# interface GigabitEthernet0/5/0/1.15
```

ブリッジドメインにインターフェイスを追加し、パケットの転送と、同じブリッジドメイン内の他のインターフェイスからのパケットの受信を可能にします。これで、インターフェイス EFP は、このブリッジ ドメイン上の接続回線になります。

**ステップ 11** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

**ステップ 12** **show run interface [GigabitEthernet | TenGigE]** *instance.subinterface*

**例**：

```
RP/0/RSP0/cpu 0: router# show run interface GigabitEthernet0/5/0/1.15
```

（任意）ルータのサブインターフェイスの統計情報を表示します。

# EFP 出力フィルタリングの設定

ここでは、Cisco ASR 9000 シリーズルータで EFP 出力フィルタリング機能を設定する手順について説明します。

EFP 出力フィルタリングは L2 サブインターフェイス固有の機能で、出力方向でサブインターフェイス カプセル化フィルタリングがどのように実行されるかを厳密に制御します。EFP の動作とモデルに従い、サブインターフェイスから送信されるすべてのパケットは、同じパケットがサブインターフェイスで受信される場合には、サブインターフェイスのカプセル化または書き換えの条件に一致する必要があります（送信元 MAC アドレスと宛先 MAC アドレスは交換されます）。

EFP 出力フィルタリングには 2 つの段階があります。第 1 段階では rewrite コマンドは使用されず、第 2 段階では rewrite コマンドが使用されます。

第 1 段階のフィルタリングでは、パケットはカプセル化と照合され、一致するかどうか確認されます。これは、パケットをその EFP に転送するかどうか判別するために入力でパケットをチェックするのと同じ方法です。

第 2 段階のフィルタリングでは、出力の書き換え前の状態のパケットが正しいことを確認するために、出力の書き換えが実行される前にパケットがチェックされます。これは、出力パケットの VLAN カプセル化が、入力書き換え後の仮想の入力パケットと同一である必要があることを意味します。

書き換えと EFP 出力フィルタリングの両方がインターフェイスに設定されており、EFP 出力フィルタリングが原因で、出力トラフィックが予期せずにドロップされる場合、ユーザはドロップがどの段階で発生するか最初に確認する必要があります。

（注） 出力ドロップカウンタにより、そのインターフェイスの「show interface」表示で、出力 EFP フィルタリングが原因で発生したドロップが表示されます。出力ドロップカウンタは、複数の原因によるドロップの合計であり、EFP 出力フィルタリングが必ずしも原因ではありません。

**ethernet egress-filter** コマンドを使用することで、グローバルまたは L2 サブインターフェイスモードで出力 EFP フィルタリングを設定できます。

- **ethernet egress-filter strict** は、グローバル コンフィギュレーション モードで出力 EFP フィルタリングを設定します。
- **ethernet egress-filter {strict | disabled}** は、L2 サブインターフェイスモードで出力 EFP フィルタリングを設定します。

## 手順の概要

1. **configure**
2. **ethernet egress-filter strict**
3. **interface {GigabitEthernet | TenGigE | FastEthernet | Bundle-Ether}** *instance.subinterface*
4. **ethernet egress-filter {strict | disabled}**
5. **exit**

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/CPU0:PE44_ASR-9010# config Thu Jun 4 07:50:02.660 PST
RP/0/RSP0/CPU0:PE44_ASR-9010(config)#
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **ethernet egress-filter strict**

例：

```
RP/0/RSP0/CPU0:PE44_ASR-9010(config)#  ethernet egress-filter strict
```

デバイス上のすべてのサブインターフェイスに対して厳密な出力フィルタリングをデフォルトでイネーブルにします。

**ステップ 3** **interface {GigabitEthernet | TenGigE | FastEthernet | Bundle-Ether}** *instance.subinterface*

例：

```
RP/0/RSP0/CPU0:PE44_ASR-9010(config)#  interface GigabitEthernet 0/1/0/1.1
 RP/0/RSP0/CPU0:PE44_ASR-9010(config-subif)#
```

L2 サブインターフェイスを作成します。

**ステップ 4** **ethernet egress-filter {strict | disabled}**

例：

```
RP/0/RSP0/CPU0:PE44_ASR-9010(config-subif)# ethernet egress-filter strict
```

L2 サブインターフェイスに対し出力フィルタリングを明示的にイネーブルまたはディセーブルにすることができます。また、グローバル設定を上書きするために使用できます。

**ステップ 5** **exit**

例：

```
RP/0/RSP0/CPU0:PE44_ASR-9010(config-subif)# exit
RP/0/RSP0/CPU0:PE44_ASR-9010(config)# exit
```

コンフィギュレーション モードを終了します。

# 802.1Q VLAN インターフェイスの設定

## 802.1Q VLAN サブインターフェイスの設定

ここでは、802.1Q VLAN サブインターフェイスの設定手順について説明します。これらのサブインターフェイスを削除するには、この章の「802.1Q VLAN サブインターフェイスの削除」セクションを参照してください。

### 手順の概要

1. **configure**
2. **interface** {**GigabitEthernet** | **TenGigE** | **Bundle-Ether**} *instance.subinterface*
3. **l2transport**
4. **encapsulation dot1q** *vlan-id*
5. **commit** コマンドまたは **end** コマンドを使用します。
6. **show ethernet trunk bundle-ether** *instance*

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface** {**GigabitEthernet** | **TenGigE** | **Bundle-Ether**} *instance.subinterface*

例：

```
RP/0/RSP0/cpu 0: router(config)# interface TenGigE 0/2/0/4.10
```

サブインターフェイス コンフィギュレーション モードを開始し、インターフェイス タイプ、ロケーション、サブインターフェイス番号を指定します。

- *instance* 引数を次のインスタンスのいずれかに置換します。
  - 物理イーサネット インターフェイス インスタンスまたはイーサネット バンドル インスタンス。名前表記は *rack/slot/module/port* です。値の間に表記の一部としてスラッシュが必要です。
  - イーサネット バンドル インスタンス。範囲は 1 ～ 65535 です。
- *subinterface* 引数をサブインターフェイスの値に置き換えます。範囲は 0 ～ 4095 です。
- 名前の表記は *instance.subinterface* の形式で、表記の一部として引数をピリオドで区切る必要があります。

**ステップ 3** **l2transport**

例：

```
RP/0/RSP0/cpu 0: router(config-subif)# l2transport
```

ポートでレイヤ 2 トランスポート モードをイネーブルにし、レイヤ 2 トランスポート コンフィギュレーション モードを開始します。

**ステップ 4** **encapsulation dot1q** *vlan-id*

例：

```
RP/0/RSP0/cpu 0: router(config-subif-l2)# encapsulation dot1q 100
```

VLAN 接続回線をサブインターフェイスに割り当てます。

- vlan-id 引数にはサブインターフェイス ID を指定します。範囲は 1 ～ 4094 です（0 と 4095 は予約されています）。基本の Dot1Q 接続回線を設定するには、次の構文を使用します。

  **encapsulation dot1q vlan-id**

- QinQ 接続回線を設定するには、次の構文を使用します。

  **encapsulation dot1q vlan-id second-dot1q vlan-id**

（注）　次は、各種の **encapsulation** コマンドです。

- encapsulation dot1q 100
- encapsulation dot1q 100 second-dot1q 101
- encapsulation dot1ad 200 dot1q 201

**ステップ 5**　**commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

**ステップ 6**　**show ethernet trunk bundle-ether** *instance*

**例**：

```
RP/0/RSP0/cpu 0: router# show ethernet trunk bundle-ether 5
```

（任意）インターフェイス コンフィギュレーションを表示します。

イーサネット バンドル インスタンスの範囲は 1 ～ 65535 です。

## ネイティブ VLANの設定

ここでは、インターフェイスにネイティブ VLAN を設定する方法について説明します。

### 手順の概要

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE** | **Bundle-Ether**] *instance.subinterface* **l2transport**
3. **encapsulation** [ **dot1q** *vlan-id*, **untagged**]
4. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1**　**configure**

**例**：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モード を開始します。

**ステップ 2** **interface** [**GigabitEthernet** | **TenGigE** | **Bundle-Ether**] *instance.subinterface* **l2transport**

**例**：

```
RP/0/RSP0/cpu 0: router(config)# interface GigabitEthernet 0/2/0/4.2 l2transport
```

サブインターフェイス コンフィギュレーション モードを開始し、インターフェイス タイプ、ロケーション、サブインターフェイス番号を指定します。

- instance 引数を次のインスタンスのいずれかに置換します。
  - 物理イーサネット インターフェイス インスタンスまたはイーサネット バンドル インスタンス。名前表記は *rack/slot/module/port* で、値の間のスラッシュは表記の一部として必要です。
  - イーサネット バンドル インスタンス。範囲は 1 ～ 65535 です。
- *subinterface* 引数をサブインターフェイスの値に置き換えます。範囲は 0 ～ 4095 です。
- 名前の表記は *instance.subinterface* の形式で、表記の一部として引数をピリオドで区切る必要があります。

（注） コマンド文字列に l2transport キーワードを含める必要があります。そうしないと、接続回線ではなく、レイヤ 3 サブインターフェイスが作成されます。

**ステップ 3** **encapsulation** [ **dot1q** *vlan-id*, **untagged**]

**例**：

```
RP/0/RSP0/cpu 0: router(config-subif)# encapsulation dot1q 400
```

802.1Q トランク インターフェイスに関連付けられた、ネイティブの VLAN を定義します。

- *vlan-id* 引数は、サブインターフェイスの ID です。
- 範囲は 1 ～ 4094 です（0 と 4095 は予約されています）。

**untagged** キーワードを指定した **encapsulation** コマンドを実行することで、dot1q 400 とタグなしフレームの両方を受信できます。

**ステップ 4** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# 802.1Q VLAN サブインターフェイスの削除

ここでは、この章の「802.1Q VLAN サブインターフェイスの設定」で設定した 802.1Q VLAN サブインターフェイスを削除する方法について説明します。

## 手順の概要

1. **configure**
2. **no interface {GigabitEthernet | TenGigE | Bundle-Ether}** *instance.subinterface*
3. ステップ 2 を繰り返し、その他の VLAN サブインターフェイスを削除します。
4. **commit** コマンドまたは **end** コマンドを使用します。
5. **show ethernet trunk bundle-ether** *instance*

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **no interface {GigabitEthernet | TenGigE | Bundle-Ether}** *instance.subinterface*

例：

```
RP/0/RSP0/cpu 0: router(config)# no interface TenGigE 0/2/0/4.10
```

サブインターフェイスを削除すると、そのサブインターフェイスに適用されているすべての設定も自動的に削除されます。

- *instance* 引数を次のインスタンスのいずれかに置換します。
  - 物理イーサネット インターフェイス インスタンスまたはイーサネット バンドル インスタンス。名前表記は *rack/slot/module/port* で、値の間のスラッシュは表記の一部として必要です。
  - イーサネット バンドル インスタンス。範囲は 1 ～ 65535 です。
- *subinterface* 引数をサブインターフェイスの値に置き換えます。範囲は 0 ～ 4095 です。

名前の表記は *instance.subinterface* の形式で、表記の一部として引数をピリオドで区切る必要があります。

**ステップ 3** ステップ 2 を繰り返し、その他の VLAN サブインターフェイスを削除します。

**ステップ 4** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。

• [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

**ステップ 5** **show ethernet trunk bundle-ether** *instance*

例：

```
RP/0/RSP0/cpu 0: router# show ethernet trunk bundle-ether 5
```

（任意）インターフェイス コンフィギュレーションを表示します。

イーサネット バンドル インスタンスの範囲は 1 ～ 65535 です。

# 設定例

## イーサネット インターフェイスの設定：例

次に、10 ギガビット イーサネットのモジュラ サービス カードのインターフェイスを設定する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/0/0/1
RP/0/RSP0/CPU0:router(config-if)# l2transport
RP/0/RSP0/CPU0:router(config-if)# mtu 1448
RP/0/RSP0/CPU0:router(config-if)# no shutdown
RP/0/RSP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes

RP/0/RSP0/CPU0:router# show interfaces TenGigE 0/0/0/1

TenGigE0/0/0/1 is down, line protocol is down
  Hardware is TenGigE, address is 0001.2468.abcd (bia 0001.81a1.6b23)
  Internet address is 172.18.189.38/27
  MTU 1448 bytes, BW 10000000 Kbit
     reliability 0/255, txload Unknown, rxload Unknown
  Encapsulation ARPA,
  Full-duplex, 10000Mb/s, LR
  output flow control is on, input flow control is on
  loopback not set
  ARP type ARPA, ARP timeout 01:00:00
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

# L2VPN AC の設定:例

次に、イーサネット インターフェイスで L2VPN AC を設定する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/5/0/0.2 l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RSP0/CPU0:router(config-subif)# ethernet egress-filter strict
RP/0/RSP0/CPU0:router(config-subif)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# clear

RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/5/0/0.2 l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RSP0/CPU0:router(config-subif)# ethernet egress-filter strict
RP/0/RSP0/CPU0:router(config-subif)# interface gigabitethernet 0/5/0/1.100 l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RSP0/CPU0:router(config-subif)# ethernet egress-filter strict
RP/0/RSP0/CPU0:router(config-subif)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group example
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain mybridge
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface gigabitethernet 0/5/0/0.2
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# interface gigabitethernet 0/5/0/1.100
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# exit
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# exit
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# exit
RP/0/RSP0/CPU0:router(config-l2vpn)# exit
RP/0/RSP0/CPU0:router(config)# show

Building configuration...
!! IOS XR Configuration 0.0.0
interface GigabitEthernet0/5/0/0.2 l2transport
 encapsulation dot1q 100
 ethernet egress-filter strict
!
interface GigabitEthernet0/5/0/1.100 l2transport
 encapsulation dot1q 100
 ethernet egress-filter strict
!
l2vpn
 bridge group example
  bridge-domain mybridge
   interface GigabitEthernet0/5/0/0.2
   !
   interface GigabitEthernet0/5/0/1.100
   !
  !
 !
end
```

# VPWS へのリンクバンドルの設定:例

## 物理インターフェイス（ポートモード）

```
interface Bundle-Ether12
 l2transport
!
interface GigabitEthernet0/1/0/10
 negotiation auto
 l2transport
```

```
!
interface GigabitEthernet0/1/0/20
 bundle id 12 mode on
 negotiation auto
!
interface GigabitEthernet0/1/0/21
 bundle id 12 mode on
 negotiation auto
!
 !
l2vpn
 xconnect group test
  p2p test
   interface Bundle-Ether12
   !
   interface GigabitEthernet0/1/0/10
   !
  !
 !
!
```

## サブインターフェイス（EFP モード）

```
interface Bundle-Ether12
!
interface Bundle-Ether12.1 l2transport
  encapsulation dot1q 12
 !
!
interface GigabitEthernet0/1/0/10
 negotiation auto
!
interface GigabitEthernet0/1/0/10.1  l2transport
  encapsulation dot1q 12
 !

!
interface GigabitEthernet0/1/0/20
 bundle id 12 mode on
 negotiation auto
!
interface GigabitEthernet0/1/0/21
 bundle id 12 mode on
 negotiation auto
!
 !
l2vpn
xconnect group test
  p2p test
   interface Bundle-Ether12.1
   !
   interface GigabitEthernet0/1/0/10.1
   !
  !
 !
!
```

# イーサネットバンドルへの L2 および L3 サービスの設定:例

次に、イーサネット バンドル インターフェイスに L3 サービスを設定する例を示します。

```
configure
interface Bundle-Ether 100
 ipv4 address 12.12.12.2 255.255.255.0

!
```

次に、イーサネット バンドル サブインターフェイスに L3 サービスを設定する例を示します。

```
configure
interface Bundle-Ether 100.1
 ipv4 address 13.13.13.2 255.255.255.0

!
```

次に、イーサネット バンドル インターフェイスに L2 サービスを設定する例を示します。

```
configure
 interface Bundle-Ether 101
 l2transport

!
```

次に、イーサネット バンドル インターフェイスに L2 サービスを設定する例を示します。

```
configure
 interface Bundle-Ether1.1 l2transport
!
```

# VLAN サブインターフェイスの設定：例

次に、VLAN サブインターフェイスを作成する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/2/0/4.1 l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 20
RP/0/RSP0/CPU0:router(config-subif)# interface TenGigE0/2/0/4.2 l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 30
RP/0/RSP0/CPU0:router(config-subif)# interface TenGigE0/2/0/4.3 l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 40
RP/0/RSP0/CPU0:router(config-subif)# commit
RP/0/RSP0/CPU0:router(config-subif)# exit
RP/0/RSP0/CPU0:router(config)# exit
```

次に、イーサネットバンドルに 2 つの VLAN サブインターフェイスを一度に作成する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether 1 l2transport
RP/0/RSP0/CPU0:router(config-if-l2)# exit
RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether  1.1 l2transport
RP/0/RSP0/CPU0:router(config-subif-l2)#  encapsulation dot1q 10
RP/0/RSP0/CPU0:router(config-subif)# exit
RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether 1.2 l2transport
RP/0/RSP0/CPU0:router(config-subif-l2)# encapsulation dot1q 20
RP/0/RSP0/CPU0:router(config-subif)# exit
```

次に、基本の Dot1Q 接続回線を作成する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/2/0/4.1 l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 20
```

```
RP/0/RSP0/CPU0:router(config-subif)# commit
RP/0/RSP0/CPU0:router(config-subif)# exit
RP/0/RSP0/CPU0:router(config)# exit
```

次に、QinQ 接続回線を作成する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/2/0/4.2 l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 20 second-dot1q 10
RP/0/RSP0/CPU0:router(config-subif)# commit
RP/0/RSP0/CPU0:router(config-subif)# exit
RP/0/RSP0/CPU0:router(config)# exit
```

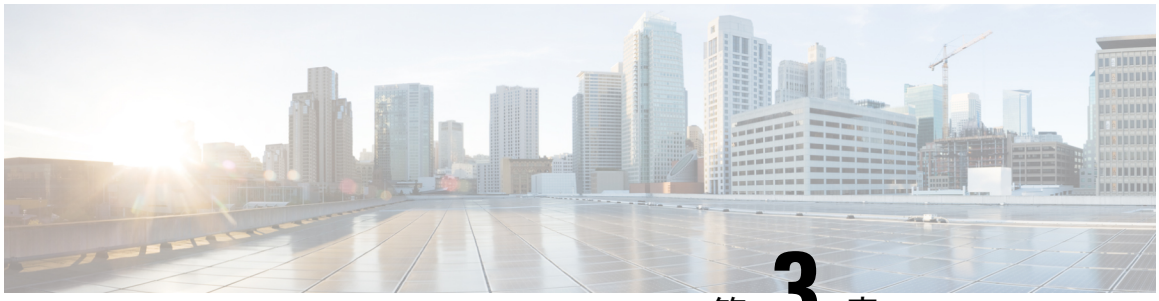
次に、Q-in-Any 接続回線を作成する例を示します。

```
RP/0/RSP/CPU0:router# configure
RP/0/RSP/CPU0:router(config)# interface TenGigE 0/2/0/4.3 l2transport
RP/0/RSP/CPU0:router(config-subif)# encapsulation dot1q 30 second-dot1q any
RP/0/RSP/CPU0:router(config-subif)# commit
RP/0/RSP/CPU0:router(config-subif)# exit
RP/0/RSP/CPU0:router(config)# exit
```

# 次の作業

イーサネット インターフェイスの設定が完了したら、イーサネット インターフェイスで各 VLAN サブインターフェイスを設定できます。VLAN サブインターフェイス設定の詳細については、このマニュアルで後述する「キャリアイーサネットモデル」の章を参照してください。

IPv6 の詳細については、『*IP Addresses and Services Configuration Guide for Cisco ASR 9000 Series Routers*』を参照してください。



第 3 章

# イーサネット機能

この章では、Cisco IOS XR ソフトウェアをサポートする Cisco ASR 9000 シリーズ アグリゲーション サービス ルータのレイヤ 2（L2）イーサネット機能を設定する方法について説明します。

イーサネットインターフェイスの設定の詳細については、この設定ガイドの「キャリアイーサネットモデル」モジュールを参照してください。

**Cisco ASR 9000 シリーズ ルータのイーサネット インターフェイス設定の機能履歴**

| リリース | 変更内容 |
|---|---|
| リリース 3.9.1 | ポリシーベースの転送およびレイヤ 2 プロトコルトンネリング機能のサポートが追加されました。 |



## イーサネット機能を実装するための前提条件

適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。

ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

# イーサネットの機能の実装に関する情報

10 ギガビット イーサネット インターフェイスを設定するには、次の概念を理解しておく必要があります。

## ポリシー ベースの転送

Cisco ASR 9000 シリーズ ルータでは、単一の MAC アドレスを、ポートの設定済みの VLAN とは異なる VLAN にマップできます。2 つの異なる EFP に入るトラフィックを分離するためには、送信元 VLAN タグおよび送信元 MAC アドレスを使用して EFP を定義する必要があります。

（注） この機能は、ASR 9000 イーサネットラインカードでのみサポートされています。

## レイヤ 2 プロトコル トンネリング

レイヤ2プロトコル トンネリング（L2PT）は、レイヤ2（L2）スイッチング ドメイン間でイーサネット プロトコル フレームをトンネリングするための、シスコ独自のプロトコルです。

L2プロトコルフレームがL2スイッチングデバイスのインターフェイスに着信すると、スイッチまたはルータはフレームで次のいずれかのアクションを実行します。

- 転送：フレームは例外的な処理なしでスイッチングまたはルーティングされます。
- ドロップ：フレームはルータで廃棄されます。
- 終端：ルータは、フレームが L2 プロトコル フレームであると認識し、プロトコル処理のためにこれをルータのコントロール プレーンに送信します。
- トンネリング：ルータは、フレームをカプセル化して、プロトコルフレームとしてのアイデンティティを非表示にします。これにより、フレームが別のルータで終端することを防ぎます。トンネルの反対側ではカプセル化を解除して、フレームを元の状態に戻します。

### L2PT の機能

Cisco ASR 9000 シリーズ ルータは、次の機能を備えています。

- 次のプロトコルをトンネリングします。
  - Cisco Discovery Protocol（CDP）
  - スパニングツリー プロトコル（STP およびそのバリエーション）
  - 仮想トランキング プロトコル（VTP）

- 次のトンネリング モードをサポートします。
  - 転送
  - 反転
- L2PT は VLAN ヘッダーを持つプロトコル フレームをカプセル化し、カプセル化を解除します。
- 巨大フレーム レートの処理機能をサポートします。Cisco ASR 9000 シリーズ ルータは、インターフェイス ライン レートで L2PT カプセル化とカプセル化解除を実行します。

（注） 専用の L2PT カウンタはありません。QoS またはその他のパラメータの L2PT 特定の調整はありません。

## 転送モードの L2PT

次の図に、転送モードで設定された L2PT を示します。

**図 *4 :* 転送モードの *L2PT***

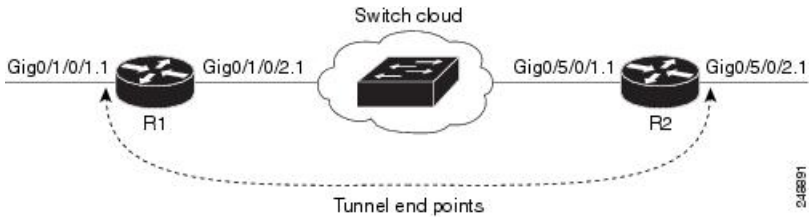



図 1 では、サービス プロバイダー ネットワーク（S ネットワーク）について説明します。カスタマー ネットワーク（C ネットワーク）は、GigabitEthernet サブインターフェイス 0/1/0/1.1 でルータ R1 に接続し、GigabitEthernet サブインターフェイス 0/5/0/2.1 でルータ R2 に接続します。C ネットワークは図に示されていません。ただし、C ネットワークは、S ネットワーク経由で L2 トラフィックを送信し、S ネットワークはエンド ツー エンドでトラフィックを切り替えます。カスタマー トラフィックは、L2 プロトコル フレームを伝送します。L2PT の目的は、これらのプロトコル フレームが S ネットワークを通過できるようにすることです。転送モードでは、L2PT は、S ネットワークのカスタマー側インターフェイスである R1 GigabitEthernet 0/1/0/1.1 と R2 GigabitEthernet 0/5/0/2.1 に適用されます。

上の図は、転送モードの L2PT の設定を示しています。

R1：

```
!
interface GigabitEthernet0/1/0/1
 negotiation auto
!
interface GigabitEthernet0/1/0/1.1 l2transport
 encapsulation default
 l2protocol cpsv tunnel
!
```

```
interface GigabitEthernet0/1/0/2
 negotiation auto
!
interface GigabitEthernet0/1/0/2.1 l2transport
 encapsulation default
!
l2vpn
 xconnect group examples
  p2p r1-connect
   interface GigabitEthernet0/1/0/1.1
   interface GigabitEthernet0/1/0/2.1
  !
 !
!
```

R2：

```
!
interface GigabitEthernet0/5/0/1
 negotiation auto
!
interface GigabitEthernet0/5/0/1.1 l2transport
 encapsulation default
!
interface GigabitEthernet0/5/0/2
 negotiation auto
!
interface GigabitEthernet0/5/0/2.1 l2transport
 encapsulation default
 l2protocol cpsv tunnel
!
l2vpn
 xconnect group examples
  p2p r2-connect
   interface GigabitEthernet0/5/0/1.1
   interface GigabitEthernet0/5/0/2.1
  !
 !
!
```

プロトコル トラフィックは、GigabitEthernet サブインターフェイス 0/1/0/1.1 でルータ R1 に入ります。ルータ R1 はプロトコル フレームとしてフレームを検出して、カスタマー側インターフェイスで L2PT カプセル化を実行します。R1 内では、ローカル接続 *r1-connect* は、R1 の顧客側インターフェイスとサービスプロバイダー側インターフェイスを接続します。トラフィックは、他の複数のサービス プロバイダー ネットワークのルータまたはスイッチ（スイッチ クラウド）を介して GigabitEthernet サブインターフェイス 0/1/0/2.1 のルータ R1 から GigabitEthernet サブインターフェイス 0/5/0/1.1 のルータ R2 に通過します。ルータ R2 は、ローカル接続 *r2-connect* を介して顧客側インターフェイスとサービスプロバイダー側インターフェイスを接続します。したがって、トラフィックは、カスタマー側インターフェイスの GigabitEthernet 0/5/0/2.1 に送信されます。このインターフェイスで、L2PT のカプセル化が解除され、プロトコル トラフィックはルータ R2 からカスタマー ネットワークに流れます。

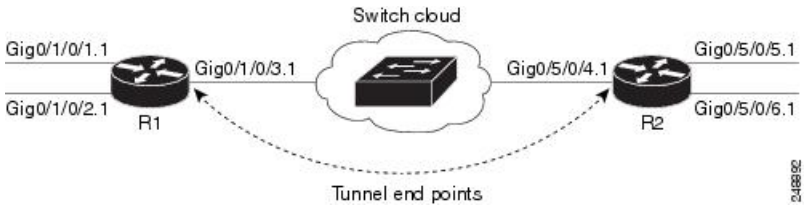
L2PT が設定されていない場合、R1 に送信されるカスタマー プロトコル フレームは終端します。カスタマー トラフィックは、さまざまなトラフィックで構成できます。プロトコル フレームは、全体的なトラフィック ストリームのうちわずかな割合で構成されます。

# プロトコル フレーム タギングを使用した反転モードの L2PT

Cisco ASR 9000 シリーズ ルータは、VLAN ヘッダーを持つサポートされている L2 プロトコル フレームで L2PT カプセル化およびカプセル化解除を実行できます。L2 プロトコル フレームに VLAN ヘッダーは含まれません。ただし、カスタマー キャンパス間でカスタマー プロトコル トラフィックを転送するサービス プロバイダー（SP）ネットワークでは、この機能を配置して、SP ネットワーク内で使用できます。

次の図に、反転モードで設定された L2PT を示します。R1 に入るカスタマー トラフィックはトランキングされており、すべてのトラフィックがタグ付きであると想定します。唯一のタグなしトラフィックは、カスタマーネットワークから発信されるプロトコル トラフィックです。

**図 *5 :* 反転モードの *L2PT***




反転モードで L2PT が設定されている場合、L2PT カプセル化は、フレームがインターフェイスを出ると行われます。同様に、反転モードのカプセル化解除は、フレームがインターフェイスに入ったときに実行されます。したがって、L2PT トンネルは、カスタマー側インターフェイスではなく、サービス プロバイダー側インターフェイス間で形成されます。

この例では、プロトコル トラフィックがルータ R1 に入ると、VLAN タグが追加されます。トラフィックがサービス プロバイダー ネットワークを通じて送信される前に、2 番目の VLAN タグが追加されます（100）。Cisco ASR 9000 シリーズ ルータは、二重タグ付きプロトコルフレームで L2PT カプセル化を実行します。

上の図に、4 つの顧客側インターフェイス（R1：GigabitEthernet サブインターフェイス 0/1/0.1.1、GigabitEthernet サブインターフェイス 0/1/0/2.1 および R2：GigabitEthernet サブインターフェイス 0/5/0/5.1、GigabitEthernet サブインターフェイス 0/5/0/6.1）、および 2 つのサービスプロバイダー側インターフェイス（R1：GigabitEthernet サブインターフェイス 0/1/0/3.1 と R2：GigabitEthernet サブインターフェイス 0/5/0/4.1）を示します。

上の図は、反転モードの L2PT の設定を示しています。

R1：

```
!
interface GigabitEthernet0/1/0/1
 negotiation auto
!
interface GigabitEthernet0/1/0/1.1 l2transport
 encapsulation untagged
 rewrite ingress tag push dot1q 100 symmetric
 ethernet egress-filter strict
!
interface GigabitEthernet0/1/0/2
 negotiation auto
!
interface GigabitEthernet0/1/0/2.1 l2transport
 encapsulation untagged
```

```
 rewrite ingress tag push dot1q 200 symmetric
 ethernet egress-filter strict
!
interface GigabitEthernet0/1/0/3
 negotiation auto
!
interface GigabitEthernet0/1/0/3.1 l2transport
 encapsulation dot1q 500
 rewrite ingress tag pop 1 symmetric
 l2protocol cpsv reverse-tunnel
 ethernet egress-filter strict
!
l2vpn
 bridge group examples
  bridge-domain r1-bridge
   interface GigabitEthernet0/1/0/1.1
   !
   interface GigabitEthernet0/1/0/2.1
!
   interface GigabitEthernet0/1/0/3.1
   !
  !
 !
!
```

R2：

```
!
interface GigabitEthernet0/5/0/4
 negotiation auto
!
interface GigabitEthernet0/5/0/4.1 l2transport
 encapsulation dot1q 500
 rewrite ingress tag pop 1 symmetric
 l2protocol cpsv reverse-tunnel
 ethernet egress-filter strict
!
interface GigabitEthernet0/5/0/5
 negotiation auto
!
interface GigabitEthernet0/5/0/5.1 l2transport
 encapsulation untagged
 rewrite ingress tag push dot1q 100 symmetric
 ethernet egress-filter strict
!
interface GigabitEthernet0/5/0/6
 negotiation auto
!
interface GigabitEthernet0/5/0/6.1 l2transport
 encapsulation untagged
 rewrite ingress tag push dot1q 200 symmetric
 ethernet egress-filter strict
!
l2vpn
 bridge group examples
  bridge-domain r2-bridge
   interface GigabitEthernet0/5/0/4.1
   !
   interface GigabitEthernet0/5/0/5.1
   !
   interface GigabitEthernet0/5/0/6.1
   !
  !
 !
```

!

次のことが前提となっています。

- ルータ R1 に入るカスタマー トラフィックはトランキングされます。つまり、すべてのトラフィックがタグ付けされています。唯一のタグなしトラフィックは、カスタマーネットワークから到着するプロトコル トラフィックです。
- ルータ R1 の GigabitEthernet 0/1/0/1 とルータ R2 の GigabitEthernet 0/5/0/5 のカスタマー側インターフェイスは、同じカスタマーに属しています。ルータ R1 の GigabitEthernet 0/1/0/2 とルータ R2 の GigabitEthernet 0/5/0/6 のカスタマー側インターフェイスは、別のカスタマーに属しています。
- 異なるカスタマーからのトラフィックは分離されたままになります。
- L2 プロトコル トラフィックだけがカスタマー側インターフェイスを経由して送信されます。
- カスタマー側インターフェイスに入る L2 プロトコル トラフィックはタグなしです。
- トラフィックは、スイッチ クラウドを正常にパススルーするには、L2PT カプセル化されている必要があります。

このトポロジの目的は、ルータ R1 と R2 が複数のカスタマーインターフェイスからカスタマープロトコル トラフィックを受信する必要があり、単一のサービス プロバイダー インターフェイスとリンク間でトラフィックを多重化する必要があることです。カプセル化解除の最後に、反転が実行されます。GigabitEthernet サブインターフェイス 0/1/0/2.1 のルータ R1 に入るトラフィックは、GigabitEthernet サブインターフェイス 0/5/0/6.1 だけからルータ R2 を出るのに対して、GigabitEthernet サブインターフェイス 0/1/0/1.1 のルータ R1 に入るトラフィックは、GigabitEthernet サブインターフェイス 0/5/0/5.1 だけからルータ R2 を出ます。

GigabitEthernet インターフェイス 0/1/0/1 のルータ R1 に入るプロトコル フレームは、この方法でネットワークを通過します。

- プロトコル フレームは、フレームがタグなしであるため、GigabitEthernet サブインターフェイス 0/1/0/1.1 に送信されます。
- GigabitEthernet サブインターフェイス 0/1/0/1.1 で rewrite ステートメントを使用すると、ID 100 のタグがフレームに追加されます。
- フレームは、ルータ R1 のブリッジドメイン r1-bridge に入ります。
- ブリッジ（r1-bridge）は、発信元 AC（スプリット ホライズン AC）を除き、ブリッジ ドメイン上のすべての接続回線（AC）にフレームをフラッディングします。
- GigabitEthernet サブインターフェイス 0/1/0/2.1 でのイーサネット出力フィルタリングは、タグ ID のミスマッチを検出し、フレームをドロップします。このように、ブリッジドメインのフラッディングされたトラフィックは、他の顧客のインターフェイスを出ることができません。
- フレームのフラッディングされたコピーは GigabitEthernet サブインターフェイス 0/1/0/3.1 に送信されます。

- GigabitEthernet サブインターフェイス 0/1/0/3.1 は 2 番目のタグを追加します。
- フレームは、GigabitEthernet インターフェイス 0/1/0/3 を介してルータ R1 を出る前に GigabitEthernet サブインターフェイス 0/1/0/3.1 によって L2PT カプセル化を受信します。

（注） 現在フレームには二重のタグが付いており（内部が 100、外部が 500）になっており、L2PT MAC DA があります。

- フレームは、L2PT カプセル化が原因で、ルータ R2 GigabitEthernet インターフェイス 0/5/0/4 に渡されます。
- フレームは、GigabitEthernet インターフェイス 0/5/0/4 のルータ R2 に入った後、GigabitEthernet サブインターフェイス 0/5/0/4.1 に送信されます。
- GigabitEthernet サブインターフェイス 0/5/0/4.1 に入るときに、L2PT カプセル解除動作がフレームで実行されます。
- 外部タグ ID 500 は、GigabitEthernet サブインターフェイス 0/5/0/4.1 によって削除されます。
- ルータ R2 のブリッジ（r2-bridge）は、すべての AC にフレームをフラッディングします。
- イーサネット出力フィルタリングは、フレームが出る AC を除くすべての AC でフレームをドロップします。
- フレームが GigabitEthernet サブインターフェイス 0/5/0/5.1 のルータ R2 を出るため、ID 100 のタグが削除されます。
- GigabitEthernet インターフェイス 0/5/0/5 のルータ R2 から出るフレームは、GigabitEthernet インターフェイス 0/1/0/1 を介してルータ R1 に入った元のフレームと同じです。

## L2PT 設定メモ

L2PT を設定する際は、次の点に注意してください。

- **l2protocol** コマンドは、メインまたは L2 のいずれかのサブインターフェイスで設定できます。
- **l2protocol** コマンドは、物理またはバンドルインターフェイスで設定できます。
- **l2protocol** および **ethernet filtering** コマンドが同じインターフェイスで設定されている場合、L2PT カプセル化はイーサネットフィルタリングの前に発生します。これは、L2PT によって、CDP、STP、および VTP プロトコル フレームがイーサネット フィルタリングによってドロップされないようにすることを意味します。
- L2PT が他のインターフェイス機能で設定されている場合、L2PT カプセル化は、他のインターフェイス機能の処理の前に発生します。

- L2PT カプセル化およびカプセル化解除は、タグなしプロトコル フレーム、一重タグ フレーム、および二重タグ付きフレームでサポートされます。タグ Ethertype 0x8100、0x88A8、および 0x9100 はサポートされていますが、0x9200 はサポートされていません。

# イーサネット機能の実装方法

（注） イーサネット インターフェイスの設定については、『Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Configuration Guide』を参照してください。

# ポリシーベースの転送の設定

## ポリシーベースの転送のイネーブル化

ポリシーベースの転送をイネーブルにするには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **interface** *type interface-path-id.subinterface* **l2transport**
3. 次のいずれかを実行します。
   - **encapsulation dot1q** *vlan-id* **ingress source-mac** *mac-address* または
   - **encapsulation dot1ad** *vlan-id* **ingress source-mac** *mac-address* または
   - **encapsulation untagged ingress source-mac** *mac-address* または
   - **encapsulation dot1q** *vlan-id* **second-dot1q** *vlan-id* **ingress source-mac** *mac-address*
4. 次のいずれかを実行します。
   - **rewrite ingress tag translate 1-to-1 dot1q** *vlan-id* **symmetric** または
   - **rewrite tag push dot1q***vlan-id* **symmetric**
5. **ethernet egress-filter strict**
6. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1 configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface** *type interface-path-id.subinterface* **l2transport**

例：

```
RP/0/RSP0/cpu 0: router# interface GigabitEthernet 0/2/0/4.10 l2transport
```

サブインターフェイス コンフィギュレーション モードを開始し、ポートでレイヤ 2 トランスポート モードをイネーブルにし、レイヤ 2 トランスポート コンフィギュレーション モードを開始します。

**ステップ 3** 次のいずれかを実行します。

- **encapsulation dot1q** *vlan-id* **ingress source-mac** *mac-address* または
- **encapsulation dot1ad** *vlan-id* **ingress source-mac** *mac-address* または
- **encapsulation untagged ingress source-mac** *mac-address* または
- **encapsulation dot1q** *vlan-id* **second-dot1q** *vlan-id* **ingress source-mac** *mac-address*

例：

```
RP/0/RSP0/cpu 0: router(config-subif)#
encapsulation dot1q 10 ingress source-mac 0.1.2
or
RP/0/RSP0/cpu 0: router(config-subif)#
encapsulation dot1ad 10 ingress source-mac 0.1.4
or
RP/0/RSP0/cpu 0: router(config-subif)#
encapsulation untagged ingress source-mac 0.1.3
or
RP/0/RSP0/cpu 0: router(config-subif)#
encapsulation dot1ad 10 dot1q 10 ingress source-mac 0.1.2
or
RP/0/RSP0/cpu 0: router(config-subif)#
encapsulation dot1q 10 second-dot1q 20 ingress source-mac 0.1.2
```

一致する VLAN ID および EtherType をインターフェイスに割り当てます。

**ステップ 4** 次のいずれかを実行します。

- **rewrite ingress tag translate 1-to-1 dot1q** *vlan-id* **symmetric** または
- **rewrite tag push dot1q***vlan-id* **symmetric**

例：

```
RP/0/RSP0/cpu 0: router(config-subif)# rewrite ingress tag translate 1-to-1 dot1q 100 symmetric
or
rewrite ingress tag push dot1q 101 symmetric
```

サービス インスタンスへのフレーム入力で実行されるカプセル化調整を指定します。

**ステップ 5** **ethernet egress-filter strict**

例：

```
RP/0/RSP0/cpu 0: router(config-subif)# ethernet egress-filter strict
```

すべてのサブインターフェイスで厳密な出力フィルタリングをイネーブルにします。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## 送信元バイパス フィルタの設定

送信元バイパス フィルタを追加するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **interface** *type interface-path-id.subinterface* **l2transport**
3. 次のいずれかを実行します。
   - **encapsulation dot1q** *vlan-id* または
   - **encapsulation dot1ad** *vlan-id* または
   - **encapsulation untagged** または
   - **encapsulation dot1ad** *vlan-id* **dot1q** *vlan-id* または
   - **encapsulation dot1q** *vlan-id* **second-dot1q** *vlan-id* または
4. **rewrite ingress tag translate translate 1-to-1 dot1q***vlan-id* **symmetric**
5. **ethernet egress-filter disable**
6. **ethernet source bypass egress-filter**
7. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

**例**：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モード を開始します。

**ステップ 2** **interface** *type interface-path-id.subinterface* **l2transport**

**例**：

```
RP/0/RSP0/cpu 0: router(config)# interface GigabitEthernet 0/2/0/4.1 l2transport
```

サブインターフェイス コンフィギュレーション モードを開始し、ポートでレイヤ 2 トランスポート モードをイネーブルにし、レイヤ 2 トランスポート コンフィギュレーション モードを開始します。

**ステップ 3** 次のいずれかを実行します。

- **encapsulation dot1q** *vlan-id* または
- **encapsulation dot1ad** *vlan-id* または
- **encapsulation untagged** または
- **encapsulation dot1ad** *vlan-id* **dot1q** *vlan-id* または
- **encapsulation dot1q** *vlan-id* **second-dot1q** *vlan-id* または

**例**：

```
RP/0/RSP0/cpu 0: router(config-subif)#
encapsulation dot1q 10
or
RP/0/RSP0/cpu 0: router(config-subif)#
encapsulation dot1ad 10
or
RP/0/RSP0/cpu 0: router(config-subif)#
encapsulation untagged
or
RP/0/RSP0/cpu 0: router(config-subif)#
encapsulation dot1ad 10 dot1q 10
or
RP/0/RSP0/cpu 0: router(config-subif)#
encapsulation dot1q 10 second-dot1q 20
```

一致する VLAN ID および EtherType をインターフェイスに割り当てます。

**ステップ 4** **rewrite ingress tag translate translate 1-to-1 dot1q***vlan-id* **symmetric**

**例**：

```
RP/0/RSP0/cpu 0: router
(config-subif)# rewrite ingress tag translate 1-to-1 dot1q 100 symmetric
```

サービス インスタンスへのフレーム入力で実行されるカプセル化調整を指定します。

**ステップ 5** **ethernet egress-filter disable**

**例**：

```
RP/0/RSP0/cpu 0: router(config-subif)# ethernet egress-filter strict
```

すべてのサブインターフェイスで出力フィルタリングをディセーブルにします。

**ステップ 6** **ethernet source bypass egress-filter**

**例**：

```
RP/0/RSP0/cpu 0: router(config-subif)# ethernet source bypass egress-filter
```

サブインターフェイスで送信元バイパス出力フィルタリングをイネーブルにします。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# 設定例

## ポリシーベースの転送の設定：例

次に、ポリシーベースの転送を設定する例を示します。

```
config
 interface GigabitEthernet0/0/0/2.3 l2transport
 encapsulation dot1q 10 ingress source-mac 0000.1111.2222
 rewrite ingress tag translate 1-to-1 dot1q 100 symmetric
 ethernet egress-filter strict
!
interface GigabitEthernet0/0/0/2.4 l2transport
 encapsulation untagged ingress source-mac 0000.1111.3333
 rewrite ingress tag push dot1q 101 symmetric
 ethernet egress-filter strict
!

interface GigabitEthernet0/0/0/0/3.1 l2transport
 encapsulation dot1q 1
 rewrite ingress tag translate 1-to-1 dot1q 4094 symmetric
 ethernet egress-filter disabled
 ethernet source-bypass-egress-filter
!
```

## レイヤ 2 プロトコル トンネリングの設定：例

ここでは、転送モードと反転モードでの L2PT の設定例を示します。

### 転送モードでの L2PT の設定

次に、転送モードで L2PT を設定する例を示します。

カスタマー側ルータ（カプセル化側）：

```
!
interface GigabitEthernet0/1/0/1
 negotiation auto
!
interface GigabitEthernet0/1/0/1.1 l2transport
 encapsulation default
 l2protocol cpsv tunnel
!
interface GigabitEthernet0/1/0/2
 negotiation auto
!
interface GigabitEthernet0/1/0/2.1 l2transport
 encapsulation default
```

```
!
l2vpn
 xconnect group examples
  p2p r1-connect
   interface GigabitEthernet0/1/0/1.1
   interface GigabitEthernet0/1/0/2.1
  !
 !
!
```

カスタマー側ルータ（カプセル化解除側）：

```
!
interface GigabitEthernet0/5/0/1
 negotiation auto
!
interface GigabitEthernet0/5/0/1.1 l2transport
 encapsulation default
!
interface GigabitEthernet0/5/0/2
 negotiation auto
!
interface GigabitEthernet0/5/0/2.1 l2transport
 encapsulation default
 l2protocol cpsv tunnel
!
l2vpn
 xconnect group examples
  p2p r2-connect
   interface GigabitEthernet0/5/0/1.1
   interface GigabitEthernet0/5/0/2.1
  !
 !
!
```

## 反転モードでの L2PT の設定

次に、反転モードで L2PT を設定する例を示します。

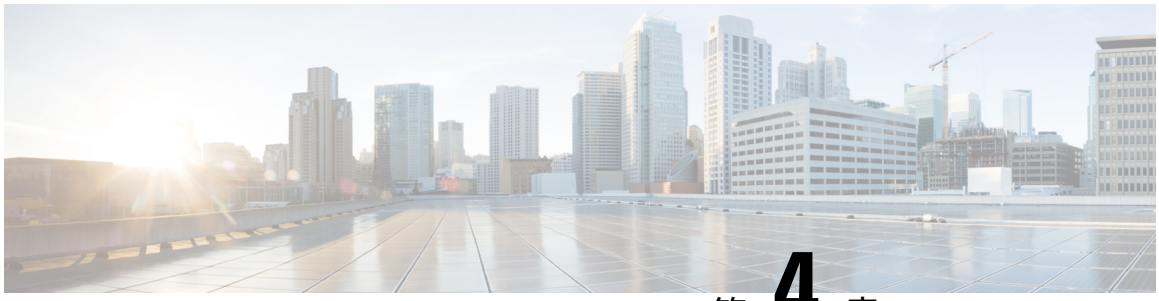
カスタマー側ルータ（カプセル化側）：

```
!
interface GigabitEthernet0/1/0/1
 negotiation auto
!
interface GigabitEthernet0/1/0/1.1 l2transport
 encapsulation untagged
 rewrite ingress tag push dot1q 100 symmetric
 ethernet egress-filter strict
!
interface GigabitEthernet0/1/0/2
 negotiation auto
!
interface GigabitEthernet0/1/0/2.1 l2transport
 encapsulation untagged
 rewrite ingress tag push dot1q 200 symmetric
 ethernet egress-filter strict
!
interface GigabitEthernet0/1/0/3
 negotiation auto
!
interface GigabitEthernet0/1/0/3.1 l2transport
 encapsulation dot1q 500
 rewrite ingress tag pop 1 symmetric
```

```
 l2protocol cpsv reverse-tunnel
 ethernet egress-filter strict
!
l2vpn
 bridge group examples
  bridge-domain r1-bridge
   interface GigabitEthernet0/1/0/1.1
   !
   interface GigabitEthernet0/1/0/2.1
   !
   interface GigabitEthernet0/1/0/3.1
   !
  !
 !
!
```

カスタマー側ルータ（カプセル化解除側）：

```
!
interface GigabitEthernet0/5/0/4
 negotiation auto
!
interface GigabitEthernet0/5/0/4.1 l2transport
 encapsulation dot1q 500
 rewrite ingress tag pop 1 symmetric
 l2protocol cpsv reverse-tunnel
 ethernet egress-filter strict
!
interface GigabitEthernet0/5/0/5
 negotiation auto
!
interface GigabitEthernet0/5/0/5.1 l2transport
 encapsulation untagged
 rewrite ingress tag push dot1q 100 symmetric
 ethernet egress-filter strict
!
interface GigabitEthernet0/5/0/6
 negotiation auto
!
interface GigabitEthernet0/5/0/6.1 l2transport
 encapsulation untagged
 rewrite ingress tag push dot1q 200 symmetric
 ethernet egress-filter strict
!
l2vpn
 bridge group examples
  bridge-domain r2-bridge
   interface GigabitEthernet0/5/0/4.1
   !
   interface GigabitEthernet0/5/0/5.1
   !
   interface GigabitEthernet0/5/0/6.1
   !
  !
 !
!
```



第 4 章

# リンク バンドルの設定

バンドルは、1 つ以上のポート グループを集約し、1 つのリンクとして扱うようにしたものです。1 つのバンドル内の各リンクの速度は異なっていてもよく、最も高速なリンクの速度は、最も低速なリンクの最大 4 倍とすることができます。各バンドルには、1 つの MAC、1 つの IP アドレス、1 つの設定セット（ACL または Quality of Service など）があります。

このルータでは、次のタイプのインターフェイスでバンドルがサポートされます。

- イーサネット インターフェイス
- VLAN サブインターフェイス

（注） バンドルには、モジュラ サービス カードとの 1 対 1 の関連付けはありません。



## リンクバンドルの設定の機能履歴

| リリース | 変更内容 |
|---|---|
| リリース 3.7.2 | この機能は、Cisco ASR 9000 シリーズ ルータで導入されました。 |

## リンク バンドルを設定するための前提条件

リンク バンドルを設定する前に、次のタスクと条件を満たしていることを確認してください。

- 適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。

  ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

- インターフェイスの IP アドレスがわかっていること。
- 設定するバンドルに含めるリンクがわかっていること。
- イーサネット リンク バンドルを設定する場合、ルータに少なくとも次のイーサネット ラインカードのいずれかが搭載されていること。
  - 2 ポート 10 ギガビット イーサネット ラインカード
  - 4 ポート 10 ギガビット イーサネット ラインカード
  - 8 ポート 10 ギガビット イーサネット ラインカード
  - 16 ポート 10 ギガビット イーサネット ラインカード
  - 20 ポート ギガビット イーサネット ラインカード
  - 40 ポート ギガビット イーサネット ラインカード

（注） 物理インターフェイス、PLIM、およびモジュラサービスカードの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Hardware Installation Guide*』を参照してください。

# リンク バンドルの設定に関する情報

リンク バンドル機能を設定するには、次の概念を理解している必要があります。

## リンク バンドルの概要

リンク バンドルは、1 つに束ねられたポートのグループであり、1 つのリンクとして振る舞います。リンク バンドルの利点は、次のとおりです。

- 複数のリンクが複数のラインカードにまたがり、1 つのインターフェイスを構成します。そのため、単一のリンクで障害が発生しても接続性は失われません。
- バンドルされたインターフェイスでは、バンドルの使用可能なすべてのメンバーにわたってトラフィックが転送されるため、帯域幅の可用性が向上します。したがって、バンドル内のリンクの1つで障害が発生した場合、トラフィックは使用可能なリンクを通過できます。パケット フローを中断することなく帯域幅を追加できます。

1 つのバンドル内の個々のリンクにはさまざまな速度を設定できますが、バンドル内のすべてのリンクが同じタイプである必要があります。

Cisco IOS XR ソフトウェアでは、次の方法によるイーサネット インターフェイスのバンドル構成をサポートしています。

- IEEE 802.3ad：バンドル内のすべてのメンバー リンクの互換性を確保するため、Link Aggregation Control Protocol（LACP）を採用した標準テクノロジー。互換性がないリンクや障害になったリンクは、バンドルから自動的に削除されます。
- EtherChannel：ユーザがリンクを設定してバンドルに参加させることができるシスコの専用テクノロジー。バンドル内のリンクに互換性があるかどうかを確認するための仕組みはありません。

# リンク バンドルの特性

このリストでは、リンクバンドルの特性と制限事項を説明します。

- LACP（Link Aggregation Control Protocol）を使用するかにかかわらず、すべてのタイプのイーサネット インターフェイスをバンドルできます。
- バンドル メンバーシップは、1 つのルータに搭載されている複数のラインカードにまたがることができます。
- 1 つのバンドルは最大 64 個の物理リンクをサポートします。
- 1 つのバンドル内でリンク速度が異なっていてもよく、バンドルのメンバー間で許容される速度の差は、最大 4 倍です。
- 物理層とリンク層の設定は、バンドルの個々のメンバー リンクに対して実行します。
- ネットワーク層プロトコルおよび上位層のアプリケーションの設定は、バンドル自体に対して実行します。
- バンドルは、管理上イネーブルまたはディセーブルにできます。
- バンドル内のそれぞれのリンクは、管理上イネーブルまたはディセーブルにできます。
- イーサネット リンク バンドルは、イーサネット チャネルと同様の方法で作成され、両方のエンド システムで同じコンフィギュレーションを入力します。
- バンドルに対して設定された MAC アドレスは、そのバンドル内の各リンクの MAC アドレスになります。
- LACP が設定されている場合、バンドル内の各リンクは、異なるメンバーに対して異なるキープアライブ周期を許可するよう設定できます。
- ロード バランシング（メンバー リンク間のデータの分散）は、パケットではなくフロー単位で実行されます。データはバンドル対するそのリンクの帯域幅に比例して、リンクに配信されます。
- QoS がサポートされており、各バンドル メンバーに均等に適用されます。

- CDP キープアライブや HDLC キープアライブなどのリンク層プロトコルは、バンドル内の各リンク上で独立して動作します。
- ルーティング アップデートや hello などの上位層プロトコルは、インターフェイス バンドルのどのメンバー リンク上でも送信されます。
- バンドルされたインターフェイスはポイント ツー ポイントです。
- リンクがバンドル内で distributing 状態になるには、その前にアップ状態になる必要があります。
- 1 つのバンドル内のすべてのリンクは、802.3ad（LACP）または EtherChannel（非 LACP）のいずれかを実行するように設定する必要があります。1 つのバンドル内の混合リンクはサポートされません。
- バンドルインターフェイスには、物理リンクと VLAN サブインターフェイスのみを含めることができます。
- リンク バンドルでのアクセス コントロール リスト（ACL）設定は、通常のインターフェイスでの ACL 設定と同じです。
- マルチキャスト トラフィックは、バンドルのメンバー上でロード バランスされます。特定のフローに対し、内部プロセスによってメンバー リンクが選択され、そのフローのすべてのトラフィックがそのメンバー上で送信されます。

# IEEE 802.3ad 規格

IEEE 802.3ad 規格では、一般にイーサネット リンク バンドルを構成する方法が定義されています。

バンドル メンバーとして設定された各リンクでは、この情報は、リンク バンドルの両端をホストするシステム間で交換されます。

- グローバルに一意のローカル システム ID
- リンクがメンバーになっているバンドルの ID（動作キー）
- リンクの ID（ポート ID）
- リンクの現在の集約ステータス

この情報は、リンク集約グループ ID（LAG ID）を構成するために使用されます。共通の LAG ID を共有するリンクは集約できます。個々のリンクには固有の LAG ID があります。

システム ID はルータを区別し、その一意性はシステムの MAC アドレスを使用することで保証されます。バンドル ID とリンク ID は、それを割り当てるルータでだけ意味を持ち、2 つのリンクが同じ ID を持たないことと、2 つのバンドルが同じ ID を持たないことが保証される必要があります。

ピア システムからの情報はローカル システムの情報と組み合わされ、バンドルのメンバーとして設定されたリンクの互換性が判断されます。

ルータ内のバンドル MAC アドレスは、バックプレーン内の予約済み MAC アドレスのセットに由来します。この MAC アドレスは、バンドル インターフェイスが存在する限り、バンドルにとどまります。バンドルは、ユーザが別の MAC アドレスを設定するまで、この MAC アドレスを使用します。バンドルの MAC アドレスは、バンドル トラフィックを通過させる際にすべてのメンバー リンクによって使用されます。バンドルに対して設定されたすべてのユニキャスト アドレスまたはマルチキャスト アドレスも、すべてのメンバー リンクで設定されます。

（注） MACアドレスを変更するとパケット転送に影響を与えるおそれがあるため、MACアドレスは変更しないことを推奨します。

# LACP バンドルインターフェイスの非リバーティブ動作

デフォルトでは、LACP でプライオリティが高いポートが再び動作可能になると、そのポートがアクティブポートになります。この復帰を回避するには、**lacp non-revertive** コマンドを実行します。これにより、プライオリティの高いポートが動作可能になった後も、プライオリティの低いポートが引き続きアクティブポートとなります。これにより、現在アクティブなプライオリティの低いポートをスタンバイ状態にして、運用可能になったプライオリティの高いポートを介してトラフィックを転送するときに発生する可能性があるトラフィックの中断を回避できます。

# QoS およびリンク バンドル

入力方向では、バンドルのローカルインスタンスにQoSが適用されます。各バンドルはキューのセットに関連付けられます。QoSは、バンドル上で設定されているさまざまなネットワーク層プロトコルに適用されます。

出方向では、メンバー リンクへの参照を持つバンドルに QoS が適用されます。QoS は、メンバーの帯域幅の合計に基づいて適用されます。

QoS が入力または出力方向のいずれかのバンドルに適用される場合、QoS は各メンバー インターフェイスに適用されます。

リンクバンドル機能は、『Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Configuration Guide』に記載されているすべての QoS 機能をサポートします。

リンク バンドル機能は、次の QoS 機能をサポートします。

- 高優先順位/低優先順位：最大帯域幅は、バンドル インターフェイスの帯域幅のパーセンテージとして計算されます。このパーセンテージは出力上のすべてのメンバー リンクに適用されるか、入力上のローカル バンドル インスタンスに適用されます。
- 保証される帯域幅：パーセンテージで提供され、すべてのメンバー リンクに適用されます。
- トラフィック シェーピング：パーセンテージで提供され、すべてのメンバー リンクに適用されます。

- WRED：最小および最大パラメータは、メンバー リンクまたはバンドル インスタンスごとの正しい比率に変換され、バンドルに適用されます。
- マーキング：ポリシーに従ったパケットの QoS レベルの変更プロセス。
- テール ドロップ：キューが一杯のときにパケットはドロップされます。

# イーサネット リンク バンドル上の VLAN

802.1Q VLAN サブインターフェイスを 802.3ad イーサネット リンク バンドル上で設定できます。イーサネット リンク バンドル上に VLAN を追加するときには、次の点に注意してください。

- バンドルごとに許可される VLAN の最大数は 4000 です。
- 各ルータに許可されるバンドル VLAN の最大数は、128000 です。

（注） バンドル VLAN のメモリ要件は、標準の物理インターフェイスよりも若干多くなります。

バンドル上でVLANサブインターフェイスを作成するには、次のように、**interface Bundle-Ether** コマンドを使用して VLAN サブインターフェイス インスタンスを追加します。

**interface Bundle-Ether** `instance.subinterface`

イーサネット リンク バンドル上で VLAN を作成した後、すべての物理 VLAN サブインターフェイス コンフィギュレーションがそのリンク バンドル上でサポートされます。

# リンク バンドルの設定の概要

リンク バンドルの設定プロセスの一般的な概要を次の手順に示します。リンクをバンドルに追加する前に、リンクから以前のネットワーク層コンフィギュレーションをすべてクリアする必要があることに注意してください。

1. グローバル コンフィギュレーション モードで、リンク バンドルを作成します。イーサネット リンク バンドルを作成するには、**interface Bundle-Ether**コマンドを入力します。
2. **ipv4 address** コマンドを使用して、IP アドレスとサブネットマスクを仮想インターフェイスに割り当てます。
3. インターフェイス コンフィギュレーション サブモードで **bundle id** コマンドを使用し、ステップ 1 で作成したバンドルにインターフェイスを追加します。1 つのバンドルに最大 32 個のリンクを追加できます。

（注） リンクは、そのリンクのインターフェイス コンフィギュレーション サブモードからバンドルのメンバーに設定できます。

## カードのフェールオーバー時のノンストップ フォワーディング

Cisco IOS XR ソフトウェアは、アクティブおよびスタンバイ RSP カード間でのフェールオーバー時のノンストップ フォワーディングをサポートしています。ノンストップ フォワーディングを使用すると、フェールオーバーが発生したときにリンク バンドルの状態が変化しません。

たとえば、アクティブな RSP が障害になった場合、スタンバイ RSP が動作可能になります。障害になった RSP のコンフィギュレーション、ノードの状態、チェックポイント データは、スタンバイ RSP に複製されます。スタンバイ RSP がアクティブ RSP になったとき、バンドルされたインターフェイスはすべて存在します。

（注） フェールオーバー先は常にスタンバイ RSP です。

（注） スタンバイ インターフェイス コンフィギュレーションが維持されることを保証するために何かを設定する必要はありません。

## リンクのフェールオーバー

バンドルのメンバー リンクの1つに障害が発生すると、トラフィックは動作可能な残りのメンバー リンクにリダイレクトされ、トラフィック フローは中断されません。

## バンドル インターフェイス：冗長性、ロード シェアリング、集約

バンドルは、1 つ以上のポート グループを集約し、1 つのリンクとして扱うようにしたものです。1 つのバンドル内の各リンクの速度は異なっていてもよく、最も高速なリンクの速度は、最も低速なリンクの最大 4 倍とすることができます。各バンドルには、1 つの MAC、1 つの IP アドレス、1 つの設定セット（ACL または Quality of Service など）があります。

このルータでは、次のタイプのインターフェイスでバンドルがサポートされます。

- イーサネット インターフェイス
- VLAN サブインターフェイス

# リンク バンドルの設定方法

## イーサネット リンク バンドルの設定

ここでは、イーサネット リンク バンドルの設定方法について説明します。

（注） イーサネット リンク バンドルでは MAC アカウンティングはサポートされていません。

（注） イーサネット バンドルをアクティブにするためには、バンドルの両方の接続ポイントで同じ設定を行う必要があります。

イーサネット リンク バンドルを作成するには、次の手順のように、バンドルを作成し、そのバンドルにメンバー インターフェイスを追加します。

## 手順の概要

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **ipv4 address** *ipv4-address-address mask*
4. **bundle minimum-active bandwidth** *kbps* （オプション）
5. **bundle minimum-active links** リンク数 （オプション）
6. **bundle maximum-active links** リンク数 （オプション）
7. **bundle maximum-active links** *links* **hot-standby**
8. **exit**
9. **interface {GigabitEthernet | TenGigE}***instance*
10. **bundle id** *bundle-id* [ **mode** { **active** | **on** | **passive**} ]
11. **no shutdown**（任意）
12. **exit**
13. ステップ 2 で作成したバンドルにさらにリンクを追加するには、ステップ 8 から 11 を繰り返します。
14. **commit** コマンドまたは **end** コマンドを使用します。
15. **exit**
16. **exit**
17. 接続のリモート エンドでステップ 1 から 15 を実行します。
18. **show bundle Bundle-Ether** *bundle-id* [ **reasons** ] （オプション）
19. **show lacp Bundle-Ether** *bundle-id* （オプション）

## 手順の詳細

### ステップ 1 configure

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

### ステップ 2 interface Bundle-Ether *bundle-id*

例：

```
RP/0/RSP0/cpu 0: router(config)# interface Bundle-Ether 3
```

新しいイーサネット リンク バンドルを作成し名前を付与します。

この **interface Bundle-Ether** コマンドを実行すると、インターフェイス設定サブモードが開始されます。このモードでは、インターフェイス固有の設定コマンドを入力できます。インターフェイス設定サブモードを終了して通常のグローバル コンフィギュレーション モードに戻るには、**exit** コマンドを使用します。

**ステップ 3** **ipv4 address** *ipv4-address-address mask*

例：

```
RP/0/RSP0/cpu 0: router(config-if)# ipv4 address 10.1.2.3 255.0.0.0
```

**ipv4 address**コンフィギュレーション サブコマンドを使用して、IP アドレスとサブネット マスクを仮想インターフェイスに割り当てます。

**ステップ 4** **bundle minimum-active bandwidth** *kbps* （オプション）

例：

```
RP/0/RSP0/cpu 0: router(config-if)# bundle minimum-active bandwidth 580000
```

ユーザがバンドルを起動状態にするために必要な最小帯域幅を設定します。

**ステップ 5** **bundle minimum-active links** リンク数 （オプション）

例：

```
RP/0/RSP0/cpu 0: router(config-if)# bundle minimum-active links 2
```

特定のバンドルをアップ状態にするために必要なアクティブリンク数を設定します。

**ステップ 6** **bundle maximum-active links** リンク数 （オプション）

例：

```
RP/0/RSP0/cpu 0: router(config-if)# bundle maximum-active links 1
```

1 個のアクティブリンクと、アクティブリンクに障害が発生した場合に、バンドルに迅速に引き継ぐことができるスタンバイモードの 1 個のリンクを指定します（1:1 保護）。

1 つのバンドルで許可されるデフォルトのアクティブ リンク数は 8 です。

（注） **bundle maximum-active** コマンドを実行すると、バンドル内で最もプライオリティが高いリンクのみがアクティブになります。プライオリティは、**bundle port-priority** コマンドの値に基づいて決定されます（値が小さいほど、プライオリティが高くなります）。したがって、アクティブにするリンクに高いプライオリティを設定することを推奨します。

**ステップ 7** **bundle maximum-active links** *links* **hot-standby**

例：

```
RP/0/RSP0/cpu 0: router(config-if)# bundle maximum-active links 1 hot-standby
```

**hot-standby** キーワードは、バンドルが一時的に最小リンク数または帯域幅しきい値未満になる間にスイッチオーバーまたはスイッチバックイベントでバンドルフラップを回避するために役立ちます。

これは、このために、wait-while タイマーと suppress-flaps タイマーのデフォルト値を設定します。

**ステップ 8** **exit**

**例**：

```
RP/0/RSP0/cpu 0: router(config-#if)exit
```

イーサネット リンク バンドルのインターフェイス コンフィギュレーション サブモードを終了します。

**ステップ 9** **interface {GigabitEthernet | TenGigE}***instance*

**例**：

```
RP/0/RSP0/cpu 0: router(config)# interface TenGigE 1/0/0/0
```

指定したインターフェイスのインターフェイス コンフィギュレーション モードを開始します。

**GigabitEthernet** キーワードまたは **TenGigE** キーワードを入力して、インターフェイスタイプを指定します。instance 引数には、*rack/slot/module* 形式のノード ID を指定します。

混合帯域幅のバンドル メンバの設定は、1:1 冗長性が設定されている場合にだけサポートされます（これは、10 GigabitEthernet インターフェイスのバックアップとして 1 GigabitEthernet メンバしか設定できないことを意味します）。

（注） 混合リンク バンドル モードは、アクティブ/スタンバイ動作が設定されている場合にだけサポートされます（通常はスタンバイ モードで低速リンクです）。

**ステップ 10** **bundle id** *bundle-id* [ **mode** { **active** | **on** | **passive**} ]

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# bundle-id 3
```

指定したバンドルにリンクを追加します。

バンドル上でアクティブ LACP またはパッシブ LACP をイネーブルにするには、オプションの **mode active** キーワードまたは **mode passive** キーワードをコマンド文字列に追加します。

LACP をサポートせずにバンドルにリンクを追加するには、オプションの **mode on** キーワードをコマンド文字列に追加します。

（注） **mode** キーワードを指定しない場合は、デフォルトのモードは **on** になります（LACP はポート上で動作しません）。

**ステップ 11** **no shutdown**（任意）

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# no shutdown
```

リンクがダウン状態の場合はアップ状態にします。**no shutdown**コマンドは、設定とリンクの状態に応じて、リンクをアップ状態またはダウン状態に戻します。

**ステップ 12** **exit**

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# exit
```

イーサネット リンク バンドルのインターフェイス コンフィギュレーション サブモードを終了します。

**ステップ 13** ステップ 2 で作成したバンドルにさらにリンクを追加するには、ステップ 8 から 11 を繰り返します。

**ステップ 14** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

**ステップ 15** **exit**

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# exit
```

インターフェイス コンフィギュレーション モードを終了します。

**ステップ 16** **exit**

**例**：

```
RP/0/RSP0/cpu 0: router(config)# exit
```

グローバル コンフィギュレーション モードを終了します。

**ステップ 17** 接続のリモート エンドでステップ 1 から 15 を実行します。

リンク バンドルの他端をアップ状態にします。

**ステップ 18** **show bundle Bundle-Ether** *bundle-id* [ **reasons** ]（オプション）

**例**：

```
RP/0/RSP0/cpu 0: router# show bundle Bundle-Ether 3 reasons
```

指定したイーサネットリンクバンドルに関する情報を表示します

**ステップ 19** **show lacp Bundle-Ether** *bundle-id*（オプション）

**例**：

```
RP/0/RSP0/cpu 0: router # show lacp Bundle-Ether 3
```

LACP ポートとそのピアに関する詳細情報を表示します。

# VLAN バンドルの設定

ここでは、VLANバンドルの設定方法について説明します。VLANバンドルの作成では、主に次の 3 つの作業を行います。

1. イーサネット バンドルを作成します。
2. VLAN サブインターフェイスを作成し、イーサネット バンドルに割り当てます。
3. イーサネット リンクをイーサネット バンドルに割り当てます。

これらの作業について、以降の手順で詳しく説明します。

（注） VLANバンドルをアクティブにするには、バンドル接続の両端で同じ設定を行う必要があります。

VLAN リンク バンドルの作成について、次の手順で説明します。

## 手順の概要

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **ipv4 address** *ipv4-address mask*
4. **bundle minimum-active bandwidth** *kbps*（オプション）
5. **bundle minimum-active links** *links*（オプション）
6. **bundle maximum-active links** *links*（オプション）
7. **exit**
8. **interface Bundle-Ether** *bundle-id.vlan-id*
9. **encapsulation dot1q** *vlan-id*
10. **ipv4 address** *ip-address mask*
11. **no shutdown**
12. **exit**
13. ステップ 2 で作成したバンドルにさらに VLAN を追加するには、ステップ 7 から 12 を繰り返します
14. **commit** コマンドまたは **end** コマンドを使用します。
15. **exit**
16. **exit**
17. **show ethernet trunk bundle-Ether** *instance*
18. **configure**
19. **interface {GigabitEthernet | TenGigE}***instance*
20. **bundle id `bundle-id` [mode {active | on | passive}]**

21. **no shutdown**
22. ステップ 2 で作成したバンドルにさらにイーサネット インターフェイスを追加するには、ステップ 19 から 21 を繰り返します。
23. **commit** コマンドまたは **end** コマンドを使用します。
24. 接続のリモート エンドでステップ 1 から 23 を実行します。
25. **show bundle Bundle-Ether** *bundle-id* [ **reasons** ]
26. **show ethernet trunk bundle-Ether** *instance*

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モード を開始します。

**ステップ 2** **interface Bundle-Ether** *bundle-id*

例：

```
RP/0/RSP0/cpu 0: router#(config)# interface Bundle-Ether 3
```

新しいイーサネット リンク バンドルを作成し名前を付与します。

この **interface Bundle-Ether** コマンドを実行すると、インターフェイス設定サブモードが開始されます。このモードでは、インターフェイス固有の設定コマンドを入力できます。インターフェイス設定サブモードを終了して通常のグローバル コンフィギュレーション モードに戻るには、**exit** コマンドを使用します。

**ステップ 3** **ipv4 address** *ipv4-address mask*

例：

```
RP/0/RSP0/cpu 0: router(config-if)# ipv4 address 10.1.2.3 255.0.0.0
```

**ipv4 address**コンフィギュレーション サブコマンドを使用して、IP アドレスとサブネット マスクを仮想インターフェイスに割り当てます。

**ステップ 4** **bundle minimum-active bandwidth** *kbps*（オプション）

例：

```
RP/0/RSP0/cpu 0: router(config-if) # bundle minimum-active bandwidth 580000
```

（任意）ユーザがバンドルをアップ状態にする前に必要な最小帯域幅を設定します。

**ステップ 5** **bundle minimum-active links** *links*（オプション）

例：

```
RP/0/RSP0/cpu 0: router(config-if)# bundle minimum-active links 2
```

（任意）特定のバンドルをアップ状態にする前に必要なアクティブ リンク数を設定します。

**ステップ 6** **bundle maximum-active links** *links*（オプション）

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# bundle maximum-active links 1
```

（任意）1 個のアクティブ リンクと、アクティブ リンクに障害が発生した場合に、バンドルに迅速に引き継ぐことができるスタンバイ モードの 1 個のリンクを指定します（1：1 保護）。

（注） 1 つのバンドルで許可されるデフォルトのアクティブ リンク数は 8 です。

（注） **bundle maximum-active** コマンドを実行すると、バンドル内で最もプライオリティが高いリンクのみがアクティブになります。プライオリティは、**bundle port-priority** コマンドの値に基づいて決定されます（値が小さいほど、プライオリティが高くなります）。したがって、アクティブにするリンクに高いプライオリティを設定することを推奨します。

**ステップ 7** **exit**

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# exit
```

インターフェイス設定サブモードを終了します。

**ステップ 8** **interface Bundle-Ether** *bundle-id.vlan-id*

**例**：

```
RP/0/RSP0/cpu 0: router#(config)#interface Bundle-Ether 3.1
```

新しい VLAN を作成し、その VLAN をステップ 2 で作成したイーサネット バンドルに割り当てます。

*bundle-id* 引数には、ステップ 2 で作成したバンドル ID を指定します。

*vlan-id* にはサブインターフェイス識別子を指定します。範囲は 1 ～ 4094 です（0 と 4095 は予約されています）。

（注） *vlan-id* 引数を **interface Bundle-Ether** *bundle-id* コマンドに指定すると、サブインターフェイス設定モードが開始されます。

**ステップ 9** **encapsulation dot1q** *vlan-id*

**例**：

```
RP/0/RSP0/cpu 0: router#(config-subif)# encapsulation dot1q 10
```

VLAN をサブインターフェイスに割り当てます。

*vlan-id* 引数にはサブインターフェイス ID を指定します。範囲は 1 ～ 4094 です（0 と 4095 は予約されています）。

**ステップ 10** **ipv4 address** *ip-address mask*

**例**：

```
RP/0/RSP0/cpu 0: router#(config-subif)# ipv4 address 10.1.2.3/24
```

IP アドレスおよびサブネット マスクをサブインターフェイスに割り当てます。

**ステップ 11** **no shutdown**

**例**：

```
RP/0/RSP0/cpu 0: router(config-subif) # no shutdown
```

（任意）リンクがダウン状態の場合はアップ状態にします。no shutdown コマンドは、設定とリンクの状態に応じて、リンクをアップ状態またはダウン状態に戻します。

**ステップ 12** **exit**

**例**：

```
RP/0/RSP0/cpu 0: router(config-subif)#exit
```

VLAN サブインターフェイスのサブインターフェイス コンフィギュレーション モードを終了します。

**ステップ 13** ステップ 2 で作成したバンドルにさらに VLAN を追加するには、ステップ 7 から 12 を繰り返します

（任意）バンドルにさらにサブインターフェイスを追加します。

**ステップ 14** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

**ステップ 15** **exit**

**例**：

```
RP/0/RSP0/cpu 0: router (config-subif)# exit
```

インターフェイス コンフィギュレーション モードを終了します。

**ステップ 16** **exit**

**例**：

```
RP/0/RSP0/cpu 0: router (config)# exit
```

グローバル コンフィギュレーション モード を終了します。

**ステップ 17** **show ethernet trunk bundle-Ether** *instance*

**例**：

```
RP/0/RSP0/cpu 0: routerrouter# show ethernet trunk bundle-ether 5
```

（任意）インターフェイス コンフィギュレーションを表示します。

イーサネット バンドル インスタンスの範囲は 1 ～ 65535 です。

**ステップ 18** **configure**

例：

```
RP/0/RSP0/cpu 0: router#  configure
```

グローバル コンフィギュレーション モード を開始します。

**ステップ 19** **interface {GigabitEthernet | TenGigE}***instance*

例：

```
RP/0/RSP0/cpu 0: router(config)# interface TenGigE 1/0/0/0
```

指定したインターフェイスのインターフェイス コンフィギュレーション モードを開始します。

*instance* 引数には、*rack/slot/module* 形式のノード ID を指定します。

（注） リンク バンドルの両端にイーサネット インターフェイスを追加するまでは、VLAN バンドルはアクティブになりません。

**ステップ 20** **bundle id `bundle-id` [mode {active | on | passive}]**

例：

```
RP/0/RSP0/cpu 0: router(config-if)#  bundle-id 3
```

ステップ 2 から 13 で設定したバンドルにイーサネット インターフェイスを追加します。

バンドル上でアクティブ LACP またはパッシブ LACP をイネーブルにするには、オプションの **mode active** キーワードまたは **mode passive** キーワードをコマンド文字列に追加します。

LACP をサポートせずにバンドルにインターフェイスを追加するには、オプションの **mode on** キーワードをコマンド文字列に追加します。

（注） **mode** キーワードを指定しない場合は、デフォルトのモードは **on** になります（LACP はポート上で動作しません）。

**ステップ 21** **no shutdown**

例：

```
RP/0/RSP0/cpu 0: router(config-if)# no shutdown
```

（任意）リンクがダウン状態の場合はアップ状態にします。**no shutdown** コマンドは、設定とリンクの状態に応じて、リンクをアップ状態またはダウン状態に戻します。

**ステップ 22** ステップ 2 で作成したバンドルにさらにイーサネット インターフェイスを追加するには、ステップ 19 から 21 を繰り返します。

**ステップ 23** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

**ステップ 24** 接続のリモート エンドでステップ 1 から 23 を実行します。

リンク バンドルの他端をアップ状態にします。

**ステップ 25** **show bundle Bundle-Ether** *bundle-id* [ **reasons** ]

**例**：

```
RP/0/RSP0/cpu 0: router#show bundle Bundle-Ether 3 reasons
```

（任意）指定したイーサネット リンク バンドルに関する情報を表示します。

**show bundle Bundle-Ether** コマンドを実行すると、指定したバンドルに関する情報が表示されます。バンドルが正しく設定されており、トラフィックを伝送している場合は、**show bundle Bundle-Ether** コマンドの出力の State フィールドに数値「4」が表示されます。これは、指定された VLAN バンドル ポートが「分散している」ことを意味します。

**ステップ 26** **show ethernet trunk bundle-Ether** *instance*

**例**：

```
RP/0/RSP0/cpu 0: router# show ethernet trunk bundle-ether 5
```

（任意）インターフェイス コンフィギュレーションを表示します。

イーサネット バンドル インスタンスの範囲は 1 ～ 65535 です。

# リンク バンドルの設定例

## LACP が動作する EtherChannel バンドル：例

次に、2 つのポートを結合して、LACP が動作する EtherChannel バンドルを構成する例を示します。

```
RP/0/RSP0/CPU0:Router# config
RP/0/RSP0/CPU0:Router(config)# interface Bundle-Ether 3
RP/0/RSP0/CPU0:Router(config-if)# ipv4 address 1.2.3.4/24
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active links 1
RP/0/RSP0/CPU0:Router(config-if)# exit
RP/0/RSP0/CPU0:Router(config)# interface TenGigE 0/3/0/0
RP/0/RSP0/CPU0:Router(config-if)# bundle id 3 mode active
RP/0/RSP0/CPU0:Router(config-if)# no shutdown
RP/0/RSP0/CPU0:Router(config)# exit
RP/0/RSP0/CPU0:Router(config)# interface TenGigE 0/3/0/1
RP/0/RSP0/CPU0:Router(config-if)# bundle id 3 mode active
```

```
RP/0/RSP0/CPU0:Router(config-if)# no shutdown
RP/0/RSP0/CPU0:Router(config-if)# exit
```

# イーサネット バンドル上での VLAN の作成：例

次に、イーサネット バンドル上で 2 つの VLAN を作成し起動状態にする例を示します。

```
RP/0/RSP0/CPU0:Router# config
RP/0/RSP0/CPU0:Router(config)# interface Bundle-Ether 1
RP/0/RSP0/CPU0:Router(config-if)# ipv4 address 1.2.3.4/24
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active links 1
RP/0/RSP0/CPU0:Router(config-if)# exit
RP/0/RSP0/CPU0:Router(config)# interface Bundle-Ether 1.1
RP/0/RSP0/CPU0:Router(config-subif)# encapsulation dot1q 10
RP/0/RSP0/CPU0:Router(config-subif)# ip addr 10.2.3.4/24
RP/0/RSP0/CPU0:Router(config-subif)# no shutdown
RP/0/RSP0/CPU0:Router(config-subif)# exit
RP/0/RSP0/CPU0:Router(config)# interface Bundle-Ether 1.2
RP/0/RSP0/CPU0:Router(config-subif)# encapsulation dot1q 20
RP/0/RSP0/CPU0:Router(config-subif)# ip addr 20.2.3.4/24
RP/0/RSP0/CPU0:Router(config-subif)# no shutdown
RP/0/RSP0/CPU0:Router(config-subif)# exit
RP/0/RSP0/CPU0:Router(config)#interface tengige 0/1/5/7
RP/0/RSP0/CPU0:Router(config-if)# bundle-id 1 mode act
RP/0/RSP0/CPU0:Router(config-if)# commit
RP/0/RSP0/CPU0:Router(config-if)# exit
RP/0/RSP0/CPU0:Router(config)# exit
RP/0/RSP0/CPU0:Router # show ethernet trunk bundle-ether 1
```

# Cisco 7600 EtherChannel に接続された ASR 9000 リンクバンドル：例

次に、ASR 9000 シリーズ ルータ（ASR-9010）と、L2 および L3 サービスの両方をサポートするメトロ イーサネット ネットワーク内の Cisco 7600 シリーズ ルータ（P19_C7609-S）間のバンドルのエンドツーエンドの例を示します。

Cisco ASR 9000 シリーズルータでは、バンドルは、LACP、1:1 リンク保護、2 つの L2 サブインターフェイス、2 つのレイヤ 3 サブインターフェイスで設定されます。

**IOS XR 側**：

```
hostname PE44_IOS-XR_Router

interface Bundle-Ether16
 description Connect to P19_C7609-S Port-Ch 16
 mtu 9216
 no ipv4 address
 bundle maximum-active links 1
!
interface Bundle-Ether16.160 l2transport
 description Connect to P19_C7609-S Port-Ch 16 EFP 160
 encapsulation dot1q 160
!
interface Bundle-Ether16.161 l2transport
 description Connect to P19_C7609-S Port-Ch 16 EFP 161
 encapsulation dot1q 161
!
interface Bundle-Ether16.162
```

```
 description Connect to P19_C7609-S Port-Ch 16.162
 ipv4 address 10.194.8.44 255.255.255.0
 encapsulation dot1q 162
!
interface Bundle-Ether16.163
 description Connect to P19_C7609-S Port-Ch 16.163
 ipv4 address 10.194.12.44 255.255.255.0
 encapsulation dot1q 163
!


interface TenGigE 0/1/0/16
 description Connected to P19_C7609-S GE 8/0/16
 bundle id 16 mode active
 bundle port-priority 1
!
interface TenGigE 0/1/0/17
 description Connected to P19_C7609-S GE 8/0/17
 bundle id 16 mode active
 bundle port-priority 2
!
```

### IOS XR 側：CE デバイスへの接続：

```
hostname PE44_IOS-XR_Router

interface TenGigE 0/1/0/3.160 l2transport
 description VLAN 160 over BE 16.160
 encapsulation dot1q 100 second-dot1q 160
 rewrite ingress tag pop 1 symmetric
!
interface TenGigE 0/1/0/3.161 l2transport
 description VLAN 161 over BE 16.161
 encapsulation dot1q 161
!
l2vpn
 !
 xconnect group 160
  p2p 160
   interface Bundle-Ether16.160
   interface TenGigE 0/1/0/3.160
   description VLAN_160_over_BE_16.160
  !
!
 xconnect group 161
  p2p 161
   interface Bundle-Ether16.161
   interface TenGigE 0/1/0/3.161
   description VLAN_161_over_BE_16.161
  !
 !
```

### IOS XR 側：CE デバイス：

```
hostname PE64_C3750-ME
!
vlan 161
!
interface TenGigE 1/0/1
 description Connected to PE65_ME-C3400 GE 0/1
 switchport access vlan 100
 switchport mode dot1q-tunnel
!
```

```
interface TenGigE 1/0/2
 description Connected to PE44_IOS-XR_Router GE 0/1/0/3
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 100,161
 switchport mode trunk
!
interface Vlan161
 description VLAN 161 over BE 16.161 on PE44
 ip address 161.0.0.64 255.255.255.0
!



hostname PE65_ME-C3400
!
vlan 160
!
interface TenGigE 0/1
 description Connected to PE64_C3750-ME GE 1/0/1
 port-type nni
 switchport trunk allowed vlan 160
 switchport mode trunk
!
interface Vlan160
 description VLAN 160 over BE 16.160 on PE44
 ip address 160.0.0.65 255.255.255.0
!
```

**IOS 側**：

```
hostname P19_C7609-S

port-channel load-balance src-dst-port
!
interface Port-channel16
 description Connected to PE44_IOS-XR_Router BE 16
 mtu 9202
 no ip address
 logging event link-status
 logging event status
 speed nonegotiate
 mls qos trust dscp
 lacp fast-switchover
 lacp max-bundle 1
 service instance 160 ethernet
  description Connected to PE44_IOS-XR_Router BE 16.160
  encapsulation dot1q 160
 !
 service instance 161 ethernet
  description Connected to PE44_IOS-XR_Router BE 16.161
  encapsulation dot1q 161
 !
!
interface Port-channel16.162
 description Connected to PE44_IOS-XR_Router BE 16.162
 encapsulation dot1Q 162
 ip address 10.194.8.19 255.255.255.0
!
interface Port-channel16.163
 description Connected to PE44_IOS-XR_Router BE 16.163
 encapsulation dot1Q 163
 ip address 10.194.12.19 255.255.255.0
!
```

```
interface TenGigE 8/0/16
 no shut
 description Connected to PE44_IOS-XR_Router GE 0/1/0/16
 mtu 9202
 no ip address
 logging event link-status
 logging event status
 speed nonegotiate
 no mls qos trust dscp
 lacp port-priority 1
 channel-protocol lacp
 channel-group 16 mode active
!
interface TenGigE 8/0/17
 no shut
 description Connected to PE44_IOS-XR_Router GE 0/1/0/17
 mtu 9202
 no ip address
 logging event link-status
 logging event status
 speed nonegotiate
 no mls qos trust dscp
 lacp port-priority 2
 channel-protocol lacp
 channel-group 16 mode active
!
```

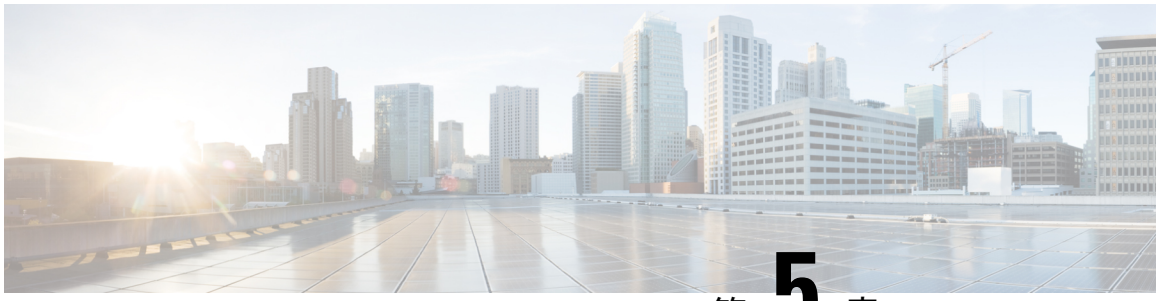
### IOS 側：CE デバイスへの接続：

```
hostname P19_C7609-S

interface TenGigE 8/0/7
 description Connected to PE62_C3750-ME GE 1/0/2
 mtu 9000
 no ip address
 speed nonegotiate
 mls qos trust dscp
 service instance 160 ethernet
  description VLAN 160 over Port-Ch 16
  encapsulation dot1q 100 second-dot1q 160
  rewrite ingress tag pop 1 symmetric
 !
 service instance 161 ethernet
  description VLAN 161 over Port-Ch 16
  encapsulation dot1q 161
 !
!
connect eline-161 Port-channel16 161 TenGigE 8/0/7 161
 !
!
connect eline-160 Port-channel16 160 TenGigE 8/0/7 160
 !
!
```

### IOS 側：CE デバイス：

```
hostname PE62_C3750-ME
!
vlan 161
!
interface TenGigE 1/0/1
 description Connected to PE63_ME-C3400 GE 0/1
 switchport access vlan 100
```

```
 switchport mode dot1q-tunnel
!
interface TenGigE 1/0/2
 description Connected to P19_C7609-S GE 8/0/7
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 100,161
 switchport mode trunk
!
interface Vlan161
 description VLAN 161 over Port-Chan 16 on P19
 ip address 161.0.0.62 255.255.255.0
!



hostname PE63_ME-C3400
!
vlan 160
!
interface TenGigE 0/1
 description Connected to PE62_C3750-ME GE 1/0/1
 port-type nni
 switchport trunk allowed vlan 160
 switchport mode trunk
!
interface Vlan160
 description VLAN 160 over Port-Chan 16 on P19
 ip address 160.0.0.63 255.255.255.0
!
```



第 5 章

# ポイントツーポイント レイヤ 2 サービスの実装

このモジュールでは、ポイントツーポイントレイヤ 2（L2）接続の概念および設定情報を提供します。

次のポイントツーポイント サービスがサポートされます。

- ローカルスイッチング：単一の Cisco ASR 9000 シリーズ ルータへのポイントツーポイント内部回線（ローカル接続とも呼ばれます）。
- 疑似回線：Cisco ASR 9000 シリーズ ルータからの仮想ポイントツーポイント回線。疑似回線は、MPLS 上で実装されます。

（注） ポイントツーポイントレイヤ 2 サービスは、MPLS レイヤ 2 VPN とも呼ばれます。

（注） Cisco ASR 9000 シリーズ ルータでのポイントツーポイントレイヤ 2 サービスの詳細、およびこのモジュールに記載されているコマンドの説明については、「関連ドキュメント」セクションを参照してください。設定作業の実行中に必要になることのある他のコマンドのドキュメントを見つけるには、Cisco IOS XR ソフトウェア マスター コマンド インデックスで、オンライン検索してください。

### ポイントツーポイントレイヤ 2 サービスの実装機能の履歴

| リリース | 変更内容 |
|---|---|
| リリース 3.7.2 | この機能が導入されました。 |
| リリース 3.9.0 | スケール拡張機能が導入されました。 |
| リリース 4.0.0 | Any Transport over MPLS（AToM）機能のサポートが追加されました。 |

| リリース | 変更内容 |
|---|---|
| リリース 4.0.1 | 次の機能のサポートが追加されました。<br>• 疑似回線のロード バランシング<br>• Any Transport over MPLS（AToM）機能<br>• HDLC over MPLS（HDLCoMPLS）<br>• PPP over MPLS（PPPoMPLS） |
| リリース 4.1.0 | Flexible ルータ ID 機能のサポートが追加されました。 |
| リリース 4.2.0 | 次の機能のサポートが追加されました。<br>• MPLS トランスポート プロファイル<br>• Circuit EMulation（CEM）over Packet |
| リリース 4.3.0 | L2VPN ノンストップ ルーティング機能のサポートが追加されました。 |
| リリース 4.3.1 | 次の機能のサポートが追加されました。<br>• L2TPv3 over IPv6 トンネル<br>• ATMoMPLS セルリレー VP モード<br>• GTP ロードバランシング |
| リリース 5.1.0 | 次の機能のサポートが追加されました。<br>• ATM/CEMoMPLS の双方向疑似回線（PW）<br>• マルチセグメント PW の PW グループ化<br>• ATM/CEMoMPLS のホットスタンバイ PW<br>• MR-APS のホットスタンバイ PW との統合 |
| リリース 5.1.2 | 次のサポートが追加されました。<br>• 動的単一セグメント疑似回線<br>• 疑似回線の障害が発生した後のネットワーク コンバージェンスの高速化 |

| リリース | 変更内容 |
|---|---|
| リリース 6.1.2 | 次の機能に対するサポートが追加されました。<br>• L2TPv3 over IPv4<br>• PWHE インターフェイスおよびアクセス疑似回線を設定するための EVPN-VPWS 拡張 |



# ポイントツーポイント レイヤ 2 サービス実装の前提条件

適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。

ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

# ポイントツーポイント レイヤ 2 サービスの実装に関する情報

ポイントツーポイントレイヤ 2 サービスを実装するには、次の概念を理解する必要があります。

## レイヤ 2 バーチャル プライベート ネットワークの概要

レイヤ 2 バーチャル プライベート ネットワーク（L2VPN）は、IP または MPLS 対応の L2 スイッチド IP ネットワークで LAN の動作をエミュレートすることで、イーサネット デバイス同士が共通の LAN セグメントに接続した場合と同様に通信できるようにします。ポイントツーポイント L2 接続は、L2VPN を作成する場合に重要です。

インターネット サービス プロバイダー（ISP）が、フレーム リレーまたは非同期転送モード（ATM）インフラストラクチャを IP インフラストラクチャに置き換える場合、IP または MPLS 対応の L2 スイッチド IP インフラストラクチャを使用する標準的な方法を提供する必要があります。これらの方法は、カスタマーに実用的な L2 インターフェイスを提供し、具体的には、カスタマー サイトのペア間の仮想回線を提供します。

L2VPN システムを構築するには、ISP とカスタマーの間での調整が必要です。ISP は L2 接続を提供し、カスタマーは ISP から取得したデータ リンク リソースを使用してネットワークを構築します。L2VPN サービスでは、ISP は、カスタマーのネットワーク トポロジ、ポリシー、ルーティング情報、ポイントツーポイントリンクに関する情報や、他の ISP からのネットワーク ポイントツーポイント リンク関する情報を必要としません。

ISP には、次の機能を備えたプロバイダー エッジ（PE）ルータが必要です。

- レイヤ 3（L3）パケット内への L2 プロトコル データ ユニット（PDU）のカプセル化。
- any-to-any L2 転送のインターコネクト。
- パケット スイッチ ネットワーク上での L2 Quality-of-Service（QoS）のエミュレーション。
- L2 サービスの設定の簡素化。
- 各種のトンネリングメカニズム（MPLS、L2TPv3、IPSec、GRE など）のサポート。
- L2VPN プロセス データベースには、回線および接続に関するすべての情報が含まれます。

# レイヤ 2 ローカル スイッチングの概要

ローカル スイッチングにより、同じルータ上の同じタイプの 2 つのインターフェイス間で L2 データを切り替えることができます（たとえば、イーサネットからイーサネット）。インターフェイスは、同じラインカード上にあっても、2 つの異なるラインカード上にあってもかまいません。これらのタイプのスイッチング中、レイヤ 2 アドレスが、レイヤ 3 アドレスの代わりに使用されます。ローカル スイッチング接続は、一方の接続回線（AC）から他方の接続回線に L2 トラフィックを切り替えます。ローカル スイッチング接続で設定される 2 つのポートは、そのローカル接続に関連する AC です。ローカル スイッチング接続の動作は、2 つのブリッジ ポートしかないブリッジ ドメインの動作と類似しており、トラフィックはローカル接続の一方のポートに入り、他方のポートから出ます。ただし、ローカル接続に関するブリッジングがないため、MAC 学習やフラッディングはありません。また、インターフェイスの状態が DOWN の場合、ローカル接続の AC は UP 状態ではありません（この動作は、ブリッジ ドメインの動作に準拠したときにも異なります）。

ローカル スイッチング AC は、L2 トランク（メイン）インターフェイス、バンドル インターフェイス、EFP など、多種多様な L2 インターフェイスを使用します。

また、同一ポートのローカルスイッチング機能を使用すると、同じインターフェイス上の 2 つの回線の間でレイヤ 2 データをスイッチングできます。

# L2VPN での ATMoMPLS の概要

ATMoMPLS は、MPLS コアを介したレイヤ 2 ポイントツーポイント接続の一種です。

ATMoMPLS 機能を実装するために、Cisco ASR 9000 シリーズ ルータはカスタマーエッジ（CE）デバイスが Cisco ASR 9000 シリーズ ルータに接続されているプロバイダーネットワークのエッジでプロバイダーエッジ（PE）ルータの役割を果たします。

# L2VPN での仮想回線接続検証

仮想回線接続性検証（VCCV）は、L2VPN の運用、管理、およびメンテナンス（OAM）機能であり、ネットワーク オペレータが、指定した疑似回線上で IP ベースのプロバイダー エッジ間（PE-to-PE）キープアライブ プロトコルを実行できるようにし、疑似回線データ パス転送で障害が発生しないようにします。ディスポジション PE は、指定した疑似回線に関連付けられる制御チャネルで VCCV パケットを受信します。疑似回線が各方向の PE 間で確立されると、VCCV に使用される制御チャネル タイプと接続検証タイプがネゴシエートされます。

2 つのタイプのパケットが判定結果出力に着信します。

- タイプ 1：通常の Ethernet-over-MPLS（EoMPLS）データ パケットを指定します。
- タイプ 2：VCCV パケットを指定します。

Cisco ASR 9000 シリーズ ルータは、シグナリング中にイネーブルにされた場合にインバンド制御ワードを使用する、ラベル スイッチド パス（LSP）VCCV タイプ 1 をサポートしています。IPv4 では、VCCV エコー応答は、応答モードである IPv4 として送信されます。応答は IP、MPLS、またはその両方の組み合わせとして転送されます。

出力側の MPLS 転送では、VCCV pings カウンタがカウントされます。ただし、入力側では、これらはルート プロセッサから発信され、MPLS 転送カウンタとしてカウントされません。

# Ethernet over MPLS

Ethernet-over-MPLS（EoMPLS）は、MPLS 対応 L3 コアを通じてイーサネット トラフィックのトンネリング メカニズムを提供し、（ラベル スタックを使用して）イーサネット プロトコル データ ユニット（PDU）を MPLS パケット内部にカプセル化して、それらを MPLS ネットワーク経由で転送します。

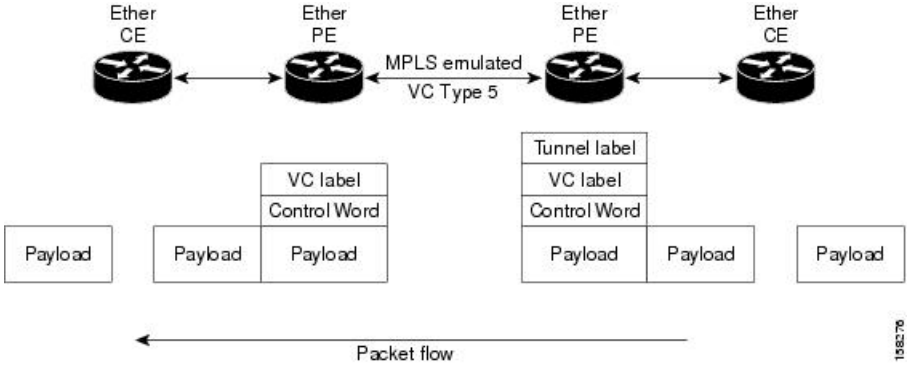
EoMPLS 機能は、次のサブセクションで説明します。

## イーサネット ポート モード

イーサネット ポート モードでは、疑似回線の両端がイーサネット ポートに接続されます。このモードでは、ポートが疑似回線を介してトンネル化されるか、またはローカル スイッチング（接続回線から接続回線へのクロス コネクトと呼ばれる）を使用して、1 つの接続回線（AC）から同じ PE ノードに接続されている別の AC にパケットまたはフレームを切り替えます。

次の図に、イーサネットポートモードの例を示します。

図 *6:* イーサネット ポート モードのパケット フロー

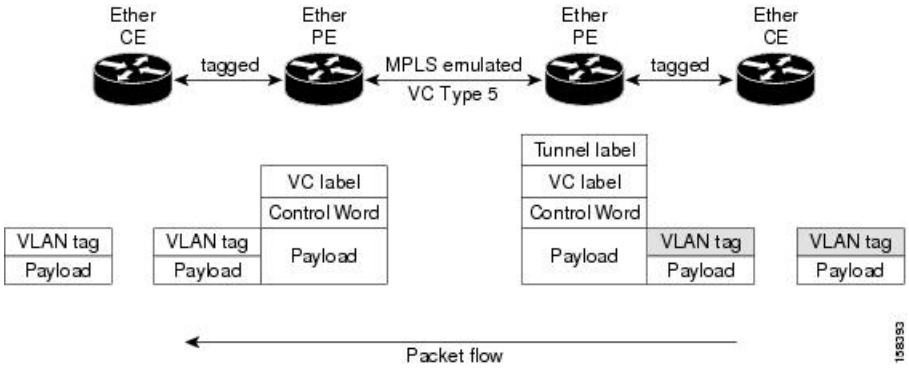



## VLAN モード

VLANモードでは、カスタマー側とプロバイダー側のリンクで、各VLANは、仮想接続（VC）タイプ 4 またはVC タイプ 5 を使用して個別 L2VPN 接続として設定できます。VC タイプ 5 はデフォルトのモードです。

次の図に示されているように、イーサネット PE は、入力ポートから疑似回線にトラフィックを内部的に切り替えるために、イーサネット ポートに内部 VLAN タグを関連付けます。ただし、疑似回線にトラフィックを移動する前に、内部 VLAN タグを削除します。

図 *7: VLAN* モードのパケット フロー




出力 VLAN PE では、PE は、疑似回線から到着するフレームに VLAN タグを関連付け、トラフィックを内部的に切り替えた後、イーサネット トランク ポートにトラフィックを送信します。

（注） ポートがトランク モードであるため、VLAN PE は VLAN タグを削除せず、追加されたタグを持つポート経由でフレームを転送します。

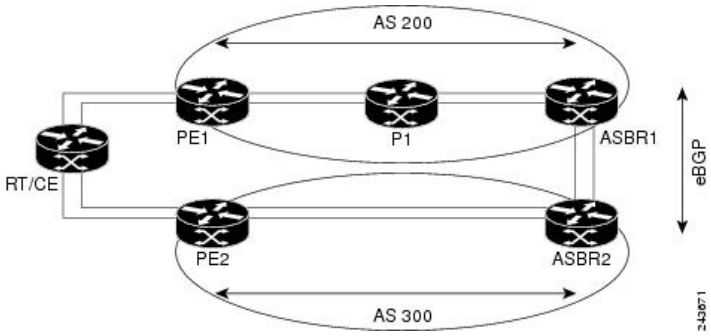
## Inter-AS モード

Inter-AS は、複数のプロバイダーまたはマルチドメイン ネットワークを通じて VPN を拡張できるピアツーピア タイプ モデルです。これにより、サービス プロバイダーは相互にピアアップでき、地理的に離れた位置でエンドツーエンドの VPN 接続が実現します。

EoMPLS サポートでは、単一 AS トポロジを想定でき、このトポロジでは、ポイントツーポイント EoMPLS 相互接続の 2 つの終端にある PE ルータを接続する疑似回線が、同一自律システムに存在します。または、複数の AS トポロジを想定でき、このトポロジでは、PE ルータが iBGP および eBGP ピアリングを使用して 2 つの異なる AS に存在できます。

次の図は、各 AS で iBGP/LDP を使用した基本的な二重 AS トポロジを持つ Inter-AS を介した MPLS を示しています。

**図 *8 : Inter-AS* を介した *EoMPLS*：基本的な二重 *AS* トポロジ**




## QinQ モード

QinQ は、複数の 802.1Q タグ（IEEE 802.1QinQ VLAN タグ スタッキング）を指定するための 802.1Q の拡張です。レイヤ 3 VPN サービス終了および L2VPN サービス転送は、QinQ サブインターフェイスではイネーブルです。

Cisco ASR 9000 シリーズルータは、プロバイダー エッジ ルータでのサブインターフェイスの設定に基づき、レイヤ 2 トンネリングまたはレイヤ 3 転送を実装します。この機能は、SPA および固定 PLIM で最大 2 つの QinQ タグのみサポートします。

- L2VPN 接続回線のレイヤ 2 QinQ VLAN：QinQ L2VPN 接続回線は、仮想回線タイプ 4 とタイプ 5 の両方の疑似回線を使用したポイントツーポイント EoMPLS ベースのクロスコネクト用と、802.1q VLAN およびポートモードでの QinQ の完全なインターワーキングのサポートなど、ポイントツーポイント ローカル スイッチングベースのクロスコネクト用のレイヤ 2 転送サブインターフェイスで設定されます。
- レイヤ 3 QinQ VLAN：レイヤ 3 の終端ポイントとして使用されます。VLAN はいずれも入力プロバイダーエッジで削除され、フレームが転送されるときリモートプロバイダーエッジで追加され戻されます。

QinQ 上のレイヤ 3 サービスは次のとおりです。

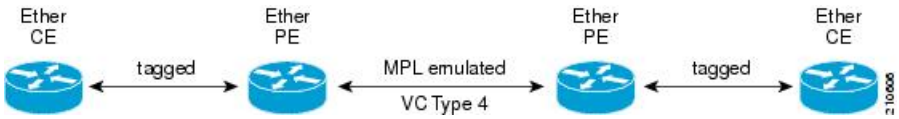
- IPv4 ユニキャストおよびマルチキャスト

- IPv6 ユニキャストおよびマルチキャスト
- MPLS
- Intermediate System-to-Intermediate System（IS-IS）で使用されるコネクションレス型ネットワーク サービス（CLNS）

QinQ モードでは、各 CE VLAN は SP VLAN 内に伝送されます。QinQ モードでは VC タイプ 5 を使用する必要がありますが、VC タイプ 4 もサポートされます。各イーサネット PE では、内部（CE VLAN）と外部（SP VLAN）の両方を設定する必要があります。

次の図に、VC タイプ 4 を使用した QinQ を示します。

**図 *9 : QinQ* を介した *EoMPLS* モード**




## QinAny モード

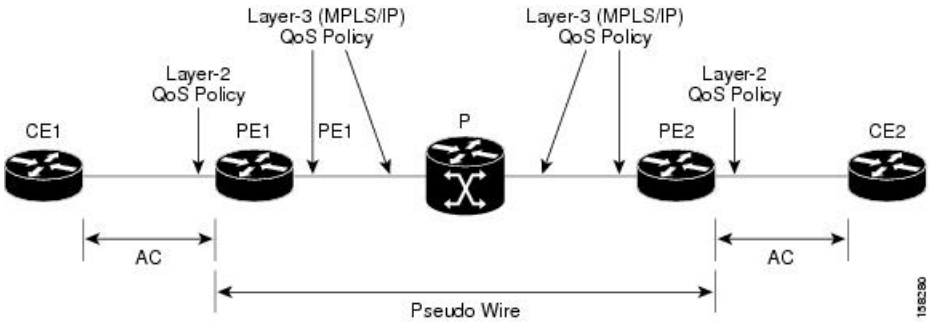
QinAny モードでは、サービス プロバイダー VLAN タグは、プロバイダー エッジ VLAN の入力ノードと出力ノードの両方で設定されます。カスタマー エッジ VLAN タグが不明なため、カスタマー エッジ VLAN タグが疑似回線上のパケットで送信されることを除き、QinAny モードはタイプ 5 VC を使用する Q-in-Q モードに似ています。

# QoS

L2VPN テクノロジーを使用して、ポートおよび VLAN の動作モードの両方に Quality of Service（QoS）レベルを割り当てることができます。

L2VPN テクノロジーでは、PE ルータの QoS 機能が、エッジ方向のインターフェイス（別名、接続回線）で L2 ペイロードベースである必要があります。次の図は、一般的な L2VPN ネットワークでの L2 および L3 QoS サービスポリシーを表しています。

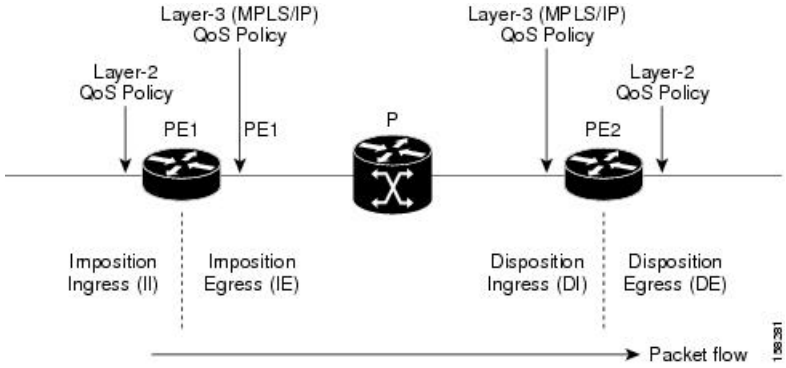
**図 *10 : L2VPN QoS* 機能の適用**




次の図は、QoS サービスポリシーを設定できるプロバイダーエッジデバイス内の 4 つのパケット処理パスを表しています。L2VPN ネットワークでは、パケットはエッジ方向のインターフェ

イスで L2 パケットとして送受信され、コア方向のインターフェイスで MPLS（EoMPLS）パケットとして転送されます。

**図 *11 : L2VPN QoS* リファレンス モデル**




# ハイ アベイラビリティ

L2VPN は、ルート プロセッサとラインカードの両方でコントロール プレーンを使用し、ラインカードでフォワーディング プレーン要素を使用します。

L2VPN の可用性は次の要件を満たします。

- ルート プロセッサまたはラインカードでのコントロール プレーンの障害は、回線の転送パスには影響しません。
- ルータ プロセッサのコントロール プレーンは、ラインカードの制御およびフォワーディング プレーンに影響を与えずに、フェールオーバーをサポートします。
- L2VPN は既存のラベル配布プロトコル（LDP）のグレースフル リスタート メカニズムと統合されます。

# 優先トンネル パス

優先トンネル パスの機能により、特定のトラフィック エンジニアリング トンネルに疑似回線をマッピングできます。接続回線は、リモート PE ルータの IP アドレス（IGP または LDP を使用して到達可能）ではなく、特定の MPLS トラフィック エンジニアリング トンネル インターフェイスに相互接続されます。優先トンネル パスを使用する場合、L2 トラフィックを転送するトラフィック エンジニアリング トンネルが 2 台の PE ルータ間で動作することが常に想定されます（つまり、始端はインポジション PE ルータで、終端はディスポジション PE ルータです）。

（注）
- 現在、優先トンネル パス設定は MPLS カプセル化だけに適用されます。

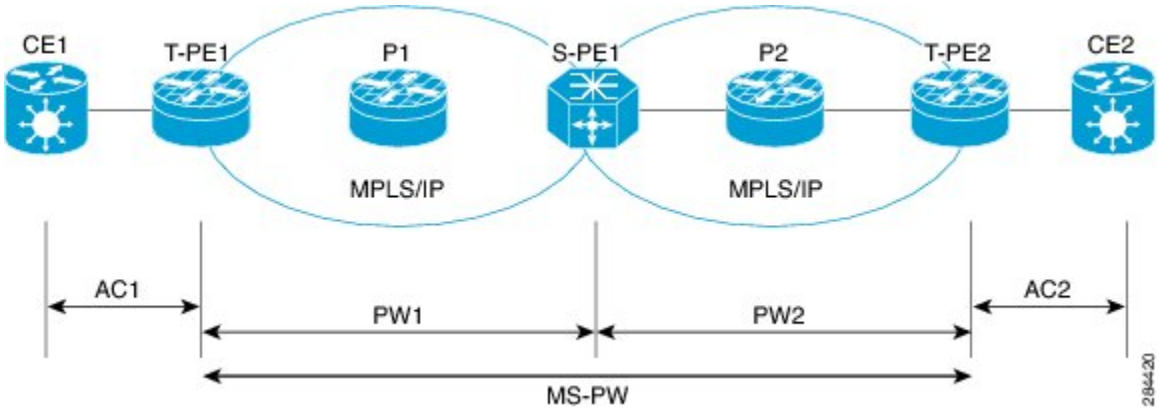
# マルチセグメント疑似回線

疑似回線は Public Switched Network（PSN）上でレイヤ 2 プロトコル データ ユニット（PDU）を転送します。マルチセグメント疑似回線は、静的または動的に設定された、複数の隣接する疑似回線セグメントのセットです。これらのセグメントは単一の疑似回線として機能し、以下を実行できます。

- 管理ドメインまたはプロビジョニング ドメインを隔離することで、エンドツーエンド サービスを管理する。
- 相互自律システム（Inter-AS）の境界を越えて、プロバイダー エッジ（PE）ノードの IP アドレスをプライベートにする。自律システム境界ルータ（ASBR）の IP アドレスを使用し、それらのルータを疑似回線の集約ルータとして扱う。ASBR は、2 つのドメインの擬似回線を結合します。

マルチセグメント疑似回線は、Inter-AS 境界または 2 つのマルチプロトコル ラベル スイッチング（MPLS）ネットワークにまたがることができます。

**図 *12*：マルチセグメント疑似回線：例**




疑似回線は、2 台の PE ノード間のトンネルです。2 種類の PE ノードがあります。

- スイッチング PE（S-PE）ノード
  - マルチセグメント疑似回線の先行する疑似回線セグメントと後続の疑似回線セグメントの PSN トンネルを終端させます。
  - マルチセグメント疑似回線の先行する疑似回線セグメントと後続の疑似回線セグメントのコントロール プレーンとデータ プレーンを切り替えます。
- 終端 PE（T-PE）ノード
  - マルチセグメント疑似回線の最初と最後の両方のセグメントに配置されます。
  - このノードで、カスタマー方向の接続回線（AC）が疑似回線フォワーダにバインドされます。

（注） すべてのマルチセグメント擬似回線は、T-PE で終端する必要があります。

マルチセグメント擬似回線は、次の場合に 2 つの一般的なケースで使用されます。

- 送信元と宛先の PE ノード間で PW 制御チャネルを確立することができない場合。

  PW コントロールチャネルを確立するには、リモート PE ノードがアクセス可能である必要があります。場合によっては、トポロジ、動作、またはセキュリティ上の制約により、ローカル PE ノードがリモートノードにアクセスできない場合があります。

  マルチセグメントの擬似回線は、2つの独立した疑似回線セグメントを動的に構築し、擬似回線スイッチングを実行して、送信元と宛先の PE ノード間の PW 制御チャネルを確立します。

- エッジ間の擬似回線エミュレーション（PWE3）のシグナリングとカプセル化プロトコルが異なる場合。

  PE ノードの接続先のネットワークでは、異なる PW シグナリングおよびカプセル化プロトコルが使用されています。場合によっては、1 つのセグメント PW を使用できません。

  マルチセグメント擬似回線はPW スイッチングポイントで適切なインターワーキングが実行されており、ネットワーク内の PE ノード間で PW 接続を有効にします。

# 疑似回線冗長性

疑似回線冗長性を使用すると、ネットワーク内の障害を検出して、サービスの提供を続行可能な別のエンドポイントにレイヤ2サービスを再ルーティングするようにネットワークを設定できます。この機能により、リモート PE ルータで発生した障害、または PE ルータと CE ルータ間のリンクで発生した障害から回復できます。

L2VPN は、ルーティング プロトコルを通じて疑似回線冗長化機能を提供します。エンドツーエンド PE ルータ間の接続が障害になった場合、指示された LDP セッションとユーザ データの代替パスに引き継ぐことができます。ただし、ネットワークの一部は、この再ルーティング メカニズムでサービスの中断から保護されません。

疑似回線冗長性を使用すると、バックアップ疑似回線を設定できます。ネットワークに冗長疑似回線と冗長ネットワーク エレメントを設定することもできます。

プライマリ疑似回線の障害前に、バックアップ疑似回線にトラフィックをスイッチングする機能が使用され、ルータのメンテナンスなどの計画された疑似回線の停止が処理されます。

（注） 疑似回線冗長性は、ポイントツーポイントの Virtual Private Wire Service（VPWS）疑似回線に対してのみ提供されます。

# 疑似回線のロード バランシング

冗長性を維持しつつ、ネットワークを最大限利用するには、通常、複数のリンクでのトラフィックのロード バランシングが必要です。精度の高い、より均等な分散を実現するには、プロビジョニングされたパイプの一部であるトラフィック フローのロード バランシングが理想的です。ロードバランシングは、IPアドレス、Macアドレス、またはそれらの組み合わせに従い、フローベースにすることができます。またロード バランシングは、送信元または宛先の IP アドレス、あるいは送信元または宛先の MAC アドレスに従い、フローベースにすることができます。IP ヘッダーの処理に進むことができない場合、または IPv6 がフローベースの場合、トラフィックはデフォルトのフローベース MAC アドレスにフォールバックします。

この機能は、L2VPN 下の疑似回線に適用されます。これには、VPWS と VPLS が含まれます。

（注） 疑似回線クラスに対し仮想回線（VC）ラベル ベースのロード バランシングをイネーブルにすると、L2VPN 下のグローバル フロー ベースのロード バランシングが上書きされます。

# 疑似回線のグループ化

疑似回線（PW）が確立されると、各 PW に、すべての PW に共通するグループ ID が割り当てられます。このグループ ID は、同一の物理ポートで作成されます。物理ポートが機能しなくなった場合や無効になった場合は、自動保護スイッチング（APS）がピアルータに対してアクティブになるように信号を送り、L2VPN が単一のメッセージを送信して、物理ポートに関連付けられたグループ ID を持つすべての PW のステータス変更をアドバタイズします。単一の L2VPN 信号であることにより、応答での煩雑な処理や切断を防ぐことができます。

CEM インターフェイスでは、フレーム化または非フレーム化 T1 および T3 などの親コントローラに対して、さまざまなレベルの設定が許可されます。最適なグループ化を行うために、物理コントローラのハンドルがグループ ID として使用されます。

（注） 疑似回線のグループ化はデフォルトでディセーブルです。

疑似回線のネットワーク コンバージェンスには、次のようなイベントでは通常の 2 秒よりも長くかかる場合があります。

- アクティブな動作ルータの手動リロード
- インターフェイスまたはコントローラのシャットダウン
- 有効な保護ルータでのラインカードのリロード、シャットダウン、または電源遮断
- 有効な保護ルータでのルータ プロセッサ フェールオーバー（RPFO）
- 2 つのコントローラまたは共有ポートアダプタ（SPA）の同時障害
- 2 つの自動保護スイッチング（APS）グループスイッチオーバー

# イーサネットワイヤサービス

イーサネットワイヤサービスは、ポイントツーポイントのイーサネットセグメントをエミュレートするサービスです。これは、プロバイダーエッジがレイヤ 2 で動作し、通常レイヤ 2 ネットワークで実行される以外、イーサネット専用回線（EPL）、レイヤ 1 ポイントツーポイント サービスに似ています。EWS は特定の UNI で受信されたすべてのフレームをカプセル化し、フレームに含まれる内容を参照せずに、これらのフレームを単一出力UNIに転送します。このサービスの動作は EWS を VLAN タグ付きフレームで使用できることを示します。VLAN タグは、一部の例外を除いて EWS（ブリッジ プロトコル データ ユニット（BPDU））に対して透過的です。これらの例外には、IEEE 802.1x、IEEE 802.2ad、および IEEE 802.3x が含まれます。これは、これらのフレームがローカルで意味を持ち、カスタマーとサービスプロバイダーの両方がそれらのフレームをローカルで終了できるよう支援されるためです。

サービスプロバイダーはインターフェイスでフレームを単純に受け取り、実際のフレームを参照せずにこれらを送信するため（ただし、形式と長さが特定のインターフェイスに適合していることは確認します）、EWS はカスタマーのイーサネットフレーム内にある VLAN タグに関与しません。

EWS は all-to-one バンドリングの概念に対応しています。つまり、EWS はポイントツーポイント回線の一方のエンドのポートと他方のエンドのポートをマッピングします。EWS はポート間サービスです。したがって、カスタマーが 1 つのスイッチまたはルータを n 個のスイッチまたはルータに接続する必要がある場合は、n 個のポートおよび n 個の疑似回線または論理回線が必要になります。

考慮すべき 1 つの重要なポイントは、EWS はイーサネットレイヤ 1 接続を広範にエミュレートするにもかかわらず、サービスは共有インフラストラクチャで提供され、したがって、すべてのインターフェイス帯域幅を常に使用できる可能性は低く、またそのようにする必要もないということです。EWS は、通常、多くのユーザが伝送パスのどこかで回線を共有する、サブラインレートサービスです。その結果、コストが EPL のコストよりも、ほとんどの場合、小さくなります。SP は、レイヤ 1 EPL とは異なり、特定契約の特定目的を達成するために、QoS およびトラフィック エンジニアリングを実装する必要があります。ただし、カスタマーアプリケーションに本当の意味でのワイヤレート透過サービスが必要な場合、DWDM（高密度波長分割多重）、CDWM（低密度波長分割多重）、SONET/SDH などの光送信デバイスを使用して提供される EPL サービスを検討する必要があります。

## IGMP スヌーピング

IGMP スヌーピングは、レイヤ 2 でマルチキャスト トラフィックを抑制する方法を提供します。IGMP スヌーピング アプリケーションは、ブリッジドメインのホストによって送信された IGMP メンバーシップレポートをスヌーピングすることで、レイヤ 2 マルチキャスト転送テーブルを設定して、少なくとも 1 つの関係メンバーを持つポートだけにトラフィックを送信できます。これにより、マルチキャストトラフィックの量が大幅に削減されます。

IGMP は、レイヤ 3 で設定され、IPv4 マルチキャストネットワーク内のホストが、関与するマルチキャストトラフィックを通知する手段、ルータがレイヤ 3 のネットワーク内のマルチキャストトラフィックのフローを制御および制限する手段を提供します。

IGMP スヌーピングは、IGMP メンバーシップ レポート メッセージの情報を使用して、対応する情報を転送テーブルに構築し、レイヤ 2 の IP マルチキャスト トラフィックを制限します。転送テーブルのエントリは <ルート, OIF リスト> という形式で、

- ルートは <*, G> ルートまたは <S, G> ルートです。
- OIF リストは、指定されたルートと、ブリッジ ドメイン内のすべてのマルチキャスト ルータ（mrouter）ポートに関する IGMP メンバーシップ レポートを送信したすべてのブリッジ ポートで構成されます。

IGMP スヌーピング機能により、マルチキャスト ネットワークで次の利点が得られます。

- 基本的な IGMP スヌーピングは、VPLS ブリッジ ドメイン全体をフラッディングするマルチキャスト トラフィックを削減することで、帯域幅の使用量を減らします。
- オプションの設定オプションを使用すると、IGMP スヌーピングは、1 つのブリッジ ポートでホストから受信された IGMP レポートをフィルタリングし、他のブリッジ ポートでホストへの漏出を防止することで、ブリッジ ドメイン間のセキュリティを確保できます。
- オプションの設定オプションを使用すると、IGMP スヌーピングは、IGMP メンバーシップ レポート（IGMPv2）を抑制することで、またはアップストリーム IP マルチキャスト ルータへの IGMP プロキシ レポーター（IGMPv3）として動作することで、アップストリーム IP マルチキャスト ルータへのトラフィックの影響を低減できます。

IGMP スヌーピングの設定方法については、『Cisco ASR 9000 Series Aggregation Services Router Multicast Configuration Guide』の「Implementing Layer 2 Multicast with IGMP Snooping」モジュールを参照してください。

適用できる IGMP スヌーピング コマンドは『Cisco ASR 9000 Series Aggregation Services Router Multicast Command Reference』で説明します。

# IP インターワーキング

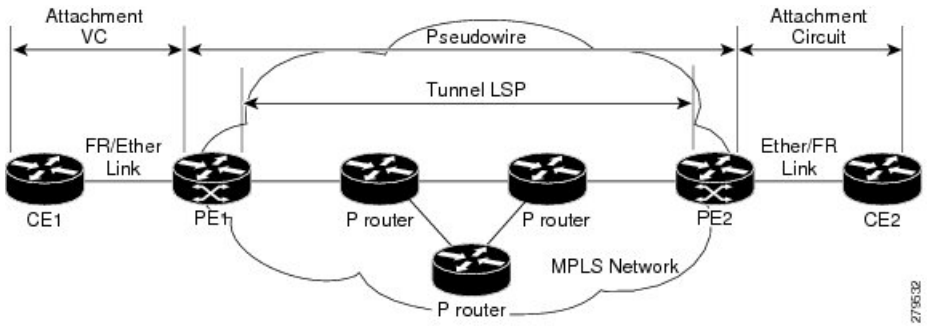
カスタマー環境では、ソリューションによりネットワーク終端で異種転送を使用する AToM をサポートする必要があります。このソリューションには、1 つのカスタマー エッジ（CE）デバイスの転送を別の転送に変換する機能（たとえば、フレーム リレーからイーサネットなど）が必要です。Cisco ASR 9000 シリーズ SPA インターフェイス プロセッサ 700 および Cisco ASR 9000 シリーズ イーサネット ライン カードにより、Cisco ASR 9000 シリーズ ルータで複数のレガシー サービスをサポートできます。

IP インターワーキングは、IP/MPLS バックボーン上でレイヤ 2 トラフィックを転送するためのソリューションです。IP インターワーキングは、AToM トンネルを使用するイーサネット、フレーム リレーなど、多くのタイプのレイヤ 2 フレームに対応します。IP インターワーキングは、プロバイダー エッジ（PE）ルータでパケットをカプセル化し、それらをバックボーンを介してクラウドの反対側の PE ルータに転送し、カプセル化を削除し、それらを宛先に転送します。トランスポート層では、一方の側でイーサネットを使用し、もう一方の側でフレーム リレーを使用できます。IP インターワーキングは、AToM トンネルの異種エンドポイント間で実行されます。

（注） MPLS とローカル接続のシナリオでは、イーサネットとフレーム リレー ベースのネットワーク間でルーテッド インターワーキングのみサポートされます。

次の図は、イーサネット接続 VC とフレームリレー接続 VC 間の相互運用性を表しています。

**図 *13 : MPLS* コア上の *IP* インターワーキング**




接続回線は（AC）は、CE デバイスを PE デバイスに接続する物理的または論理的なポートまたは回線です。疑似回線（PW）は、2 つの AC を接続する双方向仮想接続（VC）です。MPLS ネットワークでは、PW は LSP トンネル内で伝送されます。PE1 および PE2 のコア方向のラインカードとして、Cisco ASR 9000 シリーズ SPA インターフェイス プロセッサ 700 または Cisco ASR 9000 シリーズ イーサネット ラインカードが使用可能です。

IP インターワーキング モードでは、入力 PE で受信されたパケットからレイヤ 2（L2）ヘッダーが削除され、IP ペイロードだけが出力 PE に送信されます。出力 PE では、パケットが出力ポートから送信される前に、L2 ヘッダーが付加されます。

上の図では、CE1 および CE2 を、フレームリレー（FR）インターフェイスまたはギガビットイーサネット（GigE）インターフェイスにすることができます。CE1 が FR で、CE2 が GigE または dot1q、あるいは QinQ であるとします。イーサネット CE（CE2）から着信するパケットの場合、CE 方向の PE（PE2）の入力 LC は、L2 フレーミングを削除し、そのパケットを、疑似回線上で IPoMPLS カプセル化を使用して出力 PE（PE1）に転送します。出力 PE のコア方向のラインカードは、MPLS ラベルを削除しますが、制御ワードを保持し、それを FR CE（CE1）方向の出力ラインカードに伝送します。FR PE では、ラベル ディスポジション後、レイヤ 3（L3）パケットは FR 上でカプセル化されます。

同様に、FR CE から着信した IP パケットは疑似回線上で IPoMPLS カプセル化に変換されます。コアから着信するパケットは IP ペイロードのみを伝送するため、イーサネット PE 側では、ラベル ディスポジション後、PE は、パケットを CE に伝送する前に、そのパケットに L2 イーサネット パケット ヘッダーを追加して戻します。

これらのモードは、AToM で IP インターワーキングをサポートします。

- イーサネットとフレーム リレー

  イーサネット CE デバイスから着信するパケットには、MAC（ポート モード、タグなし、シングル タグ、ダブル タグ）、IPv4 ヘッダー、およびデータが含まれます。イーサネット ラインカードは L2 フレーミングを削除し、その後、出力ラインカードに L3 パケット

を転送します。出力ラインカードは、出力ポートからパケットを送信する前に、FR L2 ヘッダーを追加します。

- イーサネットとイーサネット

  CE デバイスは両方ともイーサネットです。各イーサネット インターフェイスは、ポート モード、タグなし、シングル タグ、またはダブル タグにすることができます。ただし、これは IP インターワーキングの一般的なシナリオではありません。

## AToM iMSG

この機能により、アクセス ネットワーク内のインターワーキング レイヤですべての非イーサネット機能を終了し、これらの接続を、レイヤ 3 エッジ ルータで終端可能なイーサネット セントリック サービスに変換することができます。現在は、時分割多重（TDM）ベースのサービスはレイヤ 3 エッジ ルータ上で直接終端しています。L3 ネットワークの簡素でより低コストなモデルは、TDM の複雑さをアクセス レイヤに移動することによってイネーブルになります。

レイヤ 2 カプセル化は、入力ラインカード側の入力 PE の接続回線によって IP パケットから削除されます。MPLS カプセル化された IP パケットのペイロードは、ファブリックで出力ラインカード側のコアに送信されます。出力ラインカードは MPLS コアを介してパケットを送信します。リモート PE では、MPLS ラベルが削除され、出力 AC のレイヤ 2 ヘッダーが追加されて、パケットは最終的に接続された CE に送信されます。L2VPN VPWS は、次をサポートするように拡張されました。

- ポイントツーポイント プロトコル（PPP）
- ハイレベル データリンク コントロール（HDLC）
- マルチリンク ポイントツーポイント プロトコル（MLPPP）
- すべてのカプセル化タイプの QoS サポート

  QoS の詳細については、『Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Configuration』を参照してください。

TDM AC は、次の SPA で設定できます。

- SPA-8XCHT1/E1
- SPA-4XCT3/DS0
- SPA-1XCHSTM1/OC3
- SPA-2XCHOC12/DS0
- SPA-1XCHOC48/DS3
- SPA-4XT3/E3
- SPA-4XOC3-POS-V2
- SPA-8XOC3-POS
- SPA-8XOC12-POS

- SPA-1XOC48POS/RPR
- SPA-2XOC48POS/RPR

# Any Transport over MPLS

Any Transport over MPLS（AToM）は、マルチプロトコル ラベル スイッチング（MPLS）バックボーン上でレイヤ2パケットを転送します。これにより、サービスプロバイダーは、単一の統合されたパケット ベース ネットワーク インフラストラクチャを使用することで、既存のレイヤ 2 ネットワークとカスタマー サイトを接続できます。この機能を使用すると、サービスプロバイダーは、別々のネットワークを使用する代わりに、MPLS バックボーン上でレイヤ 2 接続を提供できます。

AToM は、入力 PE ルータでレイヤ 2 フレームをカプセル化し、2 つの PE ルータ間を接続する疑似回線の反対側に位置する対応した PE ルータにそれらを送信します。出力 PE はカプセル化を削除し、レイヤ 2 フレームを送信します。

PE ルータ間でレイヤ 2 フレームを正常に転送するには、PE ルータを設定する必要があります。ルータ間で、疑似回線と呼ばれる接続を設定します。各 PE ルータで次の情報を指定します。

- イーサネットやフレームリレーなどの疑似回線で転送されるレイヤ 2 データのタイプ。
- PE ルータが通信できる、ピア PE ルータのループバック インターフェイスの IP アドレス。
- 疑似回線を識別するピア PE の IP アドレスと VC ID の一意の組み合わせ。

## コントロールワード処理

フレームリレー接続の場合、コントロールワードには、順方向明示的輻輳通知（FECN）、逆方向明示的輻輳通知（BECN）、および DE ビットが含まれます。

コントロールワードは次で必須です。

- フレーム リレー
- ATM AAL5
- Frame Relay to Ethernet ブリッジ型インターワーキング
- cHDLC/PPP IP インターワーキング
- CEM（回線エミュレーション）

システムは、AToM IP インターワーキング接続を介して、転送エンドポイントから別のエンドポイントにビットをマッピングしません。

コントロールワードがサポートされている場合も、疑似回線のために常に使用することをお勧めします。これは、L2VPN パケットの内容とは関係なく、パケットのデシーケンシングなしで適切なロードバランシングを実行できるためです。コントロールワードがない場合、ロード

バランシングを実行するために使用されるヒューリスティックでは、どのケースでも最適な結果を達成できません。

### High-Level Data Link Control over MPLS

接続回線（AC）は、HDLCカプセル化が設定されたメインインターフェイスです。ACとの間のパケットは、MPLSコア ネットワーク上の他のプロバイダーエッジ（PE）との間の、VCタイプ 0x6 の疑似回線（PW）を使用して転送されます。

HDLC over MPLS では、HDLC パケット全体が転送されます。入力 PE ルータは、HDLC フラグおよび FCS ビットだけを削除します。

### PPP over MPLS

接続回線（AC）は、PPPカプセル化が設定されたメインインターフェイスです。ACとの間で送受信されるパケットは、MPLS コア ネットワーク上の他のプロバイダー エッジ（PE）との間で、VC タイプ 0x7 の AToM PW を介して転送されます。

PPP over MPLSの場合、入力PEルータはフラグ、アドレス、制御フィールド、およびFCSビットを削除します。

## Frame Relay over MPLS

Frame Relay over MPLS（FRoMPLS）は、2 つのフレーム リレー アイランド間の専用回線タイプの接続を提供します。フレーム リレー トラフィックは MPLS ネットワーク上で転送されます。

（注） データリンク接続識別子（DLCI）の DCLI-DLCI モードがサポートされます。追加の制御情報を伝えるために、制御ワード（DLCI-DLCI モードに必要）が使用されます。

プロバイダーエッジ（PE）ルータは、加入者サイトからフレーム リレー プロトコル パケットを受信すると、フレーム リレー ヘッダーおよびフレーム チェック シーケンス（FCS）を削除し、関連する仮想回線（VC）ラベルを付けます。削除された逆方向明示的輻輳通知（BECN）、順方向明示的輻輳通知（FECN）、廃棄適性（DE）、およびコマンド/応答（C/R）ビットが制御ワードを使用して個別に送信されます（DLCI-DLCI モードの場合）。

## MPLS トランスポート プロファイル

MPLS トランスポート プロファイル（MPLS-TP）トンネルは、IP および MPLS トラフィックが通過する転送ネットワーク サービス レイヤを提供します。MPLS-TP 環境内では、疑似回線（PW）は MPLS-TP トンネルを転送メカニズムとして使用します。MPLS-TP トンネルは、SONET/SDH TDM テクノロジーからパケット スイッチングへの移行に役立つとともに、サービスの高帯域幅での使用と低コスト化をサポートします。転送ネットワークは、接続指向型で静的にプロビジョニングされ、寿命の長い接続を持ちます。通常、転送ネットワークは、ラベ

ルなどの ID を変更する制御プロトコルを回避します。MPLS-TP トンネルは、静的にプロビジョニングされた双方向ラベル スイッチド パス（LSP）を介してこの機能を提供します。

MPLS トランスポートプロファイルの設定方法の詳細については、『Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide』を参照してください。

MPLS-TP は、次のスタティックおよびダイナミックなマルチセグメント疑似回線の組み合わせをサポートします。

- スタティック - スタティック
- スタティック - ダイナミック
- ダイナミック - スタティック
- ダイナミック - ダイナミック

MPLS-TP は、次のスタティックおよびダイナミック疑似回線の組み合わせで 1 対 1 L2VPN 疑似回線冗長性をサポートします。

- スタティック疑似回線とスタティック バックアップ疑似回線
- スタティック疑似回線とダイナミック バックアップ疑似回線
- ダイナミック疑似回線とスタティック バックアップ疑似回線
- ダイナミック疑似回線とダイナミック バックアップ疑似回線

既存の TE 優先パス機能は、PW を MPLS-TP 転送トンネルにピン ダウンするために使用します。優先トンネルパスの設定の詳細については、「優先トンネル パス」を参照してください。ダイナミック疑似回線では、PW ステータスは LDP によって交換されますが、スタティック PW では、ステータスは PW OAM メッセージに転送されます。PW ステータス OAM の設定の詳細については、「PW ステータス OAM の設定」を参照してください。デフォルトでは、PW を伝送する MPLS TP トンネルのステートの変化によって PW のステートが変化する場合、アラームは生成されません。

# Circuit Emulation Over Packet Switched Network

Circuit Emulation over Packet（CEoP）は、パケット スイッチド ネットワークで TDM 回線を伝送する方法です。CEoP は物理接続に似ています。CEoP の目標は、専用回線およびレガシー TDM ネットワークを置き換えることです。

CEoP は主に次の 2 つのモードで動作します。

- SAToP（Structure Agnostic TDM over Packet）と呼ばれる非構造化モード

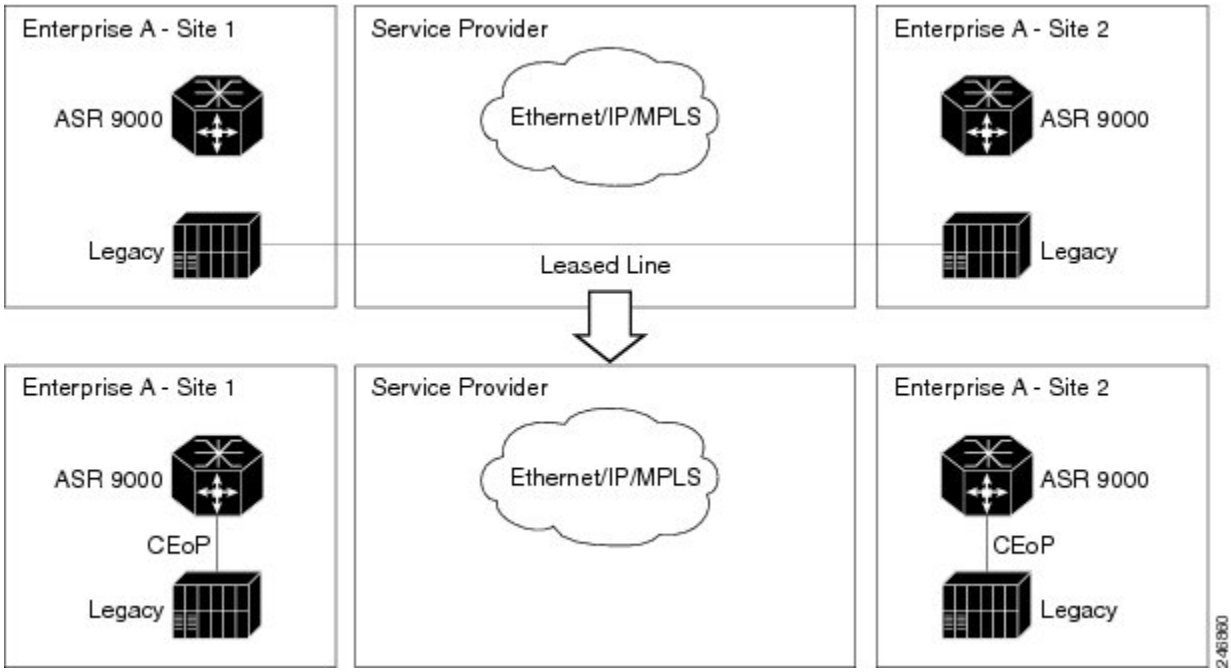
  SAToP は、非フレーム化 E1、T1、E3 および T3 などの構造にとらわれない転送だけをアドレス指定します。これにより、すべての TDM サービスはビット ストリームに分割され、PW トンネルでの送信用にカプセル化されます。このプロトコルは、TDM トラフィック データおよび同期タイミング情報を透過的に送信できます。SAToP は完全に構造を無視するため、プロバイダー エッジ（PE）ルータは、TDM データを解釈したり TDM シグ

ナリングに参加したりする必要がありません。このプロトコルは PDH ビットストリームを透過的に送信するための簡単な方法です。

- CESoPSN（Circuit Emulation Service over Packet Switched Network）という名前の構造化モード

  SAToP と違い、CESoPSN は、エミュレートされた構造化 TDM 信号を送信します。つまり、TDM フレームのフレーム構造を識別して処理し、シグナリングを送信できます。これはアイドル タイムスロット チャネルを送信しない場合がありますが、E1 トラフィック ストリームから CE デバイスの有用なタイムスロットのみを抽出し、伝送用に PW パケットにカプセル化します。CEoP SPA は、ハーフハイト（HH）の共有ポートアダプタ（SPA）です。CEoP SPA ファミリは、非構造化/構造化（NxDS0）クォータ レート、ハーフ ハイト SPA である 24xT1/E1、2xT3/E3、および 1xOC3/STM1 で構成されます。

**図 *14 : Circuit Emulation over Packet* を使用した企業データのコンバージェンス**




CEM 機能は、CEoP SPA を持つ Engine 5 ラインカードでのみサポートされています。CEM は、次でサポートされています。

- 1 ポート チャネライズド OC3 STM1 ATM CEoP SPA（SPA-1CHOC3-CE-ATM）

  CESoPSN および SAToP は、基礎となる転送メカニズムとして MPLS、UDP/IP、および L2TPv3 を使用できます。このリリースでは、MPLS 転送メカニズムだけをサポートしています。

CEoP SPA は次の動作モードをサポートしています。

- 回線エミュレーション モード（CEM）
- ATM モード

- IMA モード

（注） サポートされるのは CEM モードだけです。

#### Circuit Emulation over Packet Switched Network の利点

CEM はサービスプロバイダーとエンドユーザに次の利点を提供します。

- 機器の設置のコスト削減します。
- ネットワーク運用のコストを削減します。高価な専用回線で、コストを節約するモードだけにアクセスを制限する必要がなくなります。
- メンテナンスが必要なのはコア ネットワークだけのため、メンテナンス コストを抑制できます。
- 投資をアクセス ネットワーク全体にとどめたまま、パケット スイッチド ネットワークでコア ネットワークのリソースをより効率的に利用できます。
- エンド ユーザにより安価なサービスを提供できます。

## L2VPN ノンストップ ルーティング

L2VPN ノンストップルーティング（NSR）機能により、プロセス障害（クラッシュ）やルートプロセッサ フェールオーバー（RP FO）などの、イベントのフラッピングによるラベル配布パス（LDP）セッションを回避できます。NSR プロセス障害スイッチオーバーを使用して NSR をイネーブルにした場合、RP FO を実行することによって、プロセス障害（クラッシュ）での NSR がサポートされます。

NSR は、障害が発生したルータについて、グレースフル リスタート（GR）なしでコントロール プレーン ステートを維持できます。NSR は、定義上、プロトコル拡張の必要がないため、通常はステートフル スイッチ オーバー（SSO）を使用してコントロール プレーン ステートを維持します。

（注） NSR は、Cisco IOS XR 64 ビット オペレーティング システムの L2VPN ではデフォルトで有効になっています。L2VPN コンフィギュレーション サブモードでは **nsr** コマンドを設定できません。

## L2TPv3 over IPv6

L2TPv3 over IPv6 トンネルは、L2TPv3（レイヤ 2 トンネリング プロトコル バージョン 3）over IPv6 を使用する静的 L2VPN クロスコネクトであり、クロスコネクトごとに一意の IPv6 送信元アドレスを持ちます。L2TPv3 over IPv6 トンネルは、サブスクライバ VLAN ごとに 1 つの

L2TPv3 トンネルで構成されます。一意の IPv6 アドレスにより、顧客と配信されるサービスを完全に識別できます。

（注） L2TPv3 over IPv6 トンネルは、ASR 9000 拡張イーサネットラインカードで、ルータおよびラインカードごとに 15000 クロスコネクトの規模でサポートされています。

（注） nV サテライト アクセス インターフェイスは、L2TPv3 over IPv6 をサポートしていません。

## 概要

L2TPv3 は、レイヤ 2 バーチャル プライベート ネットワーク（VPN）を使用して、IP コアネットワークを介して、レイヤ 2 ペイロードをトンネリングするための L2TP プロトコルを定義します。2 つの顧客のネットワークサイト間のトラフィックが、L2TP データメッセージ（ペイロード）を伝送する IP パケット内にカプセル化され、IP ネットワーク経由で送信されます。IP ネットワークのバックボーンルータは、他の IP トラフィックの処理方法と同じ方法で、このペイロードを処理します。L2TPv3 over IPv6 を実装すると、一意の送信元 IPv6 アドレスを利用してイーサネット接続回線を直接識別することができます。この場合、L2TPv3 セッション ID の処理はバイパスされます。これは、各トンネルに関連付けられるセッションが 1 つだけであるためです。ただし、このローカル最適化は、同じルータ上の他の L2TPv3 トンネルのセッション ID を通じて回線の多重化を引き続きサポートする能力の妨げにはなりません。

詳細については、次を参照してください。

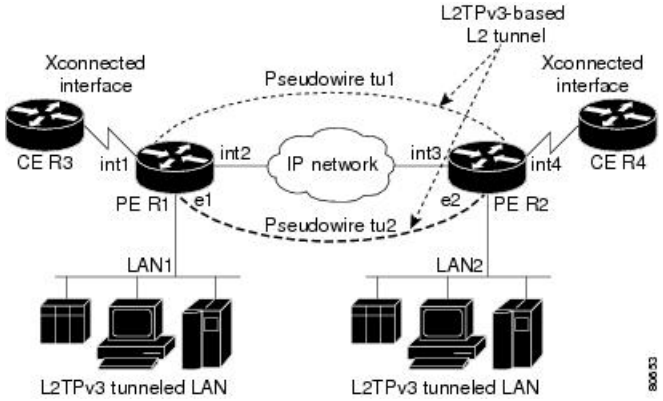
- 設定手順については、「L2TPv3 over IPv6 トンネルの設定」を参照してください。
- 設定例については、「L2TPv3 over IPv6 トンネルの設定：例」を参照してください。

# L2TPv3 over IPv4

L2TPv3（レイヤ 2 トンネリング プロトコル バージョン 3）over IPv4 は、L2TPv3 セッション ID を回線識別子として使用して、複数の接続回線が 1 組の IP アドレスエンドポイントで多重化される、パケット指向のデータネットワークを介してレイヤ 2（L2）回線をトンネリングする動的メカニズムを提供します。

下の図に、IP ネットワーク上のレイヤ 2 トンネリングを使用して VPN をセットアップするための L2TPv3 機能の使用方法を示します。2 つのカスタマー ネットワーク サイト間のすべてのトラフィックが、L2TP データ メッセージを伝送する IP パケット内にカプセル化され、IP ネットワーク経由で送信されます。IP ネットワークのバックボーンルータは、そのトラフィックを他の IP トラフィックとして処理し、顧客のネットワークのことを何も知る必要がありません。

図 *15 : L2TPv3* の動作




上の図では、PE ルータ R1 と R2 が L2TPv3 サービスを提供しています。R1 ルータと R2 ルータは、インターフェイスの int1 と int2、IP ネットワーク、およびインターフェイスの int3 と int4 を構成するパスを通る IP バックボーンネットワーク上の疑似回線を使用して相互に通信します。CE ルータの R3 と R4 がクロスコネクトされたイーサネットのペアまたは L2TPv3 セッションを使用した 802.1q VLAN インターフェイス経由で通信します。L2TPv3 セッションの tu1 は、R1 上のインターフェイス int1 と R2 上のインターフェイス int4 間に設定された疑似回線です。R1 上のインターフェイス int1 に到着したすべてのパケットが、カプセル化され、疑似回線コントロールチャネル（tu1）経由で R2 に送信されます。R2 でパケットがカプセル解放され、インターフェイス int4 経由で R4 に送信されます。R4 から R3 にパケットを送信する必要がある場合は、パケットが同じパスを逆にたどります。

（注） L2TPv3 over IPv4 機能は、Cisco ASR 9000 高密度 100GE イーサネットラインカードのみでサポートされています。

（注） nV サテライト アクセス インターフェイスは、L2TPv3 over IPv4 をサポートしていません。

詳細については、 次を参照してください。

- 設定手順については、「L2TPv3 over IPv4 トンネルの設定 （184 ページ）」を参照してください。
- 設定例については、「L2TPv3 over IPv4 トンネルの設定：例 （216 ページ）」を参照してください。

# 動的セグメント疑似回線

単一セグメント疑似回線（SS-PW）は、2 つの PE ルータ間に PW セグメントが存在するポイントツーポイント疑似回線（PW）です。

この機能では、FEC 129 情報を動的に使用して、同じ自律システム（AS）の 2 つの PE ルータ間に単一セグメント疑似回線が確立されます。この機能の目的は、シスコ製ルータとサードパーティ製ルータとの相互運用性を確保することです。

## アクティブシグナリングとパッシブシグナリング

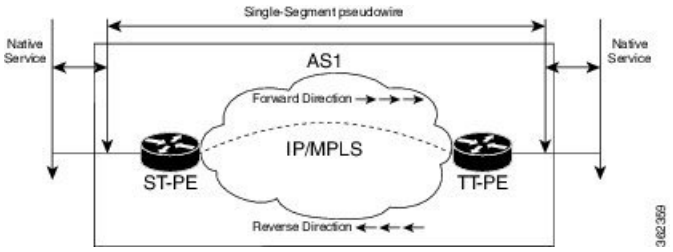
SS-PW が開始され、シグナリングメッセージの送信元となる T-PE は、送信元終端 PE（ST-PE）と呼ばれます。SS-PW シグナリングメッセージを待ち受けて応答する T-PE は、送信先終端 PE（TT-PE）と呼ばれます。

ST-PE から TT-PE へのシグナリングフローは、順方向シグナリングまたはアクティブシグナリングと呼ばれます。TT-PE から ST-PE へのシグナリングフローは、逆方向シグナリングまたはパッシブシグナリングと呼ばれます。

一般的に、プレフィックスアドレスが最も大きい PE がアクティブの役割を果たして ST-PE となり、他の PE はパッシブの TT-PE となります。

次の図は、ST-PE と TT-PE の間の SS-PW シグナリングフローを示しています。

**図 *16 : ST-PE* と *TT-PE* の間の単一セグメント疑似回線**




## 動的単一セグメント疑似回線の機能

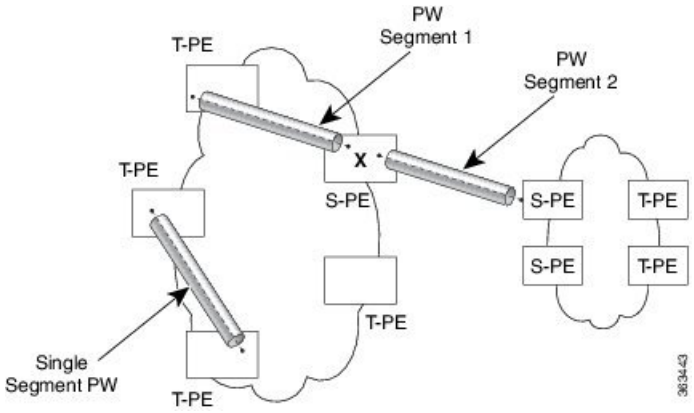
ST-PE から T-PE への疑似回線パスの動的検出は、L2 ルートテーブルを使用して実現されます。ルートテーブルのエントリ（つまり、プレフィックスと、関連付けられた L2VPN へのネクストホップのリスト）は、BGP によって入力されます。

（注） リリース 5.1.2 では、本シスコ製品は T-PE 上の TAII に到達するためのルーティング可能なプレフィックスのみをサポートしています。ルーティング可能なプレフィックスは、ターゲット LDP セッションのネイバーアドレスです。送信元から宛先へのパケットの到達可能性は、ユーザ設定によって実現されます（L2VPN 単一セグメント疑似回線の設定（105 ページ）を参照）。ただし、BGP は、すべての PE 間で L2 ルートを交換するために使用される MS-PW の後続アドレスファミリ識別子（SAFI）をサポートしています。SS-PW は、BGP MS-PW アドレスファミリを使用して機能します。他のサードパーティルータとの相互運用性を確保するために、本シスコ製品は T-PE ごとに単一の BGP MS-PW ルートをアドバタイズします。ここで、AC-ID（接続回線識別子）の値はワイルドカードエントリです。

サポートされている疑似回線機能は、pw-status、pw-grouping、および tag-impose vlan です。

次の図は、SS-PW を使用した E-line サービスネットワークを示しています。

図 *17 : SS-PW* を使用した *E-Line* サービスネットワーク




## L2VPN 単一セグメント疑似回線の設定に関する前提条件

MPLS LDP、IGP、BGP、L2VPN、およびインターフェイスを、PW の 2 つのエンドポイントで設定する必要があります。

- MPLS ラベル配布プロトコルを設定します。
- 内部ゲートウェイプロトコル（IGP）を設定します。
- ボーダー ゲートウェイ プロトコル（BGP）を設定します。
- L2VPN のインターフェイスまたは接続を設定します。

## L2VPN 単一セグメント疑似回線の設定に関する制限事項

- ルーテッド疑似回線は、Virtual Private Wire Service（VPWS）クロスコネクトでのみ有効にできます。
- クロスコネクトでは、両端を「ネイバールーテッド」疑似回線として設定することはできません。
- SS-PW はクロスコネクトの両端には設定できません。つまり、T-PE では、クロスコネクトの一方の端が SS-PW の終端となり、もう一方の端は接続回線（AC）または PW-HE である可能性があります。
- 送信元 AII と AC-ID（接続回線識別子）は、ルータごとに一意です。
- L2TP および MPLS スタティックはサポートされません。

## L2VPN 単一セグメント疑似回線の設定

ネットワークで単一セグメント疑似回線を設定するには、次の手順を実行します。

1. （オプション）関連する L2VPN グローバルパラメータの設定。「L2VPN グローバルパラメータの設定」を参照してください

この手順は、デフォルトの BGP ルート識別子（RD）自動生成値と、BGP の自律システム番号（ASN）およびルート識別子（RID）を上書きするために使用します。

2. L2VPN VPWS SS-PW の設定
3. BGP の L2VPN MS-PW アドレスファミリの設定

   アドレスファミリは、ダイナミック擬似回線ルートを交換するために BGP で設定されます。

## L2VPN グローバルパラメータの設定

L2VPN グローバルパラメータを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **router-id** *router-id*
4. **pw-routing**
5. **global-id** *global-id*
6. **bgp**
7. **rd** *route-distinguisher*
8. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 3** **router-id** *router-id*

例：

```
RP/0/RSP0/cpu 0: router(config)# router 2.2.2.2
```

ルータ ID を指定します。

**ステップ 4** **pw-routing**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# pw-routing
```

疑似回線ルーティング機能を有効にし、疑似回線ルーティング設定サブモードを開始します。

**ステップ 5** **global-id** *global-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwr)# global-id 1000
```

ルータの L2VPN グローバル ID 値を設定します。

**ステップ 6** **bgp**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwr)# bgp
```

BGP 疑似回線ルーティング機能を有効にし、BGP 設定サブモードを開始します。

**ステップ 7** **rd** *route-distinguisher*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwr-bgp)# rd 192.168.1.3:10
```

BGP ルート識別子を設定します。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## L2VPN VPWS SS-PW の設定

L2VPN VPWS SS-PW を設定するには、次の作業を実行します。

### 手順の概要

1. **configure**

2. **interface type***interface-path-id*
3. **l2vpn**
4. **xconnect group** *group-name*
5. **p2p** *xconnect-name*
6. **interface** *type interface-path-id*
7. **neighbor routed** *global-id: prefix: ac-id* **source** *ac-id*
8. （オプション） **pw-class** *class-name*
9. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface type***interface-path-id*

例：

```
RP/0/RSP0/cpu 0: routerRP/0/RP00RSP0/CPU0:router# interface TenGigE0/1/0/12
```

インターフェイス コンフィギュレーション モードを開始し、インターフェイスを設定します。

**ステップ 3** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 4** **xconnect group** *group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group pw-he1
```

自由形式の 32 文字ストリングを使用して、相互接続グループ名を設定します。

**ステップ 5** **p2p** *xconnect-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p pw-ss
```

P2P コンフィギュレーション サブモードを開始します。

**ステップ 6** **interface** *type interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface gigabitethernet 0/1/0/9
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 7** **neighbor routed** *global-id: prefix: ac-id* **source** *ac-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor routed 100:2.2.2.2:10 source 10
```

p2p クロスコネクトの疑似回線ルーティング設定サブモードを有効にします。

**ステップ 8** （オプション） **pw-class** *class-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pwr)# pw-class dynamic_sspw
```

疑似回線クラス サブモードになり、疑似回線クラス テンプレートを定義できます。

**ステップ 9** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## BGP の L2VPN MS-PW アドレスファミリの設定

BGP に L2VPN MS-PW アドレスファミリを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **router bgp** *autonomous-system-number*
3. **address-family l2vpn mspw**
4. **neighbor** *ip-address*
5. **address-family l2vpn mspw**
6. **commit** コマンドまたは **end** コマンドを使用します。

手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **router bgp** *autonomous-system-number*

例：

```
RP/0/RSP0/cpu 0: router(config)# router bgp 100
```

指定したルーティング プロセスのルータ コンフィギュレーション モードを開始します。

**ステップ 3** **address-family l2vpn mspw**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# address-family l2vpn mspw
```

L2VPN アドレス ファミリを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

**ステップ 4** **neighbor** *ip-address*

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 10.10.10.1
```

指定した自律システム内のネイバーの IP アドレスを追加します。

**ステップ 5** **address-family l2vpn mspw**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# address-family l2vpn mspw
```

ネイバーの L2VPN アドレス ファミリを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。

• [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# EVPN 仮想プライベート ワイヤ サービス（VPWS）

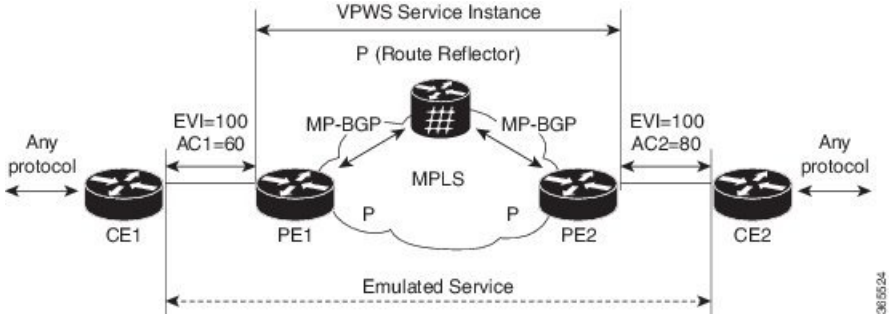
EVPN-VPWS は、ポイントツーポイント サービス用の BGP コントロール プレーン ソリューションです。これにより、PE のペア間で EVPN インスタンスを確立するためのシグナリングおよびカプセル化技術が実装されます。EVPN-VPWS には、MAC ルックアップを使用せずに、あるネットワークから別のネットワークにトラフィックを転送する機能があります。VPWS 対応の EVPN により、ポイントツーポイント イーサネット サービスにおいてシングルセグメントおよびマルチセグメント PW をシグナリングする必要性がなくなります。また、EVPN-VPWS を使用して PWHE インターフェイスとブリッジドメインアクセス疑似回線を設定することもできます。

EVPN-VPWS シングルホームテクノロジーは、IP および MPLS コアで動作します。IP コアでは BGP がサポートされ、MPLS コアではエンドポイント間でのパケットのスイッチングがサポートされます。

## EVPN–VPWS シングル ホームに関する情報

EVPN-VPWS シングル ホーム ソリューションは、EVI イーサネット自動検出ルートごとに必要です。EVPN は、すべての EVPN ルートの伝送に使用する新しい BGP ネットワーク層到達可能性情報（NLRI）を定義します。BGP 機能アドバタイズメントを使用して、2 つのスピーカーが RFC 4760 に従い、EVPN NLRI（AFI 25、SAFI 70）を確実にサポートするようにします。

EVPN VPWS のアーキテクチャでは、PE3 がコントロール プレーンでマルチプロトコル BGP を実行します。次に、EVPN-VPWS 設定を説明する図を示します。




• PE1 上の VPWS サービスには、設定時に指定する次の 3 つの要素が必要です。
  • VPN ID（EVI）
  • ローカル AC 識別子（AC1）。エミュレートされたサービスのローカル エンドを識別します。
  • リモート AC 識別子（AC2）。エミュレートされたサービスのリモート エンドを識別します。

PE1 は到達可能性を得るために、MPLS ラベルをローカル AC ごとに割り当てます。

- PE2 上の VPWS サービスは PE1 と同じ方法で設定されます。3 つの同じ要素が必要であり、サービス設定は対称になっている必要があります。

  PE2 は到達可能性を得るために、MPLS ラベルをローカル AC ごとに割り当てます。

- PE1 は各ローカルエンドポイント（AC）の EVI イーサネット AD ごとの単一の EVPN を、関連付けられた MPLS ラベルを使用してリモート PE にアドバタイズします。

  PE2 は同じタスクを実行します。

- PE2 から EVI EAD ルートごとの EVPN を受け取ると、PE1 はそのローカル L2 RIB にエントリを追加します。PE1 は AC2 に到達するパスのリスト（たとえば、ネクスト ホップが PE2 の IP アドレスであること）と AC2 の MPLS ラベルを把握しています。

  PE2 は同じタスクを実行します。

### EVPN-VPWS の利点

EVPN-VPWS の利点は次のとおりです。

- 拡張性が、疑似回線のシグナリングなしで実現されます。
- プロビジョニングの容易さ
- 疑似回線（PW）は使用されません。
- BGP のベストパス選択（最適な転送）を活用します。

## EVPN-VPWS の前提条件

- BGP が EVPN SAFI 用に設定されていることを確認します。
- EVPN ルートを交換するための「address-family l2vpn evpn」を使用した PE 間の BGP セッション。

## EVPN-VPWS に関する制限事項

- VPN ID はルータごとに一意です。
- ルートターゲットのリストを指定する場合は、PE ごとに一意である必要があります（BGP アドレスファミリごと）。

# ポイントツーポイント レイヤ 2 サービスを実装する方法

このセクションでは、ポイントツーポイントレイヤ 2 サービスの実装に必要なタスクについて説明します。

# ポイントツーポイントレイヤ 2 サービスのインターフェイスまたは接続の設定

ポイントツーポイントレイヤ2サービスのインターフェイスまたは接続を設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **l2transport**
4. **exit**
5. **interface** *type interface-path-id*
6. **commit** コマンドまたは **end** コマンドを使用します。
7. **show interface** *type interface-id*

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface** *type interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config)# interface TenGigE 0/0/0/0
```

インターフェイス コンフィギュレーション モードを開始し、インターフェイスを設定します。

**ステップ 3** **l2transport**

例：

```
RP/0/RSP0/cpu 0: router(config-if)# l2transport
```

選択したインターフェイスで L2 転送をイネーブルにします。

**ステップ 4** **exit**

例：

```
RP/0/RSP0/cpu 0: router(config-if-l2)# exit
```

現在のコンフィギュレーション モードを終了します。

**ステップ 5** **interface** *type interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config)# interface TenGigE 0/0/0/0
```

インターフェイス コンフィギュレーション モードを開始し、インターフェイスを設定します。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

**ステップ 7** **show interface** *type interface-id*

例：

```
RP/0/RSP0/cpu 0: router show interface TenGigE 0/0/0/0
```

（任意）コミットしたインターフェイスの設定を表示します。

# ローカル スイッチングの設定

ローカル スイッチングを設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group** *group-name*
4. **p2p** *xconnect-name*
5. **interface** *type interface-path-id*
6. **interface** *type interface-path-id*
7. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

**例**：

```
RP/0/RSP0/cpu 0: router(config-subif)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **xconnect group** *group-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group grp_1
```

クロスコネクト グループの名前を入力します。

**ステップ 4** **p2p** *xconnect-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p vlan1
```

ポイントツーポイント クロスコネクトの名前を入力します。

**ステップ 5** **interface** *type interface-path-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface TenGigE 0/7/0/6.5
```

インターフェイス タイプ ID を指定します。選択できる基準は、次のとおりです。

- GigabitEthernet：ギガビット イーサネット/IEEE 802.3 インターフェイス
- TenGigE：TenGigabit イーサネット/IEEE 802.3 インターフェイス
- CEM：回線エミュレーション インターフェイス

**ステップ 6** **interface** *type interface-path-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface GigabitEthernet0/4/0/30
```

インターフェイス タイプ ID を指定します。選択できる基準は、次のとおりです。

- GigabitEthernet：ギガビット イーサネット/IEEE 802.3 インターフェイス
- TenGigE：TenGigabit イーサネット/IEEE 802.3 インターフェイス

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# ローカル接続の冗長性の設定

ローカル接続の冗長性を設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group** *group-name*
4. **p2p** *xconnect-name*
5. **backup interface** *type interface-path-id*
6. **interface** *type interface-path-id*
7. **interface** *type interface-path-id*
8. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config-subif)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **xconnect group** *group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group grp_1
```

クロスコネクト グループの名前を入力します。

**ステップ 4** **p2p** *xconnect-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p vlan1
```

ポイントツーポイント クロスコネクトの名前を入力します。

**ステップ 5** **backup interface** *type interface-path-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# backup interface Bundle-Ether 0/7/0/6.5
```

ローカル接続の冗長性を設定します。

（注） 接続回線（AC）は、MC LAG に属するバンドルインターフェイスである必要があります。バックアップ インターフェイスは、バンドルまたはイーサネットポートのいずれかです。

**ステップ 6** **interface** *type interface-path-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface Bundle-Ether 0/7/0/6.2
```

インターフェイス タイプ ID を指定します。選択できる基準は、次のとおりです。

- GigabitEthernet：ギガビット イーサネット/IEEE 802.3 インターフェイス
- TenGigE：TenGigabit イーサネット/IEEE 802.3 インターフェイス
- CEM：回線エミュレーション インターフェイス

**ステップ 7** **interface** *type interface-path-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface Bundle-Ether 0/7/0/6.1
```

インターフェイス タイプ ID を指定します。選択できる基準は、次のとおりです。

- GigabitEthernet：ギガビット イーサネット/IEEE 802.3 インターフェイス
- TenGigE：TenGigabit イーサネット/IEEE 802.3 インターフェイス

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# スタティック ポイントツーポイント相互接続の設定

（注） スタティック ポイントツーポイント クロスコネクトを設定する場合、クロスコネクトに関する次の情報を考慮します。

- 相互接続はペアにより一意に識別されます。相互接続名は、グループ内で一意である必要があります。
- セグメント（接続回線または疑似回線）は一意で、1 つの相互接続だけに属することができます。
- スタティック VC のローカル ラベルはグローバルに一意で、1 つの疑似回線だけで使用できます。
- 1 台のルータにつき 16,000 以下の相互接続を設定できます。

（注） スタティック疑似回線接続はシグナリングに LDP を使用しません。

スタティック ポイントツーポイント相互接続を設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group** *group-name*
4. **p2p** *xconnect-name*
5. **interface** *type interface-path-id*
6. **neighbor** *A.B.C.D* **pw-id** *pseudowire-id*
7. **mpls static label local** { *value* } **remote** { *value* }
8. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

**例**：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **xconnect group** *group-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group grp_1
```

クロスコネクト グループの名前を入力します。

**ステップ 4** **p2p** *xconnect-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p vlan1
```

ポイントツーポイント クロスコネクトの名前を入力します。

**ステップ 5** **interface** *type interface-path-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface gigabitethernet 0/1/0/9
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 6** **neighbor** *A.B.C.D* **pw-id** *pseudowire-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor 10.2.2.2 pw-id 2000
```

クロスコネクトの疑似回線セグメントを設定します。

相互接続ピアの IP アドレスを指定するには、A.B.C.D 引数を使用します。

（注） A.B.C.D は再帰的または非再帰的プレフィックスです。

オプションで、コントロール ワードをディセーブルにするか、イーサネットまたは VLAN に transport-type を設定できます。

**ステップ 7** **mpls static label local** { *value* } **remote** { *value* }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pw)# mpls static label local 699 remote 890
```

ローカルおよびリモート ラベル ID 値を設定します。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# ダイナミック ポイントツーポイント相互接続の設定

ダイナミック ポイントツーポイント相互接続を設定するには、次の作業を実行します。

（注） ダイナミック相互接続では、LDP が稼働中である必要があります。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group** *group-name*
4. **p2p** *xconnect-name*
5. **interface** *type interface-path-id*
6. **neighbor** *A.B.C.D* **pw-id** *pseudowire-id*
7. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **xconnect group** *group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group grp_1
```

クロスコネクト グループの名前を入力します。

**ステップ 4** **p2p** *xconnect-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p vlan1
```

ポイントツーポイント クロスコネクトの名前を入力します。

**ステップ 5** **interface** *type interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface GigabitEthernet0/0/0/0.1
```

インターフェイス タイプ ID を指定します。選択できる基準は、次のとおりです。

- GigabitEthernet：GigabitEthernet/IEEE 802.3 インターフェイス。
- TenGigE：TenGigabitEthernet/IEEE 802.3 インターフェイス。
- CEM：回線エミュレーション インターフェイス

**ステップ 6** **neighbor** *A.B.C.D* **pw-id** *pseudowire-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor 2.2.2.2 pw-id 2000
```

クロスコネクトの疑似回線セグメントを設定します。

オプションで、コントロール ワードをディセーブルにするか、イーサネットまたは VLAN に transport-type を設定できます。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# Inter-AS の設定

Inter-AS の設定手順は、L2VPN 相互接続の設定作業と同じです（「スタティック ポイントツーポイント相互接続の設定」セクションおよび「ダイナミック ポイントツーポイント相互接続の設定」セクションを参照）。ただし、相互接続設定で使用されるリモート PE の IP アドレスは iBGP ピアリングを通じて到達可能です。

（注） この設定を完了するには、IBGP、EBGP、および ASBR の用語および設定に関する知識が必要です。

# L2VPN Quality of Service の設定

このセクションでは、ポートモード、VLAN モード、フレームリレーおよび ATM サブインターフェイスで L2VPN Quality of Service（QoS）を設定する方法について説明します。

## 機能制限

**l2transport** コマンドはすべての IP アドレス、L3、または CDP の設定で使用できません。

## ポート モードでの L2VPN Quality of Service ポリシーの設定

この手順では、ポート モードでの L2VPN QoS ポリシーの設定方法について説明します。

（注） ポートモードでは、インターフェイス名の形式に、サブインターフェイス番号が含まれません（たとえば、GigabitEthernet0/1/0/1）。

### 手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **l2transport**
4. **service-policy** [ **input** | **output** ] [ *policy-map-name* ]
5. **commit** コマンドまたは **end** コマンドを使用します。
6. **show qos interface** *type interface-id* **service-policy** [ **input** | **output** ] [ *policy-map-name* ]

### 手順の詳細

**ステップ 1** **configure**

**例**：

```
RP/0/RSP0/cpu 0: router# configure
```

コンフィギュレーション モードを開始します。

**ステップ 2** **interface** *type interface-path-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config)# interface GigabitEthernet 0/0/0/0
```

インターフェイス接続回線を指定します。

**ステップ 3** **l2transport**

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# l2transport
```

L2 スイッチングのインターフェイスまたは接続を設定します。

**ステップ 4** **service-policy** [ **input** | **output** ] [ *policy-map-name* ]

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# service-policy input servpol1
```

入力インターフェイスまたは出力インターフェイスに、そのインターフェイスのサービス ポリシーとして使用する QoS ポリシーを付加します。

**ステップ 5** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

**ステップ 6** **show qos interface** *type interface-id* **service-policy** [ **input** | **output** ] [ *policy-map-name* ]

**例**：

```
RP/0/RSP0/cpu 0: router# show qos interface gigabitethernet 0/0/0/0 input serpol1
```

（任意）定義した QoS サービス ポリシーを表示します。

## VLAN モードでの L2VPN Quality of Service ポリシーの設定

この手順では、VLAN モードでの L2VPN QoS ポリシーの設定方法について説明します。

（注） VLAN モードでは、インターフェイス名にサブインターフェイスを含める必要があります。例：GigabitEthernet0/1/0/1.1。l2transport コマンドは、同じ CLI 行のインターフェイス タイプに従う必要があります。例：interface GigabitEthernet 0/0/0/0.1 l2transport。

### 手順の概要

1. **configure**
2. **interface** *type interface-path-id.subinterface* **l2transport**
3. **service-policy** [ **input** | **output** ] [ *policy-map-name* ]
4. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface** *type interface-path-id.subinterface* **l2transport**

例：

```
RP/0/RSP0/cpu 0: router(config)# interface GigabitEthernet0/0/0/0.1 l2transport
```

L2 スイッチングのインターフェイスまたは接続を設定します。

（注） VLAN モードでは、interface と同じ行に **l2transport** キーワードを入力する必要があります。

**ステップ 3** **service-policy** [ **input** | **output** ] [ *policy-map-name* ]

例：

```
RP/0/RSP0/cpu 0: router(config-if)# service-policy input servpol1
```

入力インターフェイスまたは出力インターフェイスに、そのインターフェイスのサービス ポリシーとして使用する QoS ポリシーを付加します。

**ステップ 4** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# マルチセグメント疑似回線の設定

ここで説明する作業は、次のとおりです。

## マルチセグメント疑似回線設定のプロビジョニング

ポイントツーポイント（p2p）相互接続としてマルチセグメント疑似回線を設定します。P2P クロスコネクトの詳細については、「スタティック ポイントツーポイント相互接続の設定」を参照してください。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group** *group-name*
4. **p2p** *xconnect-name*
5. **neighbor** *A.B.C.D* **pw-id** *value*
6. **pw-class** *class-name*
7. **exit**
8. **neighbor** *A.B.C.D* **pw-id** *value*
9. **pw-class** *class-name*
10. **commit**

### 手順の詳細

**ステップ 1** **configure**

**例**：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

**例**：

```
RP/0/RSP0/cpu 0: router# configure
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **xconnect group** *group-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group MS-PW1
```

自由形式の 32 文字ストリングを使用して、相互接続グループ名を設定します。

**ステップ 4** **p2p** *xconnect-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p ms-pw1
```

P2P コンフィギュレーション サブモードを開始します。

**ステップ 5** **neighbor** *A.B.C.D* **pw-id** *value*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor 10.165.200.25 pw-id 100
```

相互接続の疑似回線を設定します。

IP アドレスは、該当する PE ノードの IP アドレスです。

**pw-id** は PE ノードの **pw-id** と一致する必要があります。

（注） MSPW の場合、クロスコネクト設定は、ローカル PE、S-PE、およびリモート PE で実行されます。

**ステップ 6** **pw-class** *class-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pw)# pw-class dynamic_mpls
```

疑似回線クラス サブモードになり、疑似回線クラス テンプレートを定義できます。

**ステップ 7** **exit**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pw)# exit
```

疑似回線クラス サブモードを終了し、ルータを親コンフィギュレーション モードに戻します。

**ステップ 8** **neighbor** *A.B.C.D* **pw-id** *value*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor 10.165.202.158 pw-id 300
```

相互接続の疑似回線を設定します。

IP アドレスは、該当する PE ノードの IP アドレスです。

**pw-id** は PE ノードの **pw-id** と一致する必要があります。

**ステップ 9** **pw-class** *class-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pw)# pw-class dynamic_mpls
```

疑似回線クラス サブモードになり、疑似回線クラス テンプレートを定義できます。

**ステップ 10** **commit**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pw)# commit
```

実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを続行します。

## グローバル マルチセグメント疑似回線のディスクリプションのプロビジョニング

S-PE ノードには、疑似回線切り替えポイントの Type-Length-Value（TLV）でディスクリプションが必要です。TLV は疑似回線が通過するすべてのスイッチング ポイントを記録し、トラブルシューティングのために便利な履歴を作成します。

各マルチセグメント疑似回線に独自のディスクリプションを設定できます。手順については、「相互接続のディスクリプションのプロビジョニング」を参照してください。独自のディスクリプションがない場合、このグローバルなディスクリプションが使用されます。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **description** *value*
4. **commit**

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **description** *value*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# description S-PE1
```

疑似回線切り替えポイント TLV を設定します。この TLV は、疑似回線が通過するすべてのスイッチング ポイントを記録します。

各マルチセグメント疑似回線に独自のディスクリプションを設定できます。独自のディスクリプションがない場合、このグローバルなディスクリプションが使用されます。

**ステップ 4** **commit**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# commit
```

実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを続行します。

## 相互接続のディスクリプションのプロビジョニング

S-PE ノードには、疑似回線切り替えポイントの TLV でディスクリプションが必要です。TLV は疑似回線が通過するすべてのスイッチング ポイントを記録し、トラブルシューティングのために便利な履歴を作成します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group** *group-name*
4. **p2p** *xconnect-name*
5. **description** *value*
6. **commit**

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **xconnect group** *group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group MS-PW1
```

自由形式の 32 文字ストリングを使用して、相互接続グループ名を設定します。

**ステップ 4** **p2p** *xconnect-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p ms-pw1
```

P2P コンフィギュレーション サブモードを開始します。

**ステップ 5** **description** *value*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# description MS-PW from T-PE1 to T-PE2
```

疑似回線切り替えポイント TLV を設定します。この TLV は、疑似回線が通過するすべてのスイッチング ポイントを記録します。

各マルチセグメント疑似回線に独自のディスクリプションを設定できます。独自のディスクリプションがない場合、グローバルなディスクリプションが使用されます。詳細については、「マルチセグメント疑似回線設定のプロビジョニング」を参照してください。

**ステップ 6** **commit**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# commit
```

実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを続行します。

## スイッチング ポイント TLV セキュリティのプロビジョニング

セキュリティ上の理由から、TLV を非表示にでき、それにより、疑似回線が通過するすべてのスイッチングポイントを誰かが表示することを防ぐことができます。

仮想回線接続性検証（VCCV）は、**switching-tlv** パラメータが「hide」に設定されたマルチセグメント疑似回線では機能しない場合があります。VCCV の詳細については、「L2VPN での仮想回線接続検証」を参照してください。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **pw-class** *class-name*
4. **encapsulation mpls**
5. **protocol ldp**
6. **switching-tlv hide**
7. **commit**

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **pw-class** *class-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# pw-class dynamic_mpls
```

疑似回線クラス サブモードになり、疑似回線クラス テンプレートを定義できます。

**ステップ 4 encapsulation mpls**

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-pwc)# encapsulation mpls
```

MPLS に疑似配線カプセル化を設定します。

**ステップ 5 protocol ldp**

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-pwc-encap-mpls)# protocol ldp
```

LDP に疑似回線シグナリング プロトコルを設定します。

**ステップ 6 switching-tlv hide**

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-pwc-encap-mpls)# switching-tlv hide
```

疑似回線 TLV を非表示に設定します。

**ステップ 7 commit**

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-pwc-encap-mpls)#commit
```

実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを続行します。

## マルチセグメント疑似回線のイネーブル化

**pw-status** コマンドを有効にした後、**pw-status** コマンドを使用します。**pw-status** コマンドはデフォルトでは無効になっています。**pw-status** コマンドを変更すると、L2VPN で設定されたすべての疑似回線が再プロビジョニングされます。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **pw-status**

4. **commit**

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 3** **pw-status**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# pw-status
```

このレイヤ 2 VPN で設定されるすべての疑似回線をイネーブルにします。

（注） 疑似回線ステータスを無効にするには、**pw-status disable** コマンドを使用します。

**ステップ 4** **commit**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# commit
```

実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを続行します。

# 疑似回線冗長性の設定

疑似回線の冗長性により、プライマリ疑似回線で障害が発生した場合のバックアップ疑似回線を設定できます。プライマリ疑似回線が障害になった場合、PE ルータをバックアップ疑似回線に切り替えることができます。復旧後にプライマリ疑似回線の運用が再開するように選択できます。

次のトピックでは、疑似回線の冗長性を設定する方法について説明します。

# ポイントツーポイント疑似回線の冗長性の設定

バックアップ遅延のためにポイントツーポイント疑似回線の冗長性を設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **pw-class class-name**
4. **backup disable {delay value | never}**
5. **exit**
6. **xconnect group group-name**
7. **p2p {xconnect-name}**
8. **neighbor A.B.C.D pw-id value**
9. **pw-class class-name**
10. **backup {neighbor A.B.C.D} {pw-id value}**
11. **end** または **commit**

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/CPU0:router# configure
```

コンフィギュレーション モードに入ります。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/CPU0:router(config)# l2vpn
 RP/0/RSP0/CPU0:router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **pw-class class-name**

例：

```
RP/O/RSP0/CPU0:router(config-l2vpn)# pw-class path1
RP/0/RSP0/CPU0:router(config-l2vpn-pwc)#
```

疑似回線クラス名を設定します。

**ステップ 4** **backup disable {delay value | never}**

例：

```
RP/0/RSP0/CPU0:router(config-l2vpn-pwc)# backup disable delay 20
```

このコマンドは、プライマリ疑似回線がアクティブになった後、バックアップ疑似回線から引き継ぐまでの待ち時間を指定します。

- **delay** キーワードを使用して、プライマリ疑似回線がアップ状態になってから、セカンダリ疑似回線が非アクティブになるまでの経過秒数を指定します。範囲は 0 ～ 180 です。
- プライマリ疑似回線が再び使用できるようになった場合に、セカンダリ疑似回線で障害が発生しない限り、セカンダリ疑似回線からプライマリ疑似回線にフォールバックしないように指定するには、**never** キーワードを使用します。

**ステップ 5** **exit**

例：

```
RP/0/RSP0/CPU0:router(config-l2vpn-pwc)# exit
RP/O/RSP0/CPU0:router(config-l2vpn)#
```

現在のコンフィギュレーション モードを終了します。

**ステップ 6** **xconnect group group-name**

例：

```
RP/0/RSP0/CPU0:router(config-l2vpn)# xconnect group A
RP/0/RSP0/CPU0:router(config-l2vpn-xc)#
```

クロスコネクトグループの名前を入力します。

**ステップ 7** **p2p {xconnect-name}**

例：

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc)# p2p xc1
```

ポイントツーポイント クロスコネクトの名前を入力します。

**ステップ 8** **neighbor A.B.C.D pw-id value**

例：

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# neighbor 10.1.1.2 pw-id 2
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw)#
```

クロスコネクトの疑似回線セグメントを設定します。

**ステップ 9** **pw-class class-name**

例：

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw)#pw-class path1
```

疑似回線クラス名を設定します。

**ステップ 10** **backup {neighbor A.B.C.D} {pw-id value}**

例：

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw)# backup neighbor 10.2.2.2 pw-id 5
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw-backup)#
```

相互接続のバックアップ疑似回線を設定します。

- **neighbor** キーワードを使用して、相互接続するピアを指定します。A.B.C.D 引数はピアの IPv4 アドレスです。
- **pw-id** キーワードを使用して、疑似回線 ID を設定します。範囲は 1 ～ 4294967295 です。

**ステップ 11** **end** または **commit**

**例**：

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw-backup)#end
```

または

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw-backup)#commit
```

設定変更を保存します。

- **end** コマンドを実行すると、変更をコミットするように要求されます。

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?
[cancel]:
```

  - **yes** と入力すると、実行設定ファイルへの変更が保存され、設定セッションが終了して、ルータが EXEC モードに戻ります。
  - **no** と入力すると、設定セッションが終了して、ルータが EXEC モードに戻ります。設定の変更はコミットされません。
  - **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。

- 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

## バックアップ疑似回線への強制的な手動切り替え

ルータを強制的にバックアップに切り替える、またはプライマリ疑似回線に戻すには、EXEC モードで **l2vpn switchover** コマンドを使用します。EXEC モード

手動切り替えは、コマンドが入力されたとき、コマンドで指定されたピアが実際に使用可能であり、相互接続が完全なアクティブ状態に移行する場合に限り実行されます。

# バックアップ疑似回線の設定

ポイントツーポイント ネイバーのバックアップ疑似回線を設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group** *group-name*
4. **p2p** *xconnect-name*
5. **neighbor** *ip-address* **pw-id** *value*
6. **neighbor** { *A.B.C.D* } { **pw-id** *value* }
7. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

### ステップ 1 configure

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

### ステップ 2 l2vpn

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

### ステップ 3 xconnect group *group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group A
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)#
```

クロスコネクト グループの名前を入力します。

### ステップ 4 p2p *xconnect-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p rtrX_to_rtrY
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)#
```

ポイントツーポイント クロスコネクトの名前を入力します。

**ステップ 5** **neighbor** *ip-address* **pw-id** *value*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor 1.1.1.1 pw-id 2
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pw)#
```

クロスコネクトの疑似回線セグメントを設定します。

**ステップ 6** **neighbor** { *A.B.C.D* } { **pw-id** *value* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor 10.1.1.2 pw-id 11
```

相互接続のバックアップ疑似回線を設定します。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## ポイントツーポイント疑似回線の冗長性の設定

バックアップ遅延のためにポイントツーポイント疑似回線の冗長性を設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **pw-class** { *class-name* }
4. **backup disable** { **delay***value* | **never** }
5. **exit**
6. **xconnect group** *group-name*
7. **p2p** { *xconnect-name* }
8. **neighbor** { *A.B.C.D* } { **pw-id** *value* }
9. **pw-class** { *class-name* }
10. **backup** { **neighbor** *A.B.C.D* } { **pw-id** *value* }
11. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **pw-class** { *class-name* }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# pw-class path1
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc)#
```

疑似回線クラス名を設定します。

**ステップ 4** **backup disable** { **delay***value* | **never** }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc)# backup disable delay 20
```

このコマンドは、プライマリ疑似回線がアクティブになってから、バックアップ疑似回線を引き継ぐまでの待ち時間を指定します。

- **delay** キーワードを使用して、プライマリ疑似回線がアップ状態になってから、セカンダリ疑似回線が非アクティブになるまでの経過秒数を指定します。範囲は 0 ～ 180 秒です。
- プライマリ疑似回線が再び使用できるようになった場合に、セカンダリ疑似回線で障害が発生しない限り、セカンダリ疑似回線からプライマリ疑似回線にフォールバックしないように指定するには、**never** キーワードを使用します。

**ステップ 5** **exit**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc)# exit
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

現在のコンフィギュレーション モードを終了します。

**ステップ 6** **xconnect group** *group-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group A
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)#
```

クロスコネクト グループの名前を入力します。

**ステップ 7** **p2p** { *xconnect-name* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p xc1
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)#
```

ポイントツーポイント クロスコネクトの名前を入力します。

**ステップ 8** **neighbor** { *A.B.C.D* } { **pw-id** *value* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor 10.1.1.2 pw-id 2
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pw)#
```

クロスコネクトの疑似回線セグメントを設定します。

**ステップ 9** **pw-class** { *class-name* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pw)# pw-class path1
```

疑似回線クラス名を設定します。

**ステップ 10** **backup** { **neighbor** *A.B.C.D* } { **pw-id** *value* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pw)# backup neighbor 10.2.2.2 pw-id 5
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pw-backup)#
```

相互接続のバックアップ疑似回線を設定します。

- **neighbor** キーワードを使用して、相互接続するピアを指定します。A.B.C.D 引数はピアの IPv4 アドレスです。
- **pw-id** キーワードを使用して、疑似回線 ID を設定します。範囲は 1 ～ 4294967295 です。

**ステップ 11** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## バックアップ疑似回線への強制的な手動切り替え

ルータをバックアップまたはプライマリ疑似回線に強制的に切り替えるには、EXEC モードで **l2vpn switchover** コマンドを使用します。

手動切り替えは、コマンドが入力されたとき、コマンドで指定されたピアが実際に使用可能であり、相互接続が完全なアクティブ状態に移行する場合に限り実行されます。

# 優先トンネル パスの設定

この手順では、優先トンネル パスを設定する方法について説明します。

（注） 優先パスの設定に使用されるトンネルは、MPLS トラフィック エンジニアリング（MPLS-TE）トンネルです。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **pw-class** {*name*}
4. **encapsulation mpls**
5. **preferred-path** {**interface**} {**tunnel-ip** *value* | **tunnel-te** *value* | **tunnel-tp** *value*} [**fallback disable**]
6. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RP0/CPU0:router# configure
```

コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RP0/CPU0:router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **pw-class** {*name*}

**例**：

```
RP/0/RP0/CPU0:router(config-l2vpn)# pw-class path1
```

疑似回線クラス名を設定します。

**ステップ 4** **encapsulation mpls**

**例**：

```
RP/0/RP0/CPU0:router(config-l2vpn-pwc)# encapsulation mpls
```

MPLS に疑似回線カプセル化を設定します。

**ステップ 5** **preferred-path** {**interface**} {**tunnel-ip** *value* | **tunnel-te** *value* | **tunnel-tp** *value*} [**fallback disable**]

**例**：

```
RP/0/RP0/CPU0:router(config-l2vpn-pwc-encap-mpls)# preferred-path interface tunnel-te 11 fallback
disable
```

優先パス トンネルを設定します。フォールバックのディセーブル化の設定が使用されており、優先パスとして設定されている TE/TP トンネルがダウン状態になると、対応する疑似回線もダウン状態になることがあります。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# PW ステータス OAM の設定

疑似回線ステータス OAM を設定するには、次の作業を実行します。

### 手順の概要

1. configure
2. l2vpn
3. pw-oam refresh transmit seconds
4. end または **commit**

### 手順の詳細

**ステップ 1** configure

**例**：

```
RP/0/RSP0RP0/CPU0:router# configure
```

コンフィギュレーション モードを開始します。

**ステップ 2** l2vpn

**例**：

```
RP/0/RSP0RP0/CPU0:router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** pw-oam refresh transmit seconds

**例**：

```
RP/0/RSP0RP0/CPU0:router(config-l2vpn)# pw-oam refresh transmit 100
```

疑似回線 OAM 機能を有効にします。

（注）　リフレッシュの送信間隔範囲は 1 ～ 40 秒です。

**ステップ 4** end または **commit**

**例**：

```
RP/0/RSP0RP0/CPU0:router(config-l2vpn)# end
```

または

```
RP/0/RSP0RP0/CPU0:router(config-l2vpn)# commit
```

設定変更を保存します。

- end コマンドを実行すると、変更をコミットするように要求されます。

  ```
  Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:
  ```

  - **yes** と入力すると、実行設定ファイルに変更が保存され、設定セッションが終了して、ルータが **EXEC** モードに戻ります。
  - **no** と入力すると、設定セッションが終了して、ルータが **EXEC** モードに戻ります。設定の変更はコミットされません。
  - **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

# フローベースのロード バランシングのイネーブル化

フローベースのロード バランシングをイネーブルにするには、次の作業を実行します。

## 手順の概要

1. configure
2. l2vpn
3. load-balancing flow {src-dst-mac | src-dst-ip}
4. end または **commit**

## 手順の詳細

**ステップ 1** configure

**例**：

```
RP/0/RSP0RP0/CPU0:router# configure
```

コンフィギュレーション モードを開始します。

**ステップ 2** l2vpn

**例**：

```
RP/0/RSP0RP0/CPU0:router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** load-balancing flow {src-dst-mac | src-dst-ip}

**例**：

```
RP/0/RSP0RP0/CPU0:router(config-l2vpn)# load-balancing flow src-dst-ip
```

L2VPN 下のすべての疑似回線およびバンドル EFP に対しフロー ベースのロード バランシングをイネーブルにします。ただし、疑似回線クラスを通じて疑似回線に対して、およびEFP-hashを通じてバンドルに対して明示的に指定されている場合は除きます。

**ステップ 4** end または **commit**

**例**：

```
RP/0/RSP0RP0/CPU0:router(config-l2vpn)# end
```

または

```
RP/0/RSP0RP0/CPU0:router(config-l2vpn)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、変更をコミットするように要求されます。

  ```
  Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:
  ```

  - **yes** と入力すると、実行設定ファイルに変更が保存され、設定セッションが終了して、ルータが **EXEC** モードに戻ります。

- **no** と入力すると、設定セッションが終了して、ルータが **EXEC** モードに戻ります。設定の変更はコミットされません。
- **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。

• 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

# 疑似回線クラスのフローベースのロードバランシングのイネーブル化

疑似回線クラスに対しフローベースのロードバランシングをイネーブルにするには、次の作業を実行します。

### 手順の概要

1. configure
2. l2vpn
3. pw-class {name}
4. encapsulation mpls
5. load-balancing pw-label
6. end または **commit**

### 手順の詳細

**ステップ 1** configure

**例**：

```
RP/0/RSP0RP0/CPU0:router# configure
```

コンフィギュレーション モードを開始します。

**ステップ 2** l2vpn

**例**：

```
RP/0/RSP0RP0/CPU0:router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** pw-class {name}

**例**：

```
RP/0/RSP0RP0/CPU0:router(config-l2vpn)# pw-class path1
```

疑似回線クラス名を設定します。

**ステップ 4** encapsulation mpls

**例**：

```
RP/0/RSP0RP0/CPU0:router(config-l2vpn-pwc)# encapsulation mpls
```

MPLS に疑似回線カプセル化を設定します。

**ステップ 5** load-balancing pw-label

**例**：

```
RP/0/RSP0RP0/CPU0:router(config-l2vpn-pwc-encap-mpls)# load-balancing pw-label
```

仮想回線ベースのロードバランシングを使用するために、定義されたクラスを使用してすべての疑似回線をイネーブルにします。

**ステップ 6** end または **commit**

**例**：

```
RP/0/RSP0RP0/CPU0:router(config-l2vpn-pwc-encap-mpls)# end
```

または

```
RP/0/RSP0RP0/CPU0:router(config-l2vpn-pwc-encap-mpls)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、変更をコミットするように要求されます。

  ```
  Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:
  ```

  - **yes** と入力すると、実行設定ファイルに変更が保存され、設定セッションが終了して、ルータが **EXEC** モードに戻ります。
  - **no** と入力すると、設定セッションが終了して、ルータが **EXEC** モードに戻ります。設定の変更はコミットされません。
  - **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

# 疑似回線のグループ化のイネーブル化

疑似回線のグループ化をイネーブルにするには、次の作業を実行します。

**手順の概要**

1. **configure**
2. **l2vpn**
3. **pw-grouping**
4. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **pw-grouping**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# pw-grouping
```

疑似回線のグループ化をイネーブルにします。

**ステップ 4** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# マルチキャスト接続の設定

『Cisco ASR 9000 Series Aggregation Services Router Multicast Configuration Guide』の「Implementing Multicast Routing on Cisco ASR 9000 Series Aggregation Services Routers」モジュールおよび『Cisco ASR 9000 Series Aggregation Services Router Multicast Command Reference』の「Multicast Routing and Forwarding Commands on Cisco ASR 9000 Series Aggregation Services Routers」モジュールを参照してください。

### 手順の概要

1. configure
2. multicast-routing [address-family ipv4]
3. interface all enable
4. exit

5. router igmp
6. version {1 | 2 | 3}
7. end または **commit**
8. show pim [ipv4] group-map **[ip-address-name] [info-source]**
9. show pim [vrf **vrf-name**] [ipv4] topology **[source-ip-address [group-ip-address]** | entry-flag **flag** | interface-flag | summary] [route-count]

## 手順の詳細

**ステップ 1** configure

**例**：

```
RP/0/RSP0/CPU0:router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** multicast-routing [address-family ipv4]

**例**：

```
RP/0/RSP0/CPU0:router(config)# multicast-routing
```

マルチキャスト ルーティング コンフィギュレーション モードを開始します。

- マルチキャストプロセス（**MRIB**、**MFWD**、**PIM**、および **IGMP**）が起動します。
- **IPv4** では、**IGMP** バージョン 3 はデフォルトで有効になっています。
- IPv4 の場合は、次を使用します。

  ```
  address-family ipv4
  ```

  キーワード

**ステップ 3** interface all enable

**例**：

```
RP/0/RSP0/CPU0:router(config-mcast-ipv4)# interface all enable
```

新規および既存のすべてのインターフェイスでマルチキャスト ルーティングおよび転送をイネーブルにします。

**ステップ 4** exit

**例**：

```
RP/0/RSP0/CPU0:router(config-mcast-ipv4)# exit
```

マルチキャスト ルーティング コンフィギュレーション モードを終了し、ルータを親コンフィギュレーション モードに戻します。

（注） リーフ PE の場合、ブリッジドメインで IGMPSN を有効にするには、IGMPSN プロファイル内で内部クエリ元を設定していることを確認します。

**ステップ 5** router igmp

**例**：

```
RP/0/RSP0/CPU0:router(config)# router igmp
```

（任意）ルータ IGMP コンフィギュレーション モードを開始します。

**ステップ 6** version {1 | 2 | 3}

例：

```
RP/0/RSP0/CPU0:router(config-igmp)# version 3
```

（任意）ルータ インターフェイスで使用する IGMP バージョンを選択します。

- IGMP のデフォルトはバージョン 3 です。
- ホスト レシーバは、PIM-SSM 動作の IGMPv3 をサポートする必要があります。
- このコマンドがルータ IGMP コンフィギュレーション モードで設定されている場合、パラメータはすべての新規および既存インターフェイスによって継承されます。これらのパラメータは、インターフェイス コンフィギュレーション モードでインターフェイスごとに上書きできます。

**ステップ 7** end または **commit**

例：

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# end
```

または

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#commit
```

設定変更を保存します。

- **end** コマンドを実行すると、変更をコミットするように要求されます。

  ```
  Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:
  ```

  - **yes** と入力すると、実行設定ファイルに変更が保存され、設定セッションが終了して、ルータが **EXEC** モードに戻ります。
  - **no** と入力すると、設定セッションが終了して、ルータが **EXEC** モードに戻ります。設定の変更はコミットされません。
  - **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

**ステップ 8** show pim [ipv4] group-map **[ip-address-name] [info-source]**

例：

```
RP/0//CPU0:router# show pim ipv4 group-map
```

（任意）グループと **PIM** 間モードのマッピングを表示します。

**ステップ 9** show pim [vrf **vrf-name]** [ipv4] topology **[source-ip-address [group-ip-address]** | entry-flag **flag** | interface-flag | summary] [route-count]

例：

```
RP/0/RSP0/CPU0:router# show pim topology
```

（任意）特定のグループまたはすべてのグループの PIM トポロジ テーブル情報を表示します。

# AToM IP インターワーキングの設定

AToM IP インターワーキングを設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group***group-name*
4. **p2p***xconnect-name*
5. **interworking ipv4**
6. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

### ステップ 1 configure

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

### ステップ 2 l2vpn

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

### ステップ 3 xconnect group*group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group grp_1
```

クロスコネクト グループの名前を入力します。

### ステップ 4 p2p*xconnect-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p vlan1
```

ポイントツーポイント クロスコネクトの名前を入力します。

### ステップ 5 interworking ipv4

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interworking ipv4
```

P2P で IPv4 インターワーキングを設定します。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# PPP IP インターワーキングの設定

PPP IP インターワーキングを設定するには、次の作業を実行します。

## 手順の概要

1. configure
2. interface type **interface-path-id**
3. encapsulation ppp
4. l2transport
5. end
6. l2vpn
7. xconnect group **group-name**
8. p2p **xconnect-name**
9. interface type **interface-path-id**
10. interface type **interface-path-id**
11. interworking ipv4
12. interface type **interface-path-id**
13. neighbor**A.B.C.D**pw-id
14. pw-class **interface-path-id**
15. exit
16. interworking ipv4
17. end または **commit**

## 手順の詳細

**ステップ 1** configure

例：

```
RP/0/0/CPU0:router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** interface type **interface-path-id**

例：

```
RP/0/RSP0/CPU0:router(config)# interface Serial0/2/1/0/1/1/1:0
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 3** encapsulation ppp

例：

```
RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp
```

PPP にカプセル化タイプを設定します。

**ステップ 4** l2transport

例：

```
RP/0/RSP0/CPU0:router(config-if)# l2transport
```

選択したインターフェイスでレイヤ 2 トランスポートをイネーブルにします。

**ステップ 5** end

例：

```
RP/0/RSP0/CPU0:router(config-if-l2)# end
```

グローバル コンフィギュレーション モードに戻ります。

**ステップ 6** l2vpn

例：

```
RP/0/RSP0/CPU0:router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 7** xconnect group **group-name**

例：

```
RP/0/RSP0/CPU0:router(config-l2vpn)# xconnect group grp_1
```

クロスコネクト グループの名前を入力します。

**ステップ 8** p2p **xconnect-name**

例：

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc)# p2p 1
```

ポイントツーポイント クロスコネクトの名前を入力します。

**ステップ 9** interface type **interface-path-id**

例：

```
RP/0/RSP0/CPU0:router(config)# interface Serial0/2/1/0/1/1/1:0
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 10** interface type **interface-path-id**

例：

```
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/0/0/1.1
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 11** interworking ipv4

**例**：

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# interworking ipv4
```

P2P で IPv4 インターワーキングを設定します。

**ステップ 12** interface type **interface-path-id**

**例**：

```
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/0/0/1.1
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 13** neighbor**A.B.C.Dpw-id**

**例**：

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# interface Serial0/0/0/0/2/1/1:0
```

クロスコネクトの疑似回線セグメントを設定します。

相互接続ピアの IP アドレスを指定するには、A.B.C.D 引数を使用します。

（注） A.B.C.D は再帰的または非再帰的プレフィックスです

オプションで、コントロールワードを無効にするか、イーサネットまたは VLAN に transport-type を設定できます。

**ステップ 14** pw-class **interface-path-id**

**例**：

```
RP/0/RSP0/CPU0:router (config-l2vpn-xc-p2p-pw)# pw-class class_c1
```

疑似回線クラス サブモードになり、疑似回線クラス テンプレートを定義できます。

**ステップ 15** exit

**例**：

```
RP/0/RSP0/CPU0:router (config-l2vpn-xc-p2p-pw)# exit
```

現在のコンフィギュレーション モードを終了します。

**ステップ 16** interworking ipv4

**例**：

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# interworking ipv4
```

P2P で IPv4 インターワーキングを設定します。

**ステップ 17** end または **commit**

**例**：

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# end
```

または

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#commit
```

設定変更を保存します。

- **end** コマンドを実行すると、変更をコミットするように要求されます。

  ```
  Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:
  ```

  - **yes** と入力すると、実行設定ファイルに変更が保存され、設定セッションが終了して、ルータが **EXEC** モードに戻ります。
  - **no** と入力すると、設定セッションが終了して、ルータが **EXEC** モードに戻ります。設定の変更はコミットされません。
  - **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

# PPP とイーサネット間の IP インターワーキングの設定

PPP IP インターワーキングを設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **interface type***interface-path-id*
3. **l2transport**
4. **end**
5. **l2vpn**
6. **xconnect group***group-name*
7. **p2p***xconnect-name*
8. **interface type***interface-path-id*
9. **interface type** *interface-path-id*
10. **interworking ipv4**
11. **interface type** *interface-path-id*
12. **neighbor***A.B.C.D* **pw-id**
13. **pw-class***class-name*
14. **pw-class***class-name*
15. **exit**
16. **interworking ipv4**
17. **commit** コマンドまたは **end** コマンドを使用します。

手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface type***interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config)# interface Serial0/2/1/0/1/1/1:0
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 3** **l2transport**

例：

```
RP/0/RSP0/cpu 0: router(config-if)# l2transport
```

選択したインターフェイスでレイヤ 2 トランスポートをイネーブルにします。

**ステップ 4** **end**

例：

```
RP/0/RSP0/cpu 0: router(config-if-l2)# end
```

グローバル コンフィギュレーション モードに戻ります。

**ステップ 5** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 6** **xconnect group***group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group grp_1
```

クロスコネクト グループの名前を入力します。

**ステップ 7** **p2p***xconnect-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p 1
```

ポイントツーポイント クロスコネクトの名前を入力します。

**ステップ 8** **interface type***interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config)# interface Serial0/2/1/0/1/1/1:0
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 9** **interface type** *interface-path-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config)# interface GigabitEthernet0/0/0/1.1
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 10** **interworking ipv4**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interworking ipv4
```

P2P で IPv4 インターワーキングを設定します。

**ステップ 11** **interface type** *interface-path-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface Serial0/0/0/0/2/1/1:0
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 12** **neighbor***A.B.C.D***pw-id**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface Serial0/0/0/0/2/1/1:0
```

クロスコネクトの疑似回線セグメントを設定します。

相互接続ピアの IP アドレスを指定するには、A.B.C.D 引数を使用します。

（注） A.B.C.D は再帰的または非再帰的プレフィックスです

オプションで、コントロールワードを無効にするか、イーサネットまたは VLAN に transport-type を設定できます。

**ステップ 13** **pw-class***class-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface Serial0/0/0/0/2/1/1:0
```

クロスコネクトの疑似回線セグメントを設定します。

相互接続ピアの IP アドレスを指定するには、A.B.C.D 引数を使用します。

（注） A.B.C.D は再帰的または非再帰的プレフィックスです

オプションで、コントロールワードを無効にするか、イーサネットまたは VLAN に transport-type を設定できます。

**ステップ 14** **pw-class***class-name*

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc-p2p-pw)# pw-class class_cem
```

疑似回線クラス サブモードになり、疑似回線クラス テンプレートを定義できます。

**ステップ 15** **exit**

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc-p2p-pw)# exit
```

現在のコンフィギュレーション モードを終了します。

**ステップ 16** **interworking ipv4**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interworking ipv4
```

P2P で IPv4 インターワーキングを設定します。

**ステップ 17** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# MLPPP IP インターワーキングの設定

cHDLC IP インターワーキングを設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **interface type***interface-path-id*
3. **multilink [fragment|interleave|ncp]**
4. **l2transport**
5. **end**
6. **l2vpn**
7. **xconnect group** *group-name*
8. **p2p** *xconnect-name*
9. **interface type***interface-path-id*
10. **interface type***interface-path-id*
11. **interworking ipv4**
12. **interface type***interface-path-id*
13. **neighbor***{A.B.C.D}***{pw-id***value}*
14. **pw-class***class-name*
15. **exit**
16. **interworking ipv4**
17. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface type***interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config)# interface Multilink0/2/1/0/1
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 3** **multilink [fragment|interleave|ncp]**

例：

```
RP/0/RSP0/cpu 0: router(config-if)# multilink
```

マルチリンク パラメータを変更します。

**ステップ 4** **l2transport**

例：

```
RP/0/RSP0/cpu 0: router(config-if-multilink)# l2transport
```

選択したインターフェイスでレイヤ 2 トランスポートをイネーブルにします。

**ステップ 5** **end**

例：

```
RP/0/RSP0/cpu 0: router(config-if-l2)# end
```

グローバル コンフィギュレーション モードに戻ります。

**ステップ 6** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 7** **xconnect group** *group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group grp_1
```

クロスコネクト グループの名前を入力します。

**ステップ 8** **p2p** *xconnect-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p 1
```

ポイントツーポイント クロスコネクトの名前を入力します。

**ステップ 9** **interface type***interface-path-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config)# interface Serial0/2/1/0/1/1/1:0
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 10** **interface type***interface-path-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config)# interface GigabitEthernet0/0/0/1.1
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 11** **interworking ipv4**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interworking ipv4
```

P2P で IPv4 インターワーキングを設定します。

**ステップ 12** **interface type***interface-path-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface Serial0/0/0/0/2/1/1:0
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 13** **neighbor***{A.B.C.D}***{pw-id***value}*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor 120.120.120.120 pw-id 3
```

クロスコネクトの疑似回線セグメントを設定します。

相互接続ピアの IP アドレスを指定するには、A.B.C.D 引数を使用します。

（注） A.B.C.D は再帰的または非再帰的プレフィックスです

オプションで、コントロールワードを無効にするか、イーサネットまたは VLAN に transport-type を設定できます。

**ステップ 14** **pw-class***class-name*

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc-p2p-pw)# pw-class class_cem
```

疑似回線クラス サブモードになり、疑似回線クラス テンプレートを定義できます。

**ステップ 15** **exit**

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc-p2p-pw)# exit
```

現在のコンフィギュレーション モードを終了します。

**ステップ 16** **interworking ipv4**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interworking ipv4
```

P2P で IPv4 インターワーキングを設定します。

**ステップ 17** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# Circuit Emulation over Packet Switched Network の設定

CEoP を設定するには、次の作業を実行します。

## CEM 接続回線の疑似回線への追加

CEM 接続回線を疑似回線に追加するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group***group-name*
4. **p2p***xconnect-name*
5. **interface type** *interface-path-id*
6. **neighbor***A.B.C.D* **pw-id**
7. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **xconnect group***group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group grp_1
```

クロスコネクト グループの名前を入力します。

**ステップ 4** **p2p***xconnect-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p vlan1
```

ポイントツーポイント クロスコネクトの名前を入力します。

**ステップ 5** **interface type** *interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface CEM0/1/0/9:10
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 6** **neighbor***A.B.C.D* **pw-id**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor 120.120.120.120 pw-id 3
```

クロスコネクトの疑似回線セグメントを設定します。

相互接続ピアの IP アドレスを指定するには、A.B.C.D 引数を使用します。

（注） A.B.C.D は再帰的または非再帰的プレフィックスです

オプションで、コントロールワードを無効にするか、イーサネットまたは VLAN に transport-type を設定できます。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# 疑似回線クラスの関連付け

接続回線を疑似回線クラスと関連付けるには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **pw-class** *class-name*
4. **encapsulation mpls**
5. **protocol ldp**
6. **end**
7. **xconnect group** *group-name*
8. **p2p** *xconnect-name*
9. **interface** *type interface-path-id*
10. **neighbor** *A.B.C.D* **pw-id** *pseudowire-id*
11. **pw-class** *class-name*
12. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モード を開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router (config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 3** **pw-class** *class-name*

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn)# pw-class class_cem
```

疑似回線クラス サブモードになり、疑似回線クラス テンプレートを定義できます。

**ステップ 4** **encapsulation mpls**

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-pwc)# encapsulation mpls
```

MPLS に疑似配線カプセル化を設定します。

**ステップ 5** **protocol ldp**

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-pwc-encap-mpls)# protocol ldp
```

LDP に疑似回線シグナリング プロトコルを設定します。

**ステップ 6** **end**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc-encap-mpls)# end
```

システムから変更をコミットするように求められます。

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:
```

- • **yes** と入力すると、実行設定ファイルへの変更が保存され、設定セッションが終了して、ルータが EXEC モードに戻ります。
  • **no** と入力すると、設定セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
  • cancel と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。

**ステップ 7** **xconnect group** *group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group grp_1
```

相互接続グループを設定します。

**ステップ 8** **p2p** *xconnect-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p vlan1
```

ポイントツーポイント相互接続を設定します。

**ステップ 9** **interface** *type interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface CEM0/1/0/9:20
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 10** **neighbor** *A.B.C.D* **pw-id** *pseudowire-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor 10.2.2.2 pw-id 11
```

クロスコネクトの疑似回線セグメントを設定します。

相互接続ピアの IP アドレスを指定するには、A.B.C.D 引数を使用します。

（注） A.B.C.D は再帰的または非再帰的プレフィックスです。

オプションで、コントロール ワードをディセーブルにするか、イーサネットまたはVLANにtransport-typeを設定できます。

（注） 疑似回線ステータス（pw-status）はデフォルトで有効になっています。必要に応じて、**pw-status disable** コマンドを使用して疑似回線ステータスを無効にします。

**ステップ 11** **pw-class** *class-name*

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc-p2p)# pw-class class_cem
```

指定した疑似回線クラスを P2P 接続回線と関連付けます。

**ステップ 12** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## 疑似回線ステータスのイネーブル化

疑似回線ステータスをイネーブルにするには、次の作業を実行します。

### 手順の概要

1. configure
2. l2vpn
3. pw-status
4. commit

### 手順の詳細

**ステップ 1** configure

**例**：

```
RP/0/RSP0/CPU0:router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** l2vpn

**例**：

```
RP/0/RSP0/CPU0:router (config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 3** pw-status

**例**：

```
RP/0/RSP0/CPU0:router (config-l2vpn)# pw-status
```

このレイヤ 2 VPN で設定されるすべての疑似回線をイネーブルにします。

（注） 疑似回線ステータスをディセーブルにするには、pw-status disable コマンドを使用します。

**ステップ 4** commit

**例**：

```
RP/0/RSP0/CPU0:router (config-l2vpn)# commit
```

実行コンフィギュレーション ファイルに設定変更を保存し、コンフィギュレーション セッションを続行します。

## バックアップ疑似回線の設定

ポイントツーポイント ネイバーのバックアップ疑似回線を設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group** *group-name*
4. **p2p** *xconnect-name*
5. **neighbor** *ip-address* **pw-id** *value*
6. **neighbor** { *A.B.C.D* } { **pw-id** *value* }
7. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

**例**：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

**例**：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **xconnect group** *group-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group A
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)#
```

クロスコネクト グループの名前を入力します。

**ステップ 4** **p2p** *xconnect-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p rtrX_to_rtrY
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)#
```

ポイントツーポイント クロスコネクトの名前を入力します。

**ステップ 5** **neighbor** *ip-address* **pw-id** *value*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor 1.1.1.1 pw-id 2
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pw)#
```

クロスコネクトの疑似回線セグメントを設定します。

**ステップ 6** **neighbor** { *A.B.C.D* } { **pw-id** *value* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor 10.1.1.2 pw-id 11
```

相互接続のバックアップ疑似回線を設定します。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# L2VPN ノンストップ ルーティングの設定

L2VPN ノンストップ ルーティングを設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **nsr**
4. **logging nsr**
5. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モード を開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 3** **nsr**

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn)# nsr
```

L2VPN ノンストップ ルーティングをイネーブルにします。

**ステップ 4** **logging nsr**

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn)# logging nsr
```

NSR イベントのロギングをイネーブルにします。

**ステップ 5** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# MPLS LDP ノンストップルーティングの設定

アクティブとスタンバイの Label Distribution Protocol（LDP; ラベル配布プロトコル）間でラベル情報を同期するために、LDP のノンストップルーティング（NSR）を有効にするには、次の作業を実行します。リリース 6.1.1 以降では、ステートフル LDP 機能の導入により、アクティブとスタンバイの LDP 間でラベル情報を同期するように LDP NSR を明示的に設定する必要があります。

## 手順の概要

1. **configure**
2. **mpls ldp**
3. **nsr**
4. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モード を開始します。

**ステップ 2** **mpls ldp**

例：

```
RP/0/RSP0/cpu 0: router(config)# mpls ldp
```

MPLS LDP コンフィギュレーション モードを開始します。

**ステップ 3** **nsr**

例：

```
RP/0/RSP0/cpu 0: router(config-ldp)# nsr
```

LDP ノンストップ ルーティングをイネーブルにします。

**ステップ 4** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# L2TPv3 over IPv6 トンネルの設定

L2TPv3 over IPv6 トンネルを設定するには、次のタスクを実行します。

## 疑似回線のネイバー AFI の設定

疑似回線のネイバー AFI を設定するには、次の作業を実行します。

**制約事項** L2TPv3 over IPv6 トンネルは、レイヤ 2 トランスポート サブインターフェイスでのみサポートされ、物理インターフェイスではサポートされません。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group***group-name*
4. **p2p***xconnect-name*
5. **interface***type interface-path-id*
6. **neighbor ipv6** X:X::X **pw-id***pseudowire-id*
7. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router (config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 3** **xconnect group***group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group grp_1
```

クロスコネクトグループを設定し、その名前を指定します。

**ステップ 4** **p2p***xconnect-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p vlan1
```

ポイントツーポイント相互接続を設定します。

**ステップ 5** **interface***type interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface GigabitEthernet0/4/0/30
```

インターフェイス タイプ ID を指定します。選択できる基準は、次のとおりです。

- GigabitEthernet：ギガビット イーサネット/IEEE 802.3 インターフェイス
- TenGigE：TenGigabit イーサネット/IEEE 802.3 インターフェイス

**ステップ 6** **neighbor ipv6** X:X::X **pw-id***pseudowire-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor ipv6 1111:2222::cdef pw-id 2000
```

相互接続するピアを指定し、クロスコネクトの疑似回線セグメントを設定します。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## L2TPv3 のカプセル化とプロトコルの設定

L2TPv3 のカプセル化とプロトコルを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **pw-class** *class-name*
4. **encapsulation l2tpv3**
5. **protocol l2tpv3**
6. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router (config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 3** **pw-class** *class-name*

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn)# pw-class l2tpv3_class
```

疑似回線クラスサブモードになり、疑似回線クラステンプレートを定義できます。

次のキーワードは、疑似回線クラス（pw-class）設定モードで設定できますが、これらのキーワードは L2TPv3 over IPv6 トンネルを介した では使用できません。

- **cookie**
- **dfbit**
- **ipv4 source**
- **pmtu**
- **sequencing**
- **transport-mode**

**ステップ 4** **encapsulation l2tpv3**

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-pwc)# encapsulation l2tpv3
```

疑似回線カプセル化を L2TPv3 に設定します。

**ステップ 5** **protocol l2tpv3**

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-pwc-encap-l2tpv3)# protocol l2tpv3
```

疑似回線シグナリングプロトコルを L2TPv3 に設定します。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## L2TPv3 over IPv6 トンネルの送信元 IPv6 アドレスの設定

L2TPv3 over IPv6 トンネルの送信元 IPv6 アドレスを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group** *group-name*

4. **p2p** *xconnect-name*
5. **interface type** *interface-path-id*
6. **neighbor ipv6** *peer-address* **pw-id** *pseudowire-id*
7. **source** *pw-source-address*
8. **end** または **commit**

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/CPU0:router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/CPU0:router (config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 3** **xconnect group** *group-name*

例：

```
RP/0/RSP0/CPU0:router (config-l2vpn)# xconnect group g1
```

クロスコネクトグループを設定します。

**ステップ 4** **p2p** *xconnect-name*

例：

```
RP/0/RSP0/CPU0:router (config-l2vpn-xc)# p2p xc3
```

ポイントツーポイント クロスコネクトを設定します。

**ステップ 5** **interface type** *interface-path-id*

例：

```
RP/0/RSP0/CPU0:router (config-l2vpn-xc-p2p)# interface GigabitEthernet0/0/0/4.2
```

インターフェイス タイプ ID を指定します。

**ステップ 6** **neighbor ipv6** *peer-address* **pw-id** *pseudowire-id*

例：

```
RP/0/RSP0/CPU0:router (config-l2vpn-xc-p2p)# neighbor ipv6 1111:2222::cdef pw-id 1
```

相互接続するピアを指定し、クロスコネクトの疑似回線セグメントを設定します。

**ステップ 7** **source** *pw-source-address*

例：

```
RP/0/RSP0/CPU0:router (config-l2vpn-xc-p2p-pw)# source 1111:2222::abcd
```

疑似回線の送信元 IPv6 アドレスを設定します。

（注） 送信元 IPv6 アドレスは一意であり、任意に選択する必要があります。このアドレスはルータ内のどのタイプのインターフェイスにも設定できません。

**ステップ 8** **end** または **commit**

例：

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw)# end
```

または

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、変更をコミットするように要求されます。

  ```
  Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:
  ```

  - **yes** と入力すると、実行設定ファイルに変更が保存され、設定セッションが終了して、ルータが **EXEC** モードに戻ります。
  - **no** と入力すると、設定セッションが終了して、ルータが **EXEC** モードに戻ります。設定の変更はコミットされません。
  - **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。

- 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

## ローカルおよびリモートセッションの設定

ローカルセッションとリモートセッションを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group** *group-name*
4. **p2p** *xconnect-name*
5. **interface type** *interface-path-id*
6. **neighbor ipv6** *peer-address* **pw-id** *pseudowire-id*
7. **l2tp static local session** *session-id*
8. **l2tp static remote session***session-id*
9. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/CPU0:router (config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 3** **xconnect group** *group-name*

例：

```
RP/0/RSP0/CPU0:router (config-l2vpn)# xconnect group g1
```

クロスコネクトグループを設定します。

**ステップ 4** **p2p** *xconnect-name*

例：

```
RP/0/RSP0/CPU0:router (config-l2vpn-xc)# p2p xc3
```

ポイントツーポイント クロスコネクトを設定します。

**ステップ 5** **interface type** *interface-path-id*

例：

```
RP/0/RSP0/CPU0:router (config-l2vpn-xc-p2p)# interface GigabitEthernet0/0/0/4.2
```

インターフェイス タイプ ID を指定します。

**ステップ 6** **neighbor ipv6** *peer-address* **pw-id** *pseudowire-id*

例：

```
RP/0/RSP0/CPU0:router (config-l2vpn-xc-p2p)# neighbor ipv6 1111:2222::cdef pw-id 1
```

相互接続するピアを指定し、クロスコネクトの疑似回線セグメントを設定します。

**ステップ 7** **l2tp static local session** *session-id*

例：

```
RP/0/RSP0/CPU0:router (config-l2vpn-xc-p2p-pw)# l2tp static local session 1
```

（オプション）L2TP 擬似回線のスタティック ローカル セッションを設定します。

（注） ローカルセッション ID を設定すると、カプセル化解除側の処理の場合、ASR9000 シリーズ ルータはこの ID を無視します。

**ステップ 8** **l2tp static remote session***session-id*

例：

```
RP/0/RSP0/CPU0:router (config-l2vpn-xc-p2p-pw)# l2tp static remote session 1
```

（オプション）L2TP 擬似回線のスタティック リモート セッションを設定します。

（注） 設定されている場合は、リモートセッション値（カプセル化解除側の値）は、カプセル化側の処理に使用され、L2TPv3 ヘッダーのセッション値フィールドの値がプログラムされます。

**ステップ 9** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## ローカルおよびリモート Cookie の設定

ローカルおよびリモート Cookie を設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group** *group-name*
4. **p2p** *xconnect-name*
5. **interface type** *interface-path-id*
6. **neighbor ipv6** *peer-address* **pw-id** *pseudowire-id*
7. **l2tp static local cookie size** *bytes*
8. **l2tp static local cookie size** *bytes*
9. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router (config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 3** **xconnect group** *group-name*

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn)# xconnect group g1
```

クロスコネクトグループを設定します。

**ステップ 4** **p2p** *xconnect-name*

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc)# p2p xc3
```

ポイントツーポイント クロスコネクトを設定します。

**ステップ 5** **interface type** *interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc-p2p)# interface GigabitEthernet0/0/0/4.2
```

インターフェイス タイプ ID を指定します。

**ステップ 6** **neighbor ipv6** *peer-address* **pw-id** *pseudowire-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor ipv6 1111:2222::cdef pw-id 1
```

相互接続するピアを指定し、クロスコネクトの疑似回線セグメントを設定します。

**ステップ 7** **l2tp static local cookie size** *bytes*

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc-p2p-pw)# l2tp static local cookie size 0
```

L2TP 疑似回線のスタティックローカル Cookie サイズ設定を行います。

（注）　ゼロ以外の Cookie サイズの場合、Cookie の値は必須の引数です。

**ステップ 8**　**l2tp static local cookie size** *bytes*

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc-p2p-pw)# l2tp static remote cookie size 0
```

L2TP 疑似回線のスタティックリモート Cookie サイズ設定を行います。

（注）　ゼロ以外の Cookie サイズの場合、Cookie の値は必須の引数です。

**ステップ 9**　**commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## L2TP スタティックサブモードの有効化

L2TP スタティックサブモードを有効にするには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group** *group-name*
4. **p2p***xconnect-name*
5. **interface type** *interface-path-id*
6. **neighbor ipv6** *peer-address* **pw-id** *pseudowire-id*
7. **l2tp static**
8. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1**　**configure**

例：

```
RP/0/RSP0/cpu 0: router # configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2**　**l2vpn**

例：

```
RP/0/RSP0/cpu 0: router (config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 3** **xconnect group** *group-name*

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn)# xconnect group g1
```

クロスコネクトグループを設定します。

**ステップ 4** **p2p***xconnect-name*

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc)# p2p xc3
```

ポイントツーポイント クロスコネクトを設定します。

**ステップ 5** **interface type** *interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc-p2p)# interface GigabitEthernet0/0/0/4.2
```

インターフェイス タイプ ID を指定します。

**ステップ 6** **neighbor ipv6** *peer-address* **pw-id** *pseudowire-id*

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc-p2p)# neighbor ipv6 1111:2222::cdef pw-id
```

相互接続するピアを指定し、クロスコネクトの疑似回線セグメントを設定します。

**ステップ 7** **l2tp static**

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn)#
```

L2TP スタティック設定サブモードを開始します。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## L2TPv3 ヘッダーの TOS リフレクションの有効化

L2TPv3 ヘッダーでタイプオブサービス（TOS）リフレクションを有効にするには、次の作業を実行します。

### 手順の概要

1. configure
2. l2vpn
3. pw-class**class-name**
4. encapsulation l2tpv3
5. protocol l2tpv3
6. neighbor ipv6**peer-address pw-id pseudowire-id**
7. tos {反映 | 値}
8. end または **commit**

### 手順の詳細

**ステップ 1** configure

**例**：

```
RP/0/RSP0/CPU0:router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** l2vpn

**例**：

```
RP/0/RSP0/CPU0:router (config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 3** pw-class**class-name**

**例**：

```
RP/0/RSP0/CPU0:router (config-l2vpn)# pw-class l2tpv3_class
```

疑似回線クラスサブモードになり、疑似回線クラステンプレートを定義できます。

**ステップ 4** encapsulation l2tpv3

**例**：

```
RP/0/RSP0/CPU0:router (config-l2vpn-pwc)# encapsulation l2tpv3
```

疑似回線カプセル化を L2TPv3 に設定します。

**ステップ 5** protocol l2tpv3

**例**：

```
RP/0/RSP0/CPU0:router (config-l2vpn-pwc-encap-l2tpv3)# protocol l2tpv3
```

疑似回線シグナリングプロトコルを L2TPv3 に設定します。

**ステップ 6** neighbor ipv6**peer-address pw-id pseudowire-id**

**例**：

```
RP/0/RSP0/CPU0:router (config-l2vpn-xc-p2p)# neighbor ipv6 1111:2222::cdef pw-id
```

相互接続するピアを指定し、クロスコネクトの疑似回線セグメントを設定します。

**ステップ 7** tos {反映 | 値}

例：

```
RP/0/RSP0/CPU0:router (config-l2vpn-pwc-encap-l2tpv3)# tos reflect
```

または

```
RP/0/RSP0/CPU0:router (config-l2vpn-pwc-encap-l2tpv3)# tos value 50
```

タイプオブサービス（TOS）リフレクションを有効にします。これにより、内部 IP ヘッダーから L2TPv3 ヘッダーに TOS がコピーされます。

また、L2TPv3 疑似回線クラスの TOS の値を設定する場合は、このコマンドを使用します。有効な範囲は 0 ～ 255 です。

**ステップ 8** end または **commit**

例：

```
RP/0/RSP0/CPU0:router(config-l2vpn-pwc-encap-l2tpv3)# end
```

または

```
RP/0/RSP0/CPU0:router(config-l2vpn-pwc-encap-l2tpv3)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、変更をコミットするように要求されます。

  ```
  Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:
  ```

  - **yes** と入力すると、実行設定ファイルに変更が保存され、設定セッションが終了して、ルータが **EXEC** モードに戻ります。
  - **no** と入力すると、設定セッションが終了して、ルータが **EXEC** モードに戻ります。設定の変更はコミットされません。
  - **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

## L2TPv3 over IPv6 トンネルの TTL の設定

L2TPv3 over IPv6 トンネルの存続可能時間（TTL）を設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **pw-class** *class-name*
4. **encapsulation l2tpv3**
5. **protocol l2tpv3**

6. **ttl** *value*
7. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router # configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router (config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 3** **pw-class** *class-name*

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn)# pw-class l2tpv3_class
```

疑似回線クラスサブモードになり、疑似回線クラステンプレートを定義できます。

**ステップ 4** **encapsulation l2tpv3**

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-pwc)# encapsulation l2tpv3
```

疑似回線カプセル化を L2TPv3 に設定します。

**ステップ 5** **protocol l2tpv3**

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-pwc-encap-l2tpv3)# protocol l2tpv3
```

疑似回線シグナリングプロトコルを L2TPv3 に設定します。

**ステップ 6** **ttl** *value*

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-pwc-encap-l2tpv3)# ttl 50
```

ノードホップ内の存続可能時間（TTL）を指定された値に設定します。範囲は 1 ～ 255 です。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。

• [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## L2TPv3 over IPv6 トンネルのトラフィックミラーリングの設定

L2TPv3 over IPv6 トンネルのトラフィックミラーリングを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group***group-name*
4. **p2p** *xconnect-name*
5. **monitor-session** *session-name*
6. **neighbor ipv6** *peer-address* **pw-id** *pseudowire-id*
7. **pw-class** *class-name*
8. **source***pw-source-address*
9. **l2tp static local cookie size***size***value***bytes*
10. **l2tp static remote cookie size***size***value***bytes*
11. **commit** コマンドまたは **end** コマンドを使用します。

- L2TPv3 over IPv6 の概念については、「L2TPv3 over IPv6」を参照してください。
- 設定例については、「L2TPv3 over IPv6 トンネルの設定：例」を参照してください。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router # configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router (config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 3** **xconnect group***group-name*

例：

```
RP/0/RSP0/cpu 0: router (config-l2vpn)# xconnect group span
```

クロスコネクトグループを設定します。

**ステップ 4** **p2p** *xconnect-name*

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc)# p2p span-foo
```

ポイントツーポイント クロスコネクトを設定します。

**ステップ 5** **monitor-session** *session-name*

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc-p2p)# monitor-session customer-foo
```

モニタセッションを指定します。

**ステップ 6** **neighbor ipv6** *peer-address* **pw-id** *pseudowire-id*

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc-p2p)# neighbor ipv6 1111:3333::cdef pw-id 1001
```

相互接続するピアを指定し、クロスコネクトの疑似回線セグメントを設定します。

**ステップ 7** **pw-class** *class-name*

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc-p2p-pw)# pw-class ts
```

疑似回線クラスサブモードになり、疑似回線クラステンプレートを定義できます。

**ステップ 8** **source***pw-source-address*

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc-p2p-pw)# source 1111:3333::abcd
```

疑似回線の送信元 IPv6 アドレスを設定します。

**ステップ 9** **l2tp static local cookie size***size***value***bytes*

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc-p2p-pw)# l2tp static local cookie size 8 value 0xabcd
0x1234
```

L2TP 疑似回線のスタティックローカル Cookie サイズ設定を行います。

**ステップ 10** **l2tp static remote cookie size***size***value***bytes*

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-xc-p2p-pw)# l2tp static remote cookie size 8 value 0xcdef
0x5678
```

L2TP 疑似回線のスタティックリモート Cookie サイズ設定を行います。

**ステップ 11** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

詳細については、次を参照してください。

- L2TPv3 over IPv6 の概念については、「L2TPv3 over IPv6」を参照してください。
- 設定例については、「L2TPv3 over IPv6 トンネルの設定：例」を参照してください。

# L2TPv3 over IPv4 トンネルの設定

**制約事項** L2TPv3 over Ipv4 トンネルは、レイヤ 2 転送サブインターフェイスでのみサポートされ、物理インターフェイスではサポートされません。タグなしのトラフィックを L2TPv3 over IPv4 経由で送信する必要がある場合は、タグなしとしてカプセル化されたサブインターフェイスを作成します。

次に、タグなしとしてカプセル化されたサブインターフェイスを作成する例を示します。

```
interface TenGigE0/3/0/1.123 l2transport
 encapsulation untagged
```

L2TPv3 over IPv4 トンネルを設定するには、次のタスクを実行します。

## ダイナミック L2TPv3 疑似回線の設定

リモート IPv4 ピアに接続するダイナミック L2TPv3 疑似回線を設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group** *name*
4. **p2p** *name*
5. **interface***type interface-path-id*
6. **neighbor ipv4** *ip-address* **pw-id** *number*
7. **pw-class** *pw-class-name*
8. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN 設定サブモードを開始します。

**ステップ 3** **xconnect group** *name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group L2TPV3_V4_XC_GRP
```

クロスコネクトグループの名前を入力します。

**ステップ 4** **p2p** *name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p L2TPV3_P2P_1
```

p2p コンフィギュレーション サブモードを開始して、ポイントツーポイントの相互接続を設定します。

**ステップ 5** **interface***type interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface GigabitEthernet 0/2/0/0/0.1
```

インターフェイス タイプ ID を指定します。選択できる基準は、次のとおりです。

- GigabitEthernet
- TenGigE

**ステップ 6** **neighbor ipv4** *ip-address* **pw-id** *number*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor ipv4 26.26.26.26 pw-id 100
```

相互接続の疑似回線を設定します。

**ステップ 7** **pw-class** *pw-class-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pw)# pw-class L2TPV3_V4_CLASS
```

疑似回線クラスサブモードを開始して、クロスコネクトの名前を定義します。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## L2TPv3 のカプセル化とプロトコルの設定

L2TPv3 のカプセル化とプロトコルを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **pw-class** *class-name*
4. **encapsulation l2tpv3**
5. **protocol l2tpv3**
6. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

**例**：

```
RP/0/RSP0/cpu 0: router (config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 3** **pw-class** *class-name*

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn)# pw-class l2tpv3_class
```

疑似回線クラスサブモードになり、疑似回線クラステンプレートを定義できます。

疑似回線クラス（pw-class）設定モードでは、次のキーワードを設定できます。

- **cookie**
- **dfbit**
- **ipv4 source**

**ステップ 4** **encapsulation l2tpv3**

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-pwc)# encapsulation l2tpv3
```

疑似回線カプセル化を L2TPv3 に設定します。

**ステップ 5** **protocol l2tpv3**

**例**：

```
RP/0/RSP0/cpu 0: router (config-l2vpn-pwc-encap-l2tpv3)# protocol l2tpv3
```

疑似回線シグナリングプロトコルを L2TPv3 に設定します。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# L2TP 制御チャネルパラメータの設定

L2TP 制御チャネルパラメータは、制御チャネル認証、キープアライブメッセージ、および制御チャネル ネゴシエーションで使用されます。L2tpv3 セッションでは、両方の PE ルータで同じ L2TP クラスを設定する必要があります。

次の L2TP 制御チャネルパラメータは、L2TP クラス設定モードで設定できます。

- L2TP 制御チャネルの認証
- L2TP 制御チャネル認証に使用されるパスワード
- 制御メッセージに使用される再送信パラメータ
- 制御チャネルに使用されるタイムアウトパラメータ
- メンテナンスパラメータ
- L2TPv3 コントロール メッセージ ハッシング

他の疑似回線クラスに継承可能な L2TP 制御チャネルパラメータのテンプレートを作成するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2tp-class** *l2tp-class-name*
3. **authentication**
4. **password** {**0** | **7**} *password*
5. **retransmit** { **initial retries** *initial-retries* | **retries** *retries* | **timeout** { **max** | **min** } *timeout* }
6. **hello-interval** *interval*
7. **digest** { **check disable** | **hash** { **MD5** | **SHA1** } ] | **secret** { **0** | **7** } *password* ]
8. **hidden**

## 手順の詳細

### ステップ 1 configure

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

### ステップ 2 l2tp-class *l2tp-class-name*

例：

```
RP/0/RSP0/cpu 0: router(config)# l2tp-class L2TP-CLASS
```

L2TP クラス名を指定して、L2TP クラス コンフィギュレーション モードを開始します。

**ステップ 3** **authentication**

例：

```
RP/0/RSP0/cpu 0: router(config-l2tp-class)# authentication
```

PE ルータ間の制御チャネルの認証を有効にします。

**ステップ 4** **password** {**0** | **7**} *password*

例：

```
RP/0/RSP0/cpu 0: router(config-l2tp-class)# password 7 pwd_1
```

制御チャネル認証に使用されるパスワードを設定します。

- **[0 | 7]**：共有秘密の入力フォーマットを指定します。デフォルト値は 0 です。
  - **0**：暗号化されたパスワードが後に続くことを指定します。
  - **7**：暗号化されていないパスワードが後に続くことを指定します。
- *password*：ピアルータ間の共通パスワードを定義します。

**ステップ 5** **retransmit** { **initial retries** *initial-retries* | **retries** *retries* | **timeout** { **max** | **min** } *timeout* }

例：

```
RP/0/RSP0/cpu 0: router(config-l2tp-class)# retransmit retries 10
```

制御パケットの再送信に影響するパラメータを設定します。

- **initial retries**：セッションが中断される前に再送信する SCCRQ の数を指定します。範囲は 1 ～ 1000 です。デフォルトは 2 です。
- **retries**：ピア PE ルータが無応答であると判断する前に実行する再送信の回数を指定します。範囲は 1 ～ 1000 です。デフォルトは 15 です。
- **timeout** { **max** | **min** }：制御パケット再送信の最大および最小再送信間隔（秒単位）を指定します。値の範囲は 1 ～ 8 です。デフォルトの最大インターバルは 8 です。デフォルトの最小インターバルは 1 です。

**ステップ 6** **hello-interval** *interval*

例：

```
RP/0/RSP0/cpu 0: router(config-l2tp-class)# hello-interval 10
```

L2TP hello パケット間で使用される交換インターバルを秒単位で指定します。

- *interval* 引数の有効な値の範囲は 0 ～ 1000 です。デフォルト値は 60 です。

**ステップ 7** **digest** { **check disable** | **hash** { **MD5** | **SHA1** } ] | **secret** { **0** | **7** } *password* ]

例：

```
RP/0/RSP0/cpu 0: router(config-l2tp-class)# digest hash MD5
```

L2TPv3 制御チャネル認証または整合性チェックを有効にします。

- **secret**：L2TPv3 制御チャネル認証を有効にします。

（注） **digest** コマンドを **secret** キーワードオプションを指定せずに実行した場合は、L2TPv3 整合性チェックが有効になります。

- {**0** | **7**}：共有秘密の入力フォーマットを指定します。デフォルト値は **0** です。
  - **0**：プレーンテキスト秘密が入力されたことを示します。
  - **7**：暗号化された秘密が入力されたことを示します。
- *password*：ピアルータ間の共有秘密を定義します。*password* 引数に入力する値は、{**0** | **7**} キーワードオプションで指定された入力フォーマットに合わせる必要があります。
- **hash** { **MD5** | **SHA1**}：メッセージ単位ダイジェスト計算に使用されるハッシュ関数を指定します。
  - **MD5**：HMAC-MD5 ハッシュ（デフォルト値）を指定します。
  - **SHA1**：HMAC-SHA-1 ハッシュを指定します。

**ステップ 8** **hidden**

例：

```
RP/0/RSP0/cpu 0: router(config-l2tp-class)# hidden
```

L2TPv3 ピアへの制御メッセージの送信時に AVP 隠蔽を有効にします。

# L2VPN 単一セグメント疑似回線の設定

ネットワークで単一セグメント疑似回線を設定するには、次の手順を実行します。

1. （オプション）関連する L2VPN グローバルパラメータの設定。「L2VPN グローバルパラメータの設定」を参照してください

   この手順は、デフォルトの BGP ルート識別子（RD）自動生成値と、BGP の自律システム番号（ASN）およびルート識別子（RID）を上書きするために使用します。

2. L2VPN VPWS SS-PW の設定
3. BGP の L2VPN MS-PW アドレスファミリの設定

アドレスファミリは、ダイナミック擬似回線ルートを交換するために BGP で設定されます。

## L2VPN グローバルパラメータの設定

L2VPN グローバルパラメータを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **router-id** *router-id*
4. **pw-routing**
5. **global-id** *global-id*
6. **bgp**
7. **rd** *route-distinguisher*
8. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 3** **router-id** *router-id*

例：

```
RP/0/RSP0/cpu 0: router(config)# router 2.2.2.2
```

ルータ ID を指定します。

**ステップ 4** **pw-routing**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# pw-routing
```

疑似回線ルーティング機能を有効にし、疑似回線ルーティング設定サブモードを開始します。

**ステップ 5** **global-id** *global-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwr)# global-id 1000
```

ルータの L2VPN グローバル ID 値を設定します。

**ステップ 6** **bgp**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwr)# bgp
```

BGP 疑似回線ルーティング機能を有効にし、BGP 設定サブモードを開始します。

**ステップ 7** **rd** *route-distinguisher*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwr-bgp)# rd 192.168.1.3:10
```

BGP ルート識別子を設定します。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## L2VPN VPWS SS-PW の設定

L2VPN VPWS SS-PW を設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **interface type***interface-path-id*
3. **l2vpn**
4. **xconnect group** *group-name*
5. **p2p** *xconnect-name*

6. **interface** *type interface-path-id*
7. **neighbor routed** *global-id: prefix: ac-id* **source** *ac-id*
8. （オプション） **pw-class** *class-name*
9. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface type***interface-path-id*

例：

```
RP/0/RSP0/cpu 0: routerRP/0/RP00RSP0/CPU0:router# interface TenGigE0/1/0/12
```

インターフェイス コンフィギュレーション モードを開始し、インターフェイスを設定します。

**ステップ 3** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 4** **xconnect group** *group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group pw-he1
```

自由形式の 32 文字ストリングを使用して、相互接続グループ名を設定します。

**ステップ 5** **p2p** *xconnect-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p pw-ss
```

P2P コンフィギュレーション サブモードを開始します。

**ステップ 6** **interface** *type interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface gigabitethernet 0/1/0/9
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 7** **neighbor routed** *global-id: prefix: ac-id* **source** *ac-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor routed 100:2.2.2.2:10 source 10
```

p2p クロスコネクトの疑似回線ルーティング設定サブモードを有効にします。

**ステップ 8** （オプション） **pw-class** *class-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pwr)# pw-class dynamic_sspw
```

疑似回線クラス サブモードになり、疑似回線クラス テンプレートを定義できます。

**ステップ 9** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## BGP の L2VPN MS-PW アドレスファミリの設定

BGP に L2VPN MS-PW アドレスファミリを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **router bgp** *autonomous-system-number*
3. **address-family l2vpn mspw**
4. **neighbor** *ip-address*
5. **address-family l2vpn mspw**
6. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **router bgp** *autonomous-system-number*

例：

```
RP/0/RSP0/cpu 0: router(config)# router bgp 100
```

指定したルーティング プロセスのルータ コンフィギュレーション モードを開始します。

**ステップ 3** **address-family l2vpn mspw**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# address-family l2vpn mspw
```

L2VPN アドレス ファミリを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

**ステップ 4** **neighbor** *ip-address*

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 10.10.10.1
```

指定した自律システム内のネイバーの IP アドレスを追加します。

**ステップ 5** **address-family l2vpn mspw**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# address-family l2vpn mspw
```

ネイバーの L2VPN アドレス ファミリを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## 単一セグメント疑似回線の確認

SS-PW の接続を確認するには、ping **mpls pseudowire** コマンドを使用します。

## L2VPN 単一セグメント疑似回線の情報の表示

show コマンドは、L2VPN 単一セグメント疑似回線の情報を表示するために使用されます

- show bgp l2vpn mspw
- show l2vpn pwr summary
- show l2vpn xc

# EPVN-VPWS の設定方法

EVPN-VPWS を設定するには、次の手順を実行します。

## BGP の L2VPN EVPN アドレス ファミリの設定

BGP に L2VPN EVPN アドレス ファミリを設定するには、このタスクを実行します。

### 手順の概要

1. **configure**
2. **router bgp** *autonomous-system-number*
3. **address-family l2vpn evpn**
4. **neighbor** *ip-address*
5. **address-family l2vpn evpn**
6. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **router bgp** *autonomous-system-number*

例：

```
RP/0/RSP0/cpu 0: router(config)# router bgp 100
```

指定したルーティング プロセスのルータ コンフィギュレーション モードを開始します。

**ステップ 3** **address-family l2vpn evpn**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# address-family l2vpn evpn
```

L2VPN アドレス ファミリを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

**ステップ 4** **neighbor** *ip-address*

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 10.10.10.1
```

指定した自律システム内のネイバーの IP アドレスを追加します。

**ステップ 5** **address-family l2vpn evpn**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# address-family l2vpn evpn
```

ネイバーの L2VPN アドレス ファミリを指定し、アドレス ファミリ コンフィギュレーション モードを開始します。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## EVPN-VPWS の設定

EVPN-VPWS を設定するには、次のタスクを実行します。

（注） PWHE インターフェイスは、EVPN-VPWS を使用しても設定できます。詳細については、疑似回線ヘッドエンドの設定 （324 ページ）モジュールを参照してください。

### 手順の概要

1. **configure**
2. **interface type** *interface-path-id*
3. **l2vpn**
4. **xconnect group** *group-name*

5. **p2p** *xconnect-name*
6. **interface** *type interface-path-id*
7. **neighbor evpn evi** *vpn-id* **target** *ac-id*
8. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface type** *interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config)# interface TenGigE0/1/0/12
```

インターフェイス コンフィギュレーション モードを開始し、インターフェイスを設定します。

**ステップ 3** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 4** **xconnect group** *group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group xc1
```

自由形式の 32 文字ストリングを使用して、相互接続グループ名を設定します。

**ステップ 5** **p2p** *xconnect-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p pw-ss
```

P2P コンフィギュレーション サブモードを開始します。

**ステップ 6** **interface** *type interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface gigabitethernet 0/1/0/9
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 7** **neighbor evpn evi** *vpn-id* **target** *ac-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor evpn evi 100 target 12
```

P2P クロス接続上で EVPN-VPWS エンドポイントを有効にします。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## EVPN-VPWS を使用したアクセス疑似回線の設定

ブリッジドメインは、EVPN VPWS を使用してアクセス疑似回線を設定できます。

### 手順の概要

1. **configure**
2. **interface type** *interface-path-id*
3. **l2vpn**
4. **bridge group** *bridge-group-name*
5. **bridge-domain** *bridge-domain-name*
6. **neighbor evpn evi** *vpn-id* **target** *ac-id*

### 手順の詳細

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | **configure**<br>例：<br>`RP/0/RSP0/cpu 0: router# configure` | グローバル コンフィギュレーション モードを開始します。 |
| ステップ 2 | **interface type** *interface-path-id*<br>例： | インターフェイス コンフィギュレーション モードを開始し、インターフェイスを設定します。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| | `RP/0/RSP0/cpu 0: routerRP/0/RP00RSP0/CPU0:router# interface TenGigE0/1/0/12` | |
| ステップ 3 | **l2vpn**<br>例：<br>`RP/0/RSP0/cpu 0: router(config)# l2vpn` | レイヤ 2 VPN コンフィギュレーション モードを開始します。 |
| ステップ 4 | **bridge group** *bridge-group-name*<br>例：<br>`RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group access-pw` | ブリッジ ドメインを含めることができるブリッジ グループを作成し、ブリッジドメインにネットワーク インターフェイスを割り当てます。 |
| ステップ 5 | **bridge-domain** *bridge-domain-name*<br>例：<br>`RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain bd1` | ブリッジドメインを確立し、L2VPNブリッジグループ ブリッジ ドメイン コンフィギュレーション モードを開始します。 |
| ステップ 6 | **neighbor evpn evi** *vpn-id* **target** *ac-id*<br>例：<br>`RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# neighbor evpn evi 100 target 12` | P2P クロス接続上で EVPN-VPWS エンドポイントを有効にします。 |

例

# ポイントツーポイント レイヤ 2 サービスの設定例

ここで示す設定例は、次のとおりです。

## L2VPN インターフェイスの設定：例

次に、L2VPN インターフェイスを設定する例を示します。

```
configure
 interface GigabitEthernet0/0/0/0.1 l2transport
 encapsulation dot1q 1
 rewrite ingress tag pop 1 symmetric
 end
```

# ローカル スイッチングの設定：例

次に、レイヤ 2 ローカル スイッチングを設定する例を示します。

```
configure
 l2vpn
  xconnect group examples
  p2p example1
  interface TenGigE0/7/0/6.5
  interface GigabitEthernet0/4/0/30
 commit
 end

show l2vpn xconnect group examples
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
        SB = Standby, SR = Standby Ready

XConnect                    Segment 1                   Segment 2
Group      Name       ST   Description            ST   Description            ST
examples   example1   UP   Te0/7/0/6.5            UP   Gi0/4/0/30             UP
```

# ローカル接続冗長性設定：例

次に、PoA1 に LCR を設定する例を示します。

```
! LCR - CE1
  group 107
   mlacp node 1
   mlacp system mac 0001.0001.0107
   mlacp system priority 107
    member
    neighbor 200.0.2.1
   !
   ! LCR - CE2
  group 207
   mlacp node 1
   mlacp system mac 0001.0001.0207
   mlacp system priority 207
    member
    neighbor 200.0.2.1
   !

interface Bundle-Ether107
 description CE5 - LCR
 mlacp iccp-group 107
 mlacp port-priority 10
 no shut

interface Bundle-Ether207
 description CE6 - LCR
 mlacp iccp-group 207
 mlacp port-priority 10
 no shut

interface bundle-e107.1 l2t
  description CE5 - LCR
  encap dot1q 107 second 1
  rewrite ingress tag pop 2 symmetric

interface bundle-e207.1 l2t
  description CE2 - LCR
```

```
  encap dot1q 207 second 1
  rewrite ingress tag pop 2 symmetric

interface bundle-e307.1 l2t
  description PE2 - LCR
  encap dot1q 1
  rewrite ingress tag pop 1 symmetric

l2vpn
 xconnect group lcr-scale
  p2p lcr-1
   interface bundle-e107.1
   interface bundle-e207.1
   backup interface bundle-e307.1
```

# ポイントツーポイント相互接続の設定：例

ここでは、スタティックおよびダイナミック p2p 相互接続の設定例を示します。

### スタティック設定

次に、スタティック ポイントツーポイント相互接続の設定例を示します。

```
configure
  l2vpn
   xconnect group vlan_grp_1
    p2p vlan1
    interface GigabitEthernet0/0/0/0.1
    neighbor 102.2.12.1 2 pw-id 1
  mpls static label local 699 remote 890
commit2000
```

### ダイナミック設定

次に、ダイナミック ポイントツーポイント相互接続の設定例を示します。

```
configure
  l2vpn
   xconnect group vlan_grp_1
    p2p vlan1
    interface TenGigE 0/0/0/0.1
    neighbor 2.2.1.1 pw-id 1
commit
```

# Inter-AS：例

次に、AC1 から AC2 への AC 間相互接続の設定例を示します。

```
router-id Loopback0

interface Loopback0
ipv4 address 10.0.0.5 255.255.255.255
!
interface GigabitEthernet0/1/0/0.1 l2transport
encapsulation dot1q 1
!
!
interface GigabitEthernet0/0/0/3
ipv4 address 10.45.0.5 255.255.255.0
```

```
keepalive disable
!
interface GigabitEthernet0/0/0/4
ipv4 address 10.5.0.5 255.255.255.0
keepalive disable
!
router ospf 100
log adjacency changes detail
area 0
interface Loopback0
!
interface GigabitEthernet0/0/0/3
!
interface GigabitEthernet0/0/0/4
!
!
!
router bgp 100
address-family ipv4 unicast
allocate-label all
!
neighbor 10.2.0.5
remote-as 100
update-source Loopback0
address-family ipv4 unicast
!
address-family ipv4 labeled-unicast
!
!
!
l2vpn
xconnect group cisco
p2p cisco1
interface GigabitEthernet0/1/0/0.1
neighbor 10.0.1.5 pw-id 101
!
p2p cisco2
interface GigabitEthernet0/1/0/0.2
neighbor 10.0.1.5 pw-id 102
!
p2p cisco3
interface GigabitEthernet0/1/0/0.3
neighbor 10.0.1.5 pw-id 103
!
p2p cisco4
interface GigabitEthernet0/1/0/0.4
neighbor 10.0.1.5 pw-id 104
!
p2p cisco5
interface GigabitEthernet0/1/0/0.5
neighbor 10.0.1.5 pw-id 105
!
p2p cisco6
interface GigabitEthernet0/1/0/0.6
neighbor 10.0.1.5 pw-id 106
!
p2p cisco7
interface GigabitEthernet0/1/0/0.7
neighbor 10.0.1.5 pw-id 107
!
p2p cisco8
interface GigabitEthernet0/1/0/0.8
neighbor 10.0.1.5 pw-id 108
!
```

```
p2p cisco9
interface GigabitEthernet0/1/0/0.9
neighbor 10.0.1.5 pw-id 109
!
p2p cisco10
interface GigabitEthernet0/1/0/0.10
neighbor 10.0.1.5 pw-id 110
!
!
!
mpls ldp
router-id Loopback0
log
neighbor
!
interface GigabitEthernet0/0/0/3
!
interface GigabitEthernet0/0/0/4
!
!
end
```

# L2VPN Quality of Service：例

次に、ポート モードの L2 インターフェイスにサービス ポリシーをアタッチする例を示します。

```
configure
  interface GigabitEthernet 0/0/0/0
  l2transport
  service-policy input pmap_1
commit
```

# 疑似回線：例

例には、次のデバイスおよび接続が含まれます。

- T-PE1 ノードには次の項目があります。
  - AC インターフェイスとの相互接続（CE1 方向）
  - S-PE1 ノードへの疑似回線
  - IP アドレス：209.165.200.225
- T-PE2 ノード
  - AC インターフェイスとの相互接続（CE2 方向）
  - S-PE1 ノードへの疑似回線
  - IP アドレス：209.165.200.254
- S-PE1 ノード
  - T-PE1 ノードへの疑似回線セグメントによるマルチセグメント疑似回線相互接続
  - T-PE2 ノードへの疑似回線セグメント
  - IP アドレス：209.165.202.158

## T-PE1 ノードのダイナミック疑似回線の設定：例

```
RP/0/RSP0/CPU0:T-PE1# configure
RP/0/RSP0/CPU0:T-PE1(config)# l2vpn
RP/0/RSP0/CPU0:T-PE1 (config-l2vpn)# pw-class dynamic_mpls
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-pwc)# encapsulation mpls
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-pwc-encap-mpls)# protocol ldp
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-pwc-encap-mpls)# control-word disable
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-pwc-encap-mpls)# exit
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-pwc)# exit
RP/0/RSP0/CPU0:T-PE1(config-l2vpn)# xconnect group XCON1
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-xc)# p2p xc1
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-xc-p2p)# description T-PE1 MS-PW to 10.165.202.158
via 10.165.200.254
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-xc-p2p)# interface gigabitethernet 0/1/0/0.1
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-xc-p2p)# neighbor 10.165.200.254 pw-id 100
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-xc-p2p-pw)# pw-class dynamic_mpls
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-xc-p2p-pw)# commit
```

## S-PE1 ノードのダイナミック疑似回線の設定：例

```
RP/0/RSP0/CPU0:S-PE1# configure
RP/0/RSP0/CPU0:S-PE1(config)# l2vpn
RP/0/RSP0/CPU0:S-PE1(config-l2vpn)# pw-class dynamic_mpls
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc)# encapsulation mpls
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc-encap-mpls)# protocol ldp
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc-encap-mpls)# control-word disable
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc-encap-mpls)# exit
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc)# exit
RP/0/RSP0/CPU0:S-PE1(config-l2vpn)# xconnect group MS-PW1
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc)# p2p ms-pw1
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p)# description S-PE1 MS-PW between 10.165.200.225
 and 10.165.202.158
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p)# neighbor 10.165.200.225 pw-id 100
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p-pw)# pw-class dynamic_mpls
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p-pw)# exit
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p)# neighbor 10.165.202.158 pw-id 300
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p-pw)# pw-class dynamic_mpls
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p-pw)# commit
```

## T-PE2 ノードのダイナミック疑似回線の設定：例

```
RP/0/RSP0/CPU0:T-PE2# configure
RP/0/RSP0/CPU0:T-PE2(config)# l2vpn
RP/0/RSP0/CPU0:T-PE2 (config-l2vpn)# pw-class dynamic_mpls
RP/0/RSP0/CPU0:T-PE2 (config-l2vpn-pwc)# encapsulation mpls
RP/0/RSP0/CPU0:T-PE2 (config-l2vpn-pwc-encap-mpls)# protocol ldp
RP/0/RSP0/CPU0:T-PE2 (config-l2vpn-pwc-encap-mpls)# control-word disable
RP/0/RSP0/CPU0:T-PE2 (config-l2vpn-pwc-encap-mpls)# exit
RP/0/RSP0/CPU0:T-PE2 (config-l2vpn-pwc)# exit
RP/0/RSP0/CPU0:T-PE2(config-l2vpn)# xconnect group XCON1
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-xc)# p2p xc1
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-xc-p2p)# description T-PE2 MS-PW to 10.165.200.225 via
 10.165.200.254
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-xc-p2p)# interface gigabitethernet 0/2/0/0.4
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-xc-p2p)# neighbor 10.165.200.254 pw-id 300
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-xc-p2p-pw)# pw-class dynamic_mpls
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-xc-p2p-pw)# commit
```

## T-PE1 ノードのダイナミック疑似回線と優先パスの設定：例

```
RP/0/RSP0/CPU0:T-PE1# configure
RP/0/RSP0/CPU0:T-PE1(config)# l2vpn
RP/0/RSP0/CPU0:T-PE1(config-l2vpn)# pw-class dynamic_mpls
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-pwc)# encapsulation mpls
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-pwc-encap-mpls)# protocol ldp
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-pwc-encap-mpls)# control-word disable
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-pwc-encap-mpls)# preferred-path interface tunnel-te
1000
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-pwc-encap-mpls)# exit
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-pwc)# exit
RP/0/RSP0/CPU0:T-PE1(config-l2vpn)# xconnect group XCON1
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-xc)# p2p xc1
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-xc-p2p)# description T-PE1 MS-PW to 10.165.202.158
via 10.165.200.254
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-xc-p2p)# interface gigabitethernet 0/1/0/0.1
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-xc-p2p)# neighbor 10.165.200.254 pw-id 100
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-xc-p2p-pw)# pw-class dynamic_mpls
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-xc-p2p-pw)# commit
```

## S-PE1 ノードのダイナミック疑似回線と優先パスの設定：例

```
RP/0/RSP0/CPU0:S-PE1# configure
RP/0/RSP0/CPU0:S-PE1(config)# l2vpn
RP/0/RSP0/CPU0:S-PE1(config-l2vpn)# pw-class dynamic_mpls1
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc)# encapsulation mpls
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc-encap-mpls)# protocol ldp
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc-encap-mpls)# control-word disable
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc-encap-mpls)# preferred-path interface tunnel-te
1000
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc-encap-mpls)# exit
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc)# exit
RP/0/RSP0/CPU0:S-PE1(config-l2vpn)# pw-class dynamic_mpls2
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc)# encapsulation mpls
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc-encap-mpls)# protocol ldp
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc-encap-mpls)# control-word disable
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc-encap-mpls)# preferred-path interface tunnel-te
2000
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc-encap-mpls)# exit
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc)# exit
RP/0/RSP0/CPU0:S-PE1(config-l2vpn)# xconnect group MS-PW1
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc)# p2p ms-pw1
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p)# description S-PE1 MS-PW between 10.165.200.225
 and 10.165.202.158
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p)# neighbor 10.165.200.225 pw-id 100
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p-pw)# pw-class dynamic_mpls1
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p-pw)# exit
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p)# neighbor 10.165.202.158 pw-id 300
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p-pw)# pw-class dynamic_mpls2
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p-pw)# commit
```

## T-PE2 ノードのダイナミック疑似回線と優先パスの設定：例

```
RP/0/RSP0/CPU0:T-PE2# configure
RP/0/RSP0/CPU0:T-PE2(config)# l2vpn
RP/0/RSP0/CPU0:T-PE2(config-l2vpn)# pw-class dynamic_mpls
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-pwc)# encapsulation mpls
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-pwc-encap-mpls)# protocol ldp
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-pwc-encap-mpls)# control-word disable
```

```
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-pwc-encap-mpls)# preferred-path interface tunnel-te
2000
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-pwc-encap-mpls)# exit
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-pwc)# exit
RP/0/RSP0/CPU0:T-PE2(config-l2vpn)# xconnect group XCON1
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-xc)# p2p xc1
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-xc-p2p)# description T-PE2 MS-PW to 10.165.200.225 via
 10.165.200.254
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-xc-p2p)# interface gigabitethernet 0/2/0/0.4
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-xc-p2p)# neighbor 10.165.200.254 pw-id 300
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-xc-p2p-pw)# pw-class dynamic_mpls
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-xc-p2p-pw)# commit
```

## T-PE1 ノードのスタティック疑似回線の設定：例

```
RP/0/RSP0/CPU0:T-PE1# configure
RP/0/RSP0/CPU0:T-PE1(config)# l2vpn
RP/0/RSP0/CPU0:T-PE1(config-l2vpn)# xconnect group XCON1
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-xc)# p2p xc1
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-xc-p2p)# interface gigabitethernet 0/1/0/0.1
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-xc-p2p)# neighbor 10.165.200.254 pw-id 100
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-xc-p2p-pw)# mpls static label local 50 remote 400
RP/0/RSP0/CPU0:T-PE1(config-l2vpn-xc-p2p-pw)# commit
```

## S-PE1 ノードのスタティック疑似回線の設定：例

```
RP/0/RSP0/CPU0:S-PE1# configure
RP/0/RSP0/CPU0:S-PE1(config)# l2vpn
RP/0/RSP0/CPU0:S-PE1(config-l2vpn)# xconnect group MS-PW1
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc)# p2p ms-pw1
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p)# neighbor 10.165.200.225 pw-id 100
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p-pw)# mpls static label local 400 remote 50
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p-pw)# exit
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p)# neighbor 10.165.202.158 pw-id 300
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p-pw)# mpls static label local 40 remote 500
RP/0/RSP0/CPU0:S-PE1(config-l2vpn-xc-p2p-pw)# commit
```

## T-PE2 ノードのスタティック疑似回線の設定：例

```
RP/0/RSP0/CPU0:T-PE2# configure
RP/0/RSP0/CPU0:T-PE2(config)# l2vpn
RP/0/RSP0/CPU0:T-PE2(config-l2vpn)# xconnect group XCON1
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-xc)# p2p xc1
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-xc-p2p)# interface gigabitethernet 0/2/0/0.4
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-xc-p2p)# neighbor 10.165.200.254 pw-id 300
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-xc-p2p-pw)# mpls static label local 500 remote 40
RP/0/RSP0/CPU0:T-PE2(config-l2vpn-xc-p2p-pw)# commit
```

# 優先パス：例

次に、優先トンネル パスを設定する例を示します。

```
configure
l2vpn
pw-class path1
encapsulation mpls
preferred-path interface  tunnel tp 50 fallback disable
```

# MPLS トランスポート プロファイル：例

ここでは、次の例を示します。

- 優先トンネル パスの設定：例
- PW ステータス OAM の設定：例

## 優先トンネル パスの設定：例

この設定例では、優先トンネル パスを設定する方法を示します。

```
l2vpn
 pw-class foo
  encapsulation mpls
  preferred-path interface tunnel-tp 100 fallback disable
 commit
```

## PW ステータス OAM の設定：例

この設定例では、PW ステータス OAM 機能を設定する方法を示します。

```
l2vpn
 pw-oam refresh transmit 100
 commit
```

# 疑似回線ステータスの表示：例

## show l2vpn xconnect

```
RP/0/RSP0/CPU0:router# show l2vpn xconnect
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
        LU = Local Up, RU = Remote Up, CO = Connected

XConnect                   Segment 1                   Segment 2
Group      Name       ST   Description            ST   Description            ST
------------------------   ------------------------   ------------------------
MS-PW1     ms-pw1     UP   70.70.70.70     100    UP   90.90.90.90     300    UP
------------------------------------------------------------------------------
```

## show l2vpn xconnect detail

```
RP/0/RSP0/CPU0:router# show l2vpn xconnect detail
Group MS-PW1, XC ms-pw1, state is up; Interworking none
  PW: neighbor 70.70.70.70, PW ID 100, state is up ( established )
    PW class not set
    Encapsulation MPLS, protocol LDP
    PW type Ethernet VLAN, control word enabled, interworking none
    PW backup disable delay 0 sec
    Sequencing not set
    PW Status TLV in use
        MPLS         Local                          Remote
      ------------ ------------------------------ ------------------------------
      Label        16004                          16006
      Group ID     0x2000400                      0x2000700
      Interface    GigabitEthernet0/1/0/2.2       GigabitEthernet0/1/0/0.3
      MTU          1500                           1500
```

```
    Control word enabled                        enabled
    PW type      Ethernet VLAN                  Ethernet VLAN
    VCCV CV type 0x2                            0x2
                 (LSP ping verification)        (LSP ping verification)
    VCCV CC type 0x5                            0x7
                 (control word)                 (control word)
                                                (router alert label)
                 (TTL expiry)                   (TTL expiry)
    ------------ ------------------------------ ------------------------------

  Incoming Status (PW Status TLV):
    Status code: 0x0 (Up) in Notification message
  Outgoing PW Switching TLVs (Label Mapping message):
    Local IP Address: 80.80.80.80, Remote IP address: 90.90.90.90, PW ID: 300
    Description: S-PE1 MS-PW between 70.70.70.70 and 90.90.90.90
  Outgoing Status (PW Status TLV):
    Status code: 0x0 (Up) in Notification message
  Statistics:
    packet totals: receive 0
    byte totals: receive 0
  Create time: 04/04/2008 23:18:24 (00:01:24 ago)
  Last time status changed: 04/04/2008 23:19:30 (00:00:18 ago)
PW: neighbor 90.90.90.90, PW ID 300, state is up ( established )
  PW class not set
  Encapsulation MPLS, protocol LDP
  PW type Ethernet VLAN, control word enabled, interworking none
  PW backup disable delay 0 sec
  Sequencing not set
  PW Status TLV in use
      MPLS         Local                          Remote
    ------------ ------------------------------ ------------------------------
    Label        16004                          16006
    Group ID     0x2000800                      0x2000200
    Interface    GigabitEthernet0/1/0/0.3       GigabitEthernet0/1/0/2.2
    MTU          1500                           1500
    Control word enabled                        enabled
    PW type      Ethernet VLAN                  Ethernet VLAN
    VCCV CV type 0x2                            0x2
                 (LSP ping verification)        (LSP ping verification)
    VCCV CC type 0x5                            0x7
                 (control word)                 (control word)
                                                (router alert label)
                 (TTL expiry)                   (TTL expiry)
    ------------ ------------------------------ ------------------------------
  Incoming Status (PW Status TLV):
    Status code: 0x0 (Up) in Notification message
  Outgoing PW Switching TLVs (Label Mapping message):
    Local IP Address: 80.80.80.80, Remote IP address: 70.70.70.70, PW ID: 100
    Description: S-PE1 MS-PW between 70.70.70.70 and 90.90.90.90
  Outgoing Status (PW Status TLV):
    Status code: 0x0 (Up) in Notification message
  Statistics:
    packet totals: receive 0
    byte totals: receive 0
  Create time: 04/04/2008 23:18:24 (00:01:24 ago)
  Last time status changed: 04/04/2008 23:19:30 (00:00:18 ago)
```

# Any Transport over MPLS（AToM）の設定：例

次に、Any Transport over MPLS（AToM）を設定する例を示します。

```
config
l2vpn
 xconnect group test
 p2p test
 interface POS 0/1/0/0.1
  neighbor 10.1.1.1 pw-id 100
```

# AToM IP インターワーキングの設定：例

次に、IP インターワーキングを設定する例を示します。

```
config
l2vpn
 xconnect group test
 p2p test
 interworking ipv4
```

# PPP IP インターワーキングの設定：例

次に、PPP IP インターワーキングを設定する例を示します。

```
interface Serial0/2/1/0/1/1/1:0
 encapsulation ppp
l2transport
 !
!
interface Serial0/0/0/0/2/1/1:0
encapsulation ppp
 l2transport
 !
!

!! Local Switching Configuration
l2vpn
xconnect group ppp_ip_ls
  p2p 1
   interface Serial0/2/1/0/1/1/1:0
   interface GigabitEthernet0/0/0/1.1
   interworking ipv4
  !

!! PW Configuration
l2vpn
 xconnect group ppp_ip_iw
  p2p 1
   interface Serial0/0/0/0/2/1/1:0
   neighbor 120.120.120.120 pw-id 3
    pw-class class1
   !
   interworking ipv4
```

# cHDLC IP インターワーキングの設定：例

次に、cHDLC IP インターワーキングを設定する例を示します。

```
interface Serial0/2/1/0/1/1/2:0
l2transport

interface Serial0/0/0/0/2/1/2:0
```

```
l2transport


!! Local Switching Configuration
l2vpn
xconnect group ppp_ip_ls
  p2p 1
   interface Serial0/2/1/0/1/1/2:0
   interface GigabitEthernet0/0/0/2.1
   interworking ipv4
  !

!! PW Configuration
l2vpn
 xconnect group ppp_ip_iw
  p2p 1
   interface Serial0/0/0/0/2/1/2:0
   neighbor 120.120.120.120 pw-id 3
    pw-class class1
   !
   interworking ipv4
```

# MLPPP IP インターワーキングの設定：例

次に、MLPPP IP インターワーキングを設定する例を示します。

```
interface Multilink0/2/1/0/1
 multilink
l2transport
!

interface Multilink0/2/1/0/51
Multilink
l2transport


!! Local Switching Configuration
l2vpn
 xconnect group mlppp_ip_ls
  p2p 1
   interface Multilink0/2/1/0/1
   interface GigabitEthernet0/0/0/1.151
   interworking ipv4
!
!! PW Configuration
l2vpn
xconnect group mlppp_ip_iw
  p2p 151
   interface Multilink0/2/1/0/51
   neighbor 140.140.140.140 pw-id 151
    pw-class test
   !
   interworking ipv4
  !
```

# Circuit Emulation over Packet Switched Network の設定：例

次に、Circuit Emulation Over Packet Switched Network を設定する例を示します。

### CEM 接続回線の PW への追加

```
l2vpn
 xconnect group gr1
  p2p p1
   interface CEM 0/0/0/0:10
   neighbor 3.3.3.3 pw-id 11
   !
  !
```

### 疑似回線クラスの関連付け

```
l2vpn
 pw-class class-cem
  encapsulation mpls
   protocol ldp
  !
 !
xconnect group gr1
  p2p p1
   interface CEM0/0/0/0:20
   neighbor 1.2.3.4 pw-id 11
    pw-class class-cem
   !
```

### 疑似回線ステータスのイネーブル化

```
l2vpn
  pw-status
  commit
```

### 疑似回線ステータスのディセーブル化

```
l2vpn
  pw-status disable
  commit
```

### バックアップ疑似回線の設定

```
l2vpn
 pw-status
 pw-class class-cem
  encapsulation mpls
   protocol ldp
  !
 !
 xconnect group gr1
  p2p p1
   interface CEM0/0/0/0:20
   neighbor 1.2.3.4 pw-id 11
    pw-class class-cem
    backup neighbor 9.9.9.9 pw-id 1221
     pw-class class-cem
    !
   !
```

# L2VPN ノンストップ ルーティングの設定：例

次に、L2VPN ノンストップ ルーティングを設定する例を示します。

```
config
l2vpn
 nsr
 logging nsr
```

# 疑似回線のグループ化のイネーブル化：例

次に、疑似回線のグループ化をイネーブルにする例を示します。

```
config
l2vpn
 pw-grouping
```

# L2TPv3 over IPv6 トンネルの設定：例

ここでは、次の例を示します。

## 疑似回線のネイバー AFI の設定：例

IPv6 疑似回線ネイバーをサポートするには、AFI を次のように設定する必要があります。

```
l2vpn
xconnect group g1
p2p xc3
interface GigabitEthernet0/0/0/4.2
neighbor ipv6 1111:2222::cdef pw-id 1
```

## L2TPv3 のカプセル化とプロトコルの設定：例

L2TPv3 トンネルの場合、カプセル化とプロトコルを L2TPv3 に設定する必要があります。

（注） デフォルトのカプセル化とプロトコルは MPLS です。

```
l2vpn
pw-class ts
encapsulation l2tpv3
protocol l2tpv3
```

## L2TPv3 over IPv6 トンネルの送信元 IPv6 アドレスの設定：例

次に、L2TPv3 over IPv6 トンネルの送信元 IPv6 アドレスを設定する例を示します。

```
l2vpn
xconnect group g1
p2p xc3
interface GigabitEthernet0/0/0/4.2
neighbor ipv6 1111:2222::cdef pw-id 1
source 1111:2222::abcd
```

## ローカルおよびリモートセッションの設定：例

L2TPv3 over IPv6 トンネルの場合、ローカルおよびリモートセッション ID は疑似回線で設定されます。ただし、この設定はオプションです。

```
l2vpn
xconnect group g1
p2p xc3
interface GigabitEthernet0/0/0/4.2
neighbor ipv6 1111:2222::cdef pw-id 1
l2tp static local session 1
l2tp static remote session 1
```

## ローカルおよびリモート Cookie の設定：例

L2TPv3 over IPv6 トンネルの場合、ローカルおよびリモート Cookie は疑似回線で設定されます。Cookie ロールオーバーのサポートが拡張され、セカンダリローカル Cookie を設定できるようになりました。次に、サイズ 0 の Cookie を設定する例を示します。

```
l2vpn
xconnect group g1
p2p xc3
interface GigabitEthernet0/0/0/4.2
neighbor ipv6 1111:2222::cdef pw-id 1
l2tp static local cookie size 0
l2tp static remote cookie size 0
```

次に、サイズ 4 の Cookie を設定する例を示します。

```
l2vpn
xconnect group g1
p2p xc3
interface GigabitEthernet0/0/0/4.2
neighbor ipv6 1111:2222::cdef pw-id 1
l2tp static local cookie size 4 value <0x0-0xffffffff>
l2tp static remote cookie size 4 value <0x0-0xffffffff>
```

次に、サイズ 8 の Cookie を設定する例を示します（下位 4 バイトが最初に入力され、その後に上位 4 バイトが続きます）。

```
l2vpn
xconnect group g1
p2p xc3
interface GigabitEthernet0/0/0/4.2
neighbor ipv6 1111:2222::cdef pw-id 1
l2tp static local cookie size 8 value <0x0-0xffffffff> <0x0-0xffffffff>
l2tp static remote cookie size 8 value <0x0-0xffffffff> <0x0-0xffffffff>
```

L2TPv3 over IPv6 トンネルで Cookie ロールオーバーをサポートするには、セカンダリローカル Cookie を設定します。local cookie secondary コマンドは、ローカルルータのセカンダリ Cookie 値を指定します。

（注） プライマリおよびセカンダリ Cookie は同じサイズに設定する必要があります。プライマリまたはセカンダリローカル Cookie は、リモートエンドから受信する Cookie 値と一致する必要があります。そうでない場合、パケットはドロップされます。

```
l2vpn
xconnect group g1
p2p xc3
interface GigabitEthernet0/0/0/4.2
neighbor ipv6 1111:2222::cdef pw-id 1
l2tp static local cookie secondary size 8 value <0x0-0xffffffff> <0x0-0xffffffff>
```

## L2TP スタティックサブモードの有効化：例

次に、L2TP スタティックサブモードを有効にする例を示します。

```
l2vpn
xconnect group g1
p2p xc3
interface GigabitEthernet0/0/0/4.2
neighbor ipv6 1111:2222::cdef pw-id 1
l2tp static
local cookie <>
```

## L2TPv3 ヘッダーの TOS リフレクションの有効化：例

L2TPv3 over IPv6 トンネルの場合、タイプオブサービス（TOS）リフレクションの有効化や、L2TPv3 ヘッダーへの特定の TOS 値の設定が各疑似回線クラスに対してサポートされます。

（注） デフォルトでは、TOS は VLAN ヘッダーのサービスクラス（COS）フィールドからコピーされます。基本となるパケットが IPv4 または IPv6 パケットでない場合、TOS リフレクションが設定されている場合でも、COS フィールドは VLAN ヘッダーからコピーされます。

次に、L2TPv3 ヘッダーに TOS リフレクションを設定する例を示します。

```
l2vpn
pw-class ts
encapsulation l2tpv3
protocol l2tpv3
tos reflect

This example shows how to set a TOS value in the L2TPv3 header:
l2vpn
pw-class ts
encapsulation l2tpv3
protocol l2tpv3
tos value 64
```

## L2TPv3 over IPv6 トンネルの TTL の設定：例

L2TPv3 over IPv6 トンネルの場合、疑似回線クラスで TTL 設定がサポートされます。

```
l2vpn
pw-class ts
encapsulation l2tpv3
protocol l2tpv3
ttl <1-255>
```

## L2TPv3 over IPv6 トンネルのトラフィックミラーリングの設定：例

次に、EFP をモニタセッションに関連付ける例を示します。

```
interface GigabitEthernet0/0/0/4.2 l2transport
monitor-session customer-foo
```

レイヤ 2 SPAN は L3 インターフェイスでサポートされています。ただし、レイヤ 2 フレームはミラーリングされます。

```
interface GigabitEthernet0/0/0/4.2
ipv6 address <>
monitor-session customer-foo
```

SPAN はメインインターフェイスでもサポートされています。

```
interface GigabitEthernet0/4/0/3
l2transport
monitor-session customer-foo
```

次に、モニタセッションをグローバルに作成する例を示します。

```
monitor-session customer-foo
destination pseudowire
```

次に、モニタセッションと L2TPv3 over IPv6 トンネルとの間にクロスコネクトを作成する例を示します。

```
l2vpn
xconnect group span
p2p span-foo
monitor-session customer-foo
neighbor ipv6 1111:3333::cdef pw-id 1001
pw-class ts
source 1111:3333::abcd
l2tp static local cookie size 8 value 0xabcd 0x1234
l2tp static remote cookie size 8 value 0xcdef 0x5678
```

詳細については、次を参照してください。

- L2TPv3 over IPv6 トンネルの概念については、「L2TPv3 over IPv6」を参照してください
- 設定手順については、「L2TPv3 over IPv6 トンネルの設定」を参照してください

# L2TPv3 over IPv4 トンネルの設定：例

ここでは、次の例を示します。

## ダイナミック L2TPv3 疑似回線の設定

リモート IPv4 ピアに接続するダイナミック L2TPv3 疑似回線を設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group** *name*
4. **p2p** *name*
5. **interface***type interface-path-id*
6. **neighbor ipv4** *ip-address* **pw-id** *number*
7. **pw-class** *pw-class-name*
8. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN 設定サブモードを開始します。

**ステップ 3** **xconnect group** *name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group L2TPV3_V4_XC_GRP
```

クロスコネクトグループの名前を入力します。

**ステップ 4** **p2p** *name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p L2TPV3_P2P_1
```

p2p コンフィギュレーション サブモードを開始して、ポイントツーポイントの相互接続を設定します。

**ステップ 5** **interface***type interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface GigabitEthernet 0/2/0/0/0.1
```

インターフェイス タイプ ID を指定します。選択できる基準は、次のとおりです。

- GigabitEthernet
- TenGigE

**ステップ 6** **neighbor ipv4** *ip-address* **pw-id** *number*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor ipv4 26.26.26.26 pw-id 100
```

相互接続の疑似回線を設定します。

**ステップ 7** **pw-class** *pw-class-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pw)# pw-class L2TPV3_V4_CLASS
```

疑似回線クラスサブモードを開始して、クロスコネクトの名前を定義します。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## L2TPv3 のカプセル化とプロトコルの設定：例

次に、L2TPv3 トンネルのカプセル化とプロトコルを設定する例を示します。

```
configure
l2vpn
  pw-class L2TPV3_V4_CLASS
   encapsulation l2tpv3
    protocol l2tpv3 class L2TP-CLASS
    dfbit set
    ipv4 source 25.25.25.25
    cookie size 4
  !
!
```

## L2TP 制御チャネルパラメータの設定：例

次の例は、一般的な L2TPv3 制御チャネル設定を示しています。

```
configure
l2tp-class L2TP-CLASS
 authentication
 retransmit retries 5
 retransmit initial retries 10
 retransmit initial timeout max 5
 retransmit timeout max 6
 hidden
 password 7 1511021F07257A767B
 hello-interval 10
 digest hash MD5
!
```

# EVPN-VPWS の設定例

## EVPN-VPWS の設定：例

次に、EVPN-VPWS サービスを設定する例を示します。

```
RP/0/RSP0/cpu 0: router# configure
```

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group pw-he1
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p pw-ss
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface gigabitethernet 0/1/0/9
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor evpn evi 100 target 12 source 10
```

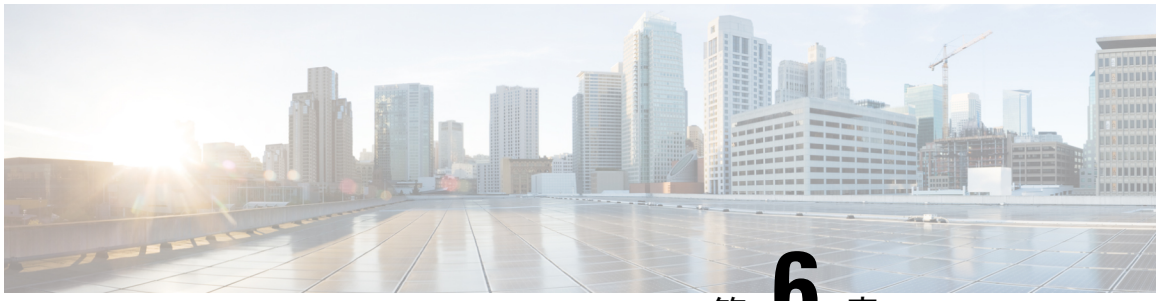
次に、PWHE インターフェイスへの EVPN-VPWS を設定する例を示します。

```
RP/0/RSP0/cpu 0: router# configure
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group xg1
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p pwhe1
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface PW-Ether 1
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor evpn evi 2 target 20 source 20
```

## EVPN-VPWS を使用したアクセス PW の設定：例

次の例は、ブリッジドメインが EVPN-VPWS を使用してアクセス擬似回線を設定する方法を示しています。

```
RP/0/RSP0/cpu 0: router# configure
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group bg1
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain bd1
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# neighbor evpn evi 1 target 100
```



第 6 章

# マルチポイント レイヤ 2 サービスの実装

このモジュールでは、マルチポイント レイヤ 2 ブリッジング サービス（バーチャル プライベート LAN サービス（VPLS）とも呼ばれます）の概念および設定情報を示します。

（注） VPLS は、レイヤ 2 VPN テクノロジーをサポートし、カスタマーにトランスペアレントなマルチポイント レイヤ 2 接続を提供します。このアプローチにより、サービス プロバイダーはブロードキャスト TV やレイヤ 2 VPN といった数多くの新しいサービスをホストすることができます。

ポイントツーポイントレイヤ 2 サービスについては、「ポイントツーポイントレイヤ *2* サービスの実装」の章を参照してください。

このモジュール内に記載されているコマンドの詳細については、「関連ドキュメント」セクションを参照してください。設定作業の実行中に必要になることのある他のコマンドのドキュメントを見つけるには、Cisco IOS XR ソフトウェア マスター コマンド インデックスで、オンライン検索してください。

**マルチポイントレイヤ 2 サービスの実装機能の履歴**

| リリース | 変更内容 |
|---|---|
| リリース 3.7.2 | この機能が導入されました。 |
| リリース 3.9.0 | 次の機能が追加されました。<br>• 不明なユニキャスト フラッディングのブロック。<br>• MAC フラッシュのディセーブル化。<br>• マルチ スパニングツリー アクセス ゲートウェイ<br>• スケール拡張機能が導入されました。スケール拡張機能の詳細については、表 1 を参照してください。 |

| リリース | 変更内容 |
|---|---|
| リリース 3.9.1 | BGP オートディスカバリおよびLDP シグナリングによる VPLS のサポートが追加されました。 |
| リリース 4.0.1 | 次の機能に対するサポートが追加されました。<br>• ダイナミック ARP インスペクション<br>• IP SourceGuard<br>• MAC アドレスのセキュリティ |
| リリース 4.1.0 | ASR 9000 SIP-700 ラインカードでのこれらの VPLS 機能のサポートが追加されました。<br>• MAC 学習およびフォワーディング<br>• MAC アドレス エージング サポート<br>• MAC 制限<br>• スプリット ホライズン グループ<br>• MAC アドレス取り消し<br>• 未知のユニキャスト、ブロードキャスト、およびマルチキャスト パケットのフラッディング<br>• アクセス疑似回線<br>• H-VPLS PW アクセス<br>• PW の冗長性<br>G.8032 イーサネット リング保護機能のサポートが追加されました。 |
| リリース 4.2.1 | Flow Aware Transport（FAT）疑似回線機能のサポートが追加されました。 |

| リリース | 変更内容 |
|---|---|
| リリース 4.3.0 | 次の機能のサポートが追加されました。<br>• 疑似回線ヘッドエンド（PWHE）<br>• ASR 9000 拡張イーサネットラインカードのスケール拡張機能：<br>• VPWS および VPLS 内の 128000 疑似回線のサポート<br>• VPLS と VPWS インスタンスでの 128000 疑似回線のサポート<br>• ブリッジの 512 疑似回線までのサポート<br>• 128000 バンドル接続回線のサポート<br>• 128000 VLAN のサポート<br>• L2VPN over GRE |
| リリース 4.3.1 | 次のサポートを追加しました。<br>• BGP 自動検出を使用した VPLS の VC タイプ 4<br>• PWHE の IPv6 サポート |
| リリース 5.1.0 | マルチポイントレイヤ 2 サービスのラベル スイッチド マルチキャスト機能のサポートが追加されました。 |
| リリース 5.1.1 | 次のサポートを追加しました。<br>• 疑似回線ヘッドエンド PW-Ether サブインターフェイス（VC タイプ 5）および疑似回線ヘッドエンド PW-IW インターワーキング インターフェイス（VC タイプ 11）<br>• 疑似回線ヘッドエンドを介した LFA |
| リリース 6.1.2 | 次のサポートを追加しました。<br>• L2VPN のサービスパス設定<br>• L2VPN ルートポリシー |


- マルチポイント レイヤ 2 サービス実装の前提条件 （224 ページ）
- マルチポイントレイヤ 2 サービス の実装に関する情報 （224 ページ）

- マルチポイント レイヤ 2 サービスの実装方法 （255 ページ）
- マルチポイント レイヤ 2 サービスの設定例 （356 ページ）

# マルチポイント レイヤ 2 サービス実装の前提条件

マルチポイント レイヤ2サービスを設定する前に、次の作業を確認し、条件が満たされていることを確認してください。

- 適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。

  ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

- プロバイダー エッジ（PE）ルータが IP によって相互に到達できるように、コアに IP ルーティングを設定します。
- レイヤ2 トラフィックを開始して終了するようにループバック インターフェイスを設定します。PE ルータが他のルータのループバック インターフェイスにアクセスできるようにします。

（注） ループバック インターフェイスは、すべてのケースで必要というわけではありません。たとえば、マルチポイント レイヤ2 サービスが TE トンネルに直接マッピングされている場合、トンネル選択ではループバック インターフェイスは必要ありません。

- ラベル スイッチド パス（LSP）が PE ルータ間に存在するよう、コアで MPLS とラベル配布プロトコル（LDP）を設定します。

# マルチポイントレイヤ 2 サービス の実装に関する情報

マルチポイントレイヤ 2 サービスを実装するには、次の概念を理解する必要があります。

## マルチポイント レイヤ 2 サービスの概要

マルチポイント レイヤ 2 サービスを使用すると、地理的に離れたローカル エリア ネットワーク（LAN）セグメントを MPLS ネットワーク経由で単一ブリッジ ドメインとして相互接続できます。MAC アドレス ラーニング、エージング、およびスイッチングなどの従来の LAN の機能はすべて、単一のブリッジ ドメインに属する、リモート接続されたすべての LAN セグメント全体でエミュレートされます。

以降の各項では、マルチポイント レイヤ 2 サービス ネットワークのいくつかのコンポーネントについて説明します。

（注） マルチポイント レイヤ 2 サービスは、仮想プライベート LAN サービスとも呼ばれます。

# ブリッジ ドメイン

ネイティブ ブリッジ ドメインは、一連の物理ポートまたは仮想ポート（VFI を含む）から構成されるレイヤ 2 のブロードキャスト ドメインです。データ フレームは、宛先 MAC アドレスに基づいてブリッジ ドメイン内でスイッチングされます。マルチキャスト、ブロードキャスト、不明な宛先ユニキャスト フレームは、ブリッジ ドメイン内でフラッディングされます。また、送信元 MAC アドレス ラーニングは、ブリッジ ドメインのすべての着信フレームで行われます。学習されたアドレスは期限切れになります。着信フレームは、入力ポート、または入力ポートと MAC ヘッダー フィールドの両方の組み合わせのいずれかに基づいてブリッジ ドメインにマッピングされます。

デフォルトでは、スプリット ホライズンは同じ VFI 下の疑似回線でイネーブルです。ただし、デフォルト設定では、スプリット ホライズンは接続回線（インターフェイスまたは疑似回線）でイネーブルではありません。

### フラッディング最適化

Cisco ASR 9000 シリーズ ルータは、ブリッジ ドメインでトラフィックをブリッジしながら、不必要にフラッディングするトラフィック量を最小限に抑えます。フラッディング最適化機能によって、この機能を実現します。ただし、特定の障害回復シナリオでは、実際には、トラフィックの損失を防止するには追加のフラッディングが推奨されます。トラフィック損失は、ブリッジ ポート リンクの 1 つが非アクティブになり、スタンバイ リンクによって置き換えられる一時的な間隔中に発生します。

一部の設定では、トラフィック フラッディングを最小化する最適化は、ブリッジのリンクの 1 つで障害が発生し、スタンバイ リンクによって置き換えられる短期間にトラフィック損失という犠牲を払って行われます。そのため、設定に適した特定のフラッディング動作を指定するには、さまざまなモードでフラッディング最適化で設定できます。

次のフラッディング最適化モードを設定できます。

### 帯域幅最適化モード

フラッディング トラフィックは、ブリッジ ドメインに接続されたブリッジ ポートまたは疑似回線のラインカードだけに送信されます。これは、デフォルトのモードです。

### コンバージェンス モード

フラッディング トラフィックはシステムのすべてのラインカードに送信されます。トラフィックは、ブリッジドメインに接続されたブリッジポートまたは疑似回線の有無に関係なくフラッディングされます。そのブリッジ ドメインに接続された等コスト MPLS パス（ECMP）が複数ある場合は、トラフィックはすべての ECMP にフラッディングされます。

コンバージェンス モードの目的は、障害によりブリッジ リンクが変更される短いインターバル中に失われる絶対トラフィック量を最小限にすることです。

### TE FRR 最適化モード

トラフィック エンジニアリング高速再ルーティング（TE FRR）の最適化モードは、ブリッジ ドメインに接続された TE FRR 疑似回線に関するフラッディング動作を除き、帯域幅最適化モードに似ています。TE FRR 最適化モードでは、トラフィックは、プライマリおよびバックアップ FRR インターフェイスの両方にフラッディングされます。ブリッジ トラフィックが FRR の回復時間の制約に準拠するように、このモードは、FRR フェールオーバー中のトラフィック損失を最小限にするために使用されます。

### ダイナミック ARP インスペクション

ダイナミック ARP インスペクション（DAI）は、アドレス解決プロトコル（ARP）スプーフィング攻撃から保護する方法です。不正な IP/MAC アドレス バインディングを持つ ARP パケットを代行受信し、ログに記録して、廃棄します。この機能により、ネットワークをある種の中間者攻撃から保護することができます。DAI 機能は、デフォルトではディセーブルです。

ARP では、IP アドレスを MAC アドレスにマッピングすることで、レイヤ 2 ブロードキャスト ドメイン内の IP 通信を可能にします。スプーフィング攻撃は、ARP 要求が実際に受信されなかった場合でも、ホストからの ARP 応答を許可するために発生します。次に攻撃が発生した後、攻撃下にあるデバイスからのすべてのトラフィックは、最初に攻撃者のシステムを通過し、次にルータ、スイッチ、またはホストを通過します。ARP スプーフィング攻撃は、サブネットに接続されているデバイスの ARP キャッシュに偽りの情報を送信することにより、レイヤ 2 ネットワークに接続されているデバイスに影響を及ぼす可能性があります。ARP キャッシュに偽りの情報を送信することを ARP キャッシュ ポイズニングといいます。

ダイナミック ARP インスペクション機能を使用することで、有効な ARP 要求と応答だけが中継されることを保証できます。ARP インスペクションは 2 種類あります。

- 必要なインスペクション：送信側の MAC アドレス、IPv4 アドレス、受信側のブリッジ ポート XID およびブリッジがチェックされます。
- オプション インスペクション：次の項目が検証されます。
  - 送信元 MAC：送信者および送信元の MAC がチェックされます。チェックは、すべての ARP または RARP パケットで行われます。
  - 宛先 MAC：ターゲットおよび宛先の MAC がチェックされます。チェックは、すべての応答または応答反転パケットで実行されます。
  - IPv4 アドレス：ARP 要求では、送信者の IPv4 アドレスが 0.0.0.0、マルチキャスト アドレス、またはブロードキャスト アドレスかどうかを調べるためにチェックが実行されます。ARP 応答および ARP 応答反転では、ターゲットの IPv4 アドレスが 0.0.0.0、マルチキャスト アドレス、またはブロードキャスト アドレスかどうかを調べるためにチェックが実行されます。このチェックは、要求、応答、および応答反転パケットに応じて実行されます。

（注） DAI機能は、接続回線およびEFPでサポートされます。現在、DAI機能は疑似回線ではサポートされません。

### IP ソース ガード

IP ソース ガード（IPSG）は、非ルーテッド レイヤ 2 インターフェイスで IP トラフィックを制限するために、DHCP スヌーピング バインディング データベースおよび手動で設定された IP ソース バインディングに基づいてトラフィックをフィルタリングするセキュリティ機能です。

IPSG 機能は、悪意のあるホストが正当のホストの IP アドレスを推測することによって正当のホストを操作しないように、レイヤ 2 ポートで送信元 IP アドレスをフィルタリングします。この機能は、動的な DHCP スヌーピングおよび静的な IP ソース バインディングを使用して、IP アドレスをホストと照合します。

まず、DHCP パケットを除き、IPSG 用に設定された EFP のすべての IP トラフィックがブロックされます。クライアントが DHCP サーバから IP アドレスを受信したあと、またスタティック IP ソース バインディングが管理者によって設定されたあと、その IP 送信元アドレスのある全トラフィックがそのクライアントから許可されます。他のホストからのトラフィックは拒否されます。このフィルタリングは、隣接ホストの IP アドレスを要求することによって、ホストのネットワーク攻撃を制限します。

（注） IPSG 機能は、接続回線および EFP でサポートされます。現在、IPSG 機能は疑似回線ではサポートされません。

## 疑似回線

疑似回線は、PE ルータのペア間のポイントツーポイント接続です。その主な機能は、共通 MPLS 形式にカプセル化することによって、基礎となるコア MPLS ネットワーク経由でイーサネットなどのサービスをエミュレートすることです。共通 MPLS 形式へのサービスのカプセル化によって、疑似回線では、通信事業者は MPLS ネットワークにサービスを統合できます。

次のスケール拡張機能は、ASR 9000 拡張イーサネットラインカードに適用できます。

- VPWS および VPLS 内の 128000 疑似回線のサポート
- VPLS と VPWS インスタンスでの 128000 疑似回線のサポート
- ブリッジの 512 疑似回線までのサポート

（注） このスケール拡張機能は、RSP3 および ASR 9000 拡張イーサネットラインカードが使用されるハードウェア設定内でサポートされます。ただし、これらの拡張機能は、RSP2、ASR 9000 拡張イーサネットラインカードおよび Cisco ASR 9000 シリーズ SPA インターフェイス プロセッサ 700 ラインカードには適用されません。

### 疑似回線を介した DHCP スヌーピング

Cisco ASR 9000 シリーズ ルータでは、DHCP サーバが疑似回線に到達可能な DHCP スヌーピングを実行できます。疑似回線は信頼できるインターフェイスと見なされます。

dhcp ipv4 snoop profile {*dhcp-snooping-profile1*} コマンドは、ブリッジ上で DHCP スヌーピングを有効にし、ブリッジに DHCP スヌーピングプロファイルを対応付けるために、ブリッジドメインで提供されます。

## 仮想転送インスタンス

VPLS は、仮想転送インスタンス（VFI）の特性に基づいています。VFI は、宛先 MAC アドレス、送信元 MAC アドレス ラーニングとエージングなどに基づいて、転送などのネイティブ ブリッジング機能を実行できる仮想ブリッジ ポートです。

VFI は、VPLS インスタンスごとに PE ルータ上に作成されます。PE ルータでは、特定の VPLS インスタンスの VFI を検索して、パケットの転送先が決定されます。VFI は、特定の VPLS インスタンスの仮想ブリッジのように動作します。VFI には、特定の VPLS に属する複数の接続回線を接続できます。PE ルータは、その VPLS インスタンス内にあるすべての他の PE ルータに対するエミュレート VC を構築し、これらのエミュレート VC を VFI に接続します。パケット転送決定は、VFI で保持されるデータ構造に基づきます。

# MPLS ベースのプロバイダー コアの VPLS

VPLS はマルチポイント レイヤ 2 VPN テクノロジーであり、ブリッジング技法によって複数のカスタマーデバイスを接続します。マルチポイントブリッジングのビルディングブロックのブリッジドメインは、各 PE ルータに存在します。PE ルータのブリッジ ドメインへのアクセス接続は、接続回線と呼ばれます。接続回線は、一連の物理ポート、仮想ポート、またはネットワーク内の各 PE デバイスのブリッジに接続されている両方ポートです。

接続回線をプロビジョニングした後、この特定のインスタンスの MPLS ネットワークを介したネイバー関係が、エンド PE を識別する一連の手動コマンドによって確立されます。ネイバー アソシエーションが完了すると、MPLS コアとカスタマー ドメイン間のゲートウェイである疑似回線のフルメッシュがネットワーク側プロバイダーエッジデバイス間で確立されています。

MPLS/IP プロバイダー コアは、1 つのブロードキャスト ドメインを構成するために、各 PE デバイス上の複数の接続回線を接続する仮想ブリッジをシミュレートします。また、これらの間でエミュレート仮想回線（VC）を構成するために、VPLS インスタンスに参加しているすべての PE ルータも必要です。

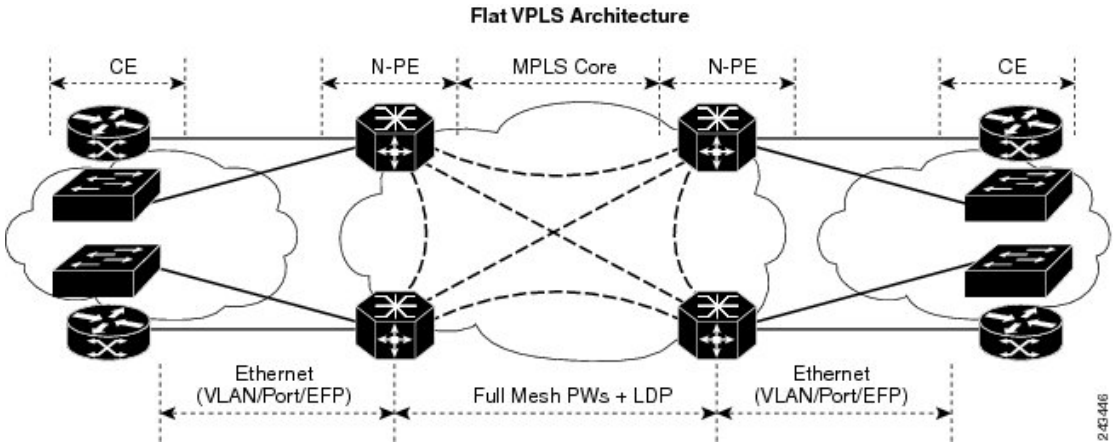
次に、サービス プロバイダー ネットワークは、宛先 MAC アドレスを調べてカスタマーに固有のブリッジ ドメイン内でパケットの交換を開始します。不明、ブロードキャスト、マルチキャストの宛先 MAC アドレスを持つすべてのトラフィックは、サービス プロバイダー ネットワークに接続するすべての接続済み CE カスタマー エッジ デバイスにフラッディングされます。ネットワーク側プロバイダー エッジ デバイスは、パケットがフラッディングされると送信元 MAC アドレスを学習します。トラフィックは、学習されたすべての MAC アドレスのカスタマー エッジ デバイスにユニキャストされます。

MPLS プロバイダーコアでは、VPLS 疑似回線トラフィックは、LDP プロトコルをサポートする任意のインターフェイスを介して動的にルーティングできます。

## VPLS アーキテクチャ

基本的な VPLS アーキテクチャまたはフラット VPLS アーキテクチャでは、プロバイダー エッジ（PE）ルータ間のエンドツーエンド接続がマルチポイント イーサネット サービスを提供できます。次の図は、IP/MPLS ネットワークでのネットワーク プロバイダー エッジ（N-PE）ノード間の相互接続を示すフラット VPLS アーキテクチャです。

**図 *18* : 基本的な *VPLS* アーキテクチャ**




VPLS ネットワークでは、各 PE ルータのブリッジ ドメイン（レイヤ 2 ブロードキャストドメイン）の作成が必要です。VPLS プロバイダーエッジデバイスは、MAC テーブルおよびブリッジ ドメイン情報を転送するすべての VPLS を保持します。さらに、すべてのフラッディング ブロードキャスト フレームおよびマルチキャスト複製を処理します。

VPLS アーキテクチャの PE は、疑似回線（PW）のフルメッシュに接続します。仮想転送インスタンス（VFI）は、疑似回線のメッシュの相互接続に使用されます。ブリッジ ドメインは、PW メッシュを介してイーサネット マルチポイント ブリッジングを提供する仮想スイッチング インスタンス（VSI）を作成するために VFI に接続されます。VPLS ネットワークは、エミュレートされたイーサネット スイッチを作成するために、MPLS 疑似回線を使用して VSI をリンクします。

VPLS では、1 つの VPLS インスタンスに関与するすべてのカスタマー装置（CE）デバイスが同じ LAN 上に表示されるため、CE デバイスでポイントツーポイント回線のフル メッシュを必要とせずに、マルチポイント トポロジで相互に直接通信できます。サービス プロバイダーは、カスタマーごとに別のブリッジ ドメインを定義することで、MPLS ネットワーク上で複数のカスタマーに VPLS サービスを提供できます。あるブリッジ ドメインからのパケットが別のブリッジ ドメインには伝送または配信されることはないため、LAN サービスのプライバシーが確保されます。

VPLS は、同じレイヤ 2 ブロードキャスト ドメインに属する複数サイト間で、イーサネット IEEE 802.3、VLAN IEEE 802.1q、および VLAN-in-VLAN（Q-in-Q）トラフィックを転送します。VPLS は、フラッディング ブロードキャスト、マルチキャスト、およびブリッジで受信し

た不明なユニキャスト フレームを含む単純な VLAN サービスを提供します。VPLS ソリューションでは、PE ルータ間で確立された疑似回線のフル メッシュが必要です。VPLS 実装は、ラベル配布プロトコル（LDP）ベースの疑似回線シグナリングに基づきます。

# レイヤ 2 スイッチングの VPLS

VPLS テクノロジーには、レイヤ 2 ブリッジングを実行するように Cisco ASR 9000 シリーズ ルータを設定する機能が含まれます。このモードでは、Cisco ASR 9000 シリーズ ルータを他のシスコスイッチと同様に動作するように設定できます。

次の機能がサポートされています。

- ブリッジング IOS XR トランク インターフェイス
- EFP でのブリッジング

これらのブリッジング機能の例については、「マルチポイント レイヤ 2 サービスの設定例」セクションを参照してください。

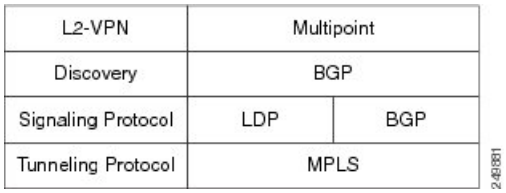
# VPLS ディスカバリおよびシグナリング

VPLS はレイヤ 2 マルチポイント サービスであり、WAN サービスで LAN サービスをエミュレートします。VPLS は、サービスプロバイダーが、パケットスイッチドネットワークで複数の LAN セグメントを相互接続し、単一の LAN として動作できるようにします。サービス プロバイダーは、VPLS を使用するお客様へのネイティブイーサネットアクセス接続を提供できます。

VPLS のコントロール プレーンは、2 つの重要なコンポーネントであるオートディスカバリおよびシグナリングで構成されます。

- VPLS オートディスカバリは、手動で VPLS ネイバーをプロビジョニングする必要性をなくします。VPLS オートディスカバリは、各 VPLS PE ルータが同じ VPLS ドメインに属する他のプロバイダー エッジ（PE）ルータを検出できるようにします。
- PE が検出されると、VPLS ドメインの PE ルータで PW のフル メッシュを形成している PE ルータの各ペア間で、疑似回線（PW）がシグナリングおよび確立されます

**図 *19 : VPLS* オートディスカバリとシグナリング**




## BGP ベースの VPLS オートディスカバリ

VPLS を含め VPN テクノロジーの重要な点は、ネットワーク デバイスが特定の VPN とのアソシエーションについて他のデバイスに自動的に信号を送信する機能です。オートディスカバリ

では、この情報を VPN のすべてのメンバーに配布する必要があります。VPLS は、BGP が最適であるマルチポイント メカニズムです。

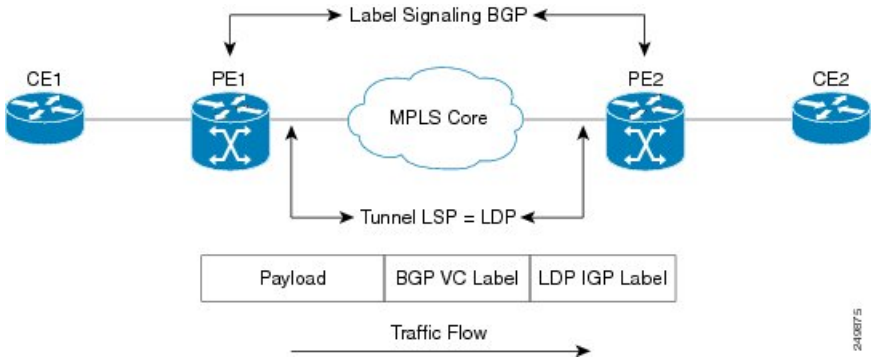
BGP ベースの VPLS オートディスカバリにより、VPLS ネイバーを手動でプロビジョニングする必要がなくなります。VPLS オートディスカバリは、各 VPLS PE ルータが同じ VPLS ドメインに属する他のプロバイダー エッジ（PE）ルータを検出できるようにします。VPLS オートディスカバリは、いつ PE ルータが追加されたか VPLS ドメインから削除されたかもトラックします。ディスカバリ プロセスが完了すると、各 PE ルータは、VPLS 疑似回線（PW）の設定に必要な情報を取得します。

BGP 自動検出が有効になっている場合でも、自動検出プロセスに参加していない VPLS PE ルータに対して疑似回線を手動で設定できます。

## BGP シグナリングによる BGP オートディスカバリ

ネットワークでの VPLS の実装では、プロバイダー エッジ（PE）ルータ間で PW のフル メッシュを確立する必要があります。PW には、BGP シグナリングを使用して信号を送信できます。

**図 *20 :* ディスカバリおよびシグナリングの属性**




BGP のシグナリングおよびオートディスカバリ方式には、次のコンポーネントがあります。

- PE が、特定の VPLS のメンバーであるリモート PE を学習するための方法。このプロセスをオートディスカバリといいます。
- PE が、特定の VPLS の特定のリモート PE で予期される疑似回線ラベルを学習する方法。このプロセスをシグナリングといいます。

BGP ネットワーク層到達可能性情報（NLRI）は、上記の 2 つのコンポーネントを同時に処理します。特定の PE によって生成される NLRI には、他の PE で必要な情報が含まれています。これらのコンポーネントは、各 PE の疑似回線を手動で設定することなく、各 VPLS の疑似回線のフル メッシュを自動的に設定できるようにします。

### BGP AD とシグナリングによる VPLS の NLRI フォーマット

次の図に、BGP AD とシグナリングによる VPLS の NLRI フォーマットを示します

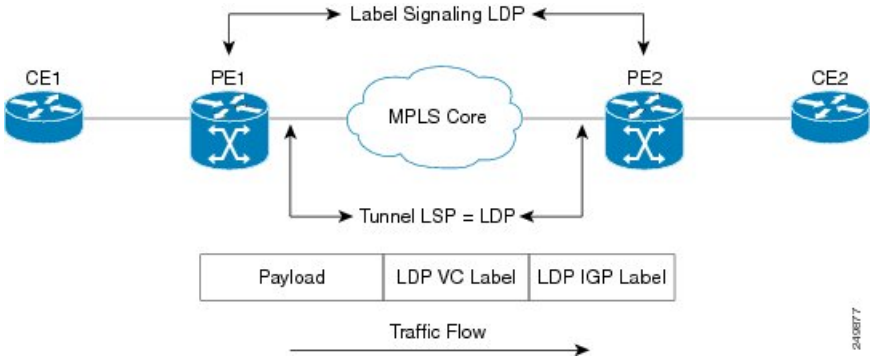
図 *21 : NLRI* フォーマット

| |
|---|
| Length (2 octets) |
| Route Distinguisher (8 octets) |
| VE ID (2 octets) |
| VE Block Offset (2 octets) |
| VE Block Size (2 octets) |
| Label Base (3 octets) |

## LDP シグナリングによる BGP オートディスカバリ

疑似回線のシグナリングでは、2 つのエンドポイント間で情報を交換する必要があります。ラベル配布プロトコル（LDP）は、ポイントツーポイントシグナリングに適しています。プロバイダー エッジ デバイス間の疑似回線のシグナリングは、ターゲット LDP セッションを使用して、ラベルの値と属性を交換し、疑似回線を設定します。

図 *22 :* ディスカバリおよびシグナリングの属性




PE ルータは、各 VPLS の BGP で ID をアドバタイズします。この ID は、VPLS インスタンス内で一意であり、VPLS ID と同様に機能します。ID は、BGP アドバタイズメントを受信している PE ルータが、アドバタイズメントに関連付けられた VPLS を識別し、正しい VPLS インスタンスにインポートできるようにします。このようにして、VPLS ごとに、PE ルータは、VPLS のメンバーである他の PE ルータを学習します。

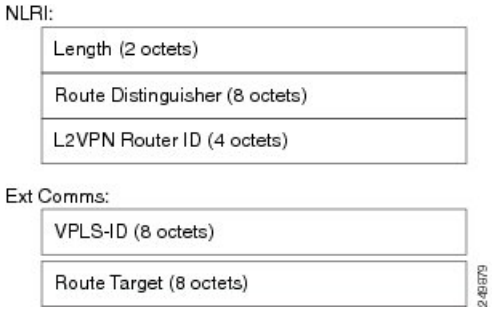
LDP プロトコルは、他のすべての PE ルータに疑似回線を設定するために使用されます。FEC 129 はシグナリングに使用されます。FEC 129 で伝送される情報には、VPLS ID、Target Attachment Individual Identifier（TAII）、および Source Attachment Individual Identifier（SAII）が含まれます。

LDP アドバタイズメントには、疑似回線上の着信トラフィックの予想される内部ラベルまたは VPLS ラベルも含まれます。これは、LDP ピアが、疑似回線を関連付ける VPLS インスタンスおよびその疑似回線でのトラフィックの送信時に使用することが予想されるラベル値を特定できるようにします。

### NLRI と拡張コミュニティ

次の図では、ネットワーク層到達可能性情報（NLRI）および拡張コミュニティ（Ext Comm）について説明します。

**図 *23 : NLRI* と拡張コミュニティ**




# L2VPN のサービスパス設定

サービスパス設定機能（SPP）は、トラフィック エンジニアリング（TE）トンネルでの L2VPN サービスのトランスポートパスを制御するのに便利です。SPP機能を使用すると、マルチプロトコル ラベル スイッチング（MPLS）ネットワークでの転送中に、サービスがパス選択を制御できます。SPP は、コントロール プレーン ポリシーを転送クラスに関連付けることによって実現されます。SPP は、BGP AD を使用した VPLS と PBB-EVPN および EVPN の EVI においてサービス単位のパス選択ができます。

L2VPN の SPP は、次の 2 つの手順で実装されます。

- パス選択：着信トラフィックをサービス単位で分類します。
- パス設定：転送クラスサービスを使用してパスを設定します。

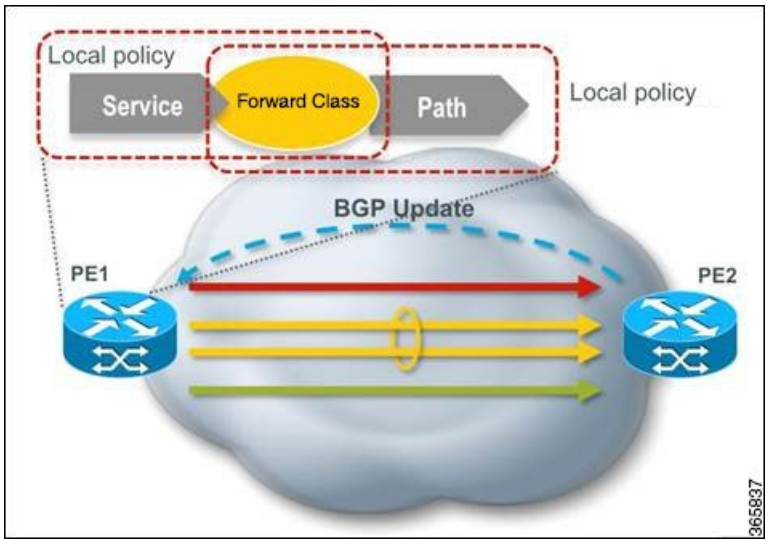
SPP の詳細については、「MPLS VPN セッションのサービスパス設定」モジュールを参照してください。

## サービスパス設定の機能概要

SPPを使用すると、コントロールプレーンで設定されたポリシーに基づいてサービスがパスを選択できます。

セットアップに 2 つのプロバイダーエッジ（PE）ルータがあるシナリオを考えてみます。PE1 は入力ノードとして機能し、PE2 は出力ノードとして機能します。

図 *24 :* サービスパス設定シナリオの例




入力 PE（PE1）は、顧客からのルートを受信します。ローカルポリシーで顧客に割り当てる属性を決定します。PE1 は、VFI または EVI サービスに基づいて作成されたローカルポリシーに基づいて、転送クラスをプレフィックスに関連付けます。一致する転送クラスで事前設定されたトンネルが、トラフィックを転送するために選択されます。

# L2VPN ルートポリシー

L2VPN ルートポリシー機能により、BGP 自動検出を使用した L2VPN VPWS および L2VPN VPLS の export コミュニティ設定が有効になります。BGP がルートポリシーを実行します。この機能は、L3VPN サービスの BGP サブモードでのルートポリシーのサポートに似ています。

次の項目は、L2VPN ルートポリシー機能について説明しています。

1. RPL は、L2VPN エクスポート VPLS（VFI）と L2VPN エクスポート VPWS（MP2MP）の 2 つの新しい接続点のポリシーアクションとして標準コミュニティを設定するために使用されます。
2. L2VPN 設定では、BGP 自動検出を使用して設定された特定のブリッジドメインに使用するルートポリシーを指定します。
3. L2VPN は、BGP に L2VPN コンテキストとともにルートポリシー名を送信します。
4. BGP プロセスは、標準コミュニティを L2 NLRI に挿入します。

# VPLS LDP シグナリングにおける Cisco IOS XR と Cisco IOS 間の相互運用性

Cisco IOS ソフトウェアは、BGP アップデート メッセージ内で、最初のバイト内の NLRI の長さをビット型式でエンコードします。ただし、Cisco IOS XR ソフトウェアは、NLRI の長さを

2 バイトで解釈します。したがって、VPLS-VPWS アドレス ファミリを使用する BGP ネイバーが IOS と IOS XR 間に設定されている場合、NLRI の不一致が発生し、ネイバー間のフラッピングの原因になります。この競合を避けるために、IOS は **prefix-length-size 2** コマンドをサポートしています。IOS が IOS XR とともに動作するようにするには、このコマンドをイネーブルにする必要があります。IOS で **prefix-length-size 2** コマンドが設定されている場合、NLRI の長さはバイト単位でエンコードされます。この設定は、IOS を IOS XR とともに動作させるために必要です。

次に、**prefix-length-size 2** コマンドを使用した IOS の設定の例を示します。

```
router bgp 1
 address-family l2vpn vpls
  neighbor 5.5.5.2 activate
  neighbor 5.5.5.2 prefix-length-size 2 --------> NLRI length = 2 bytes
 exit-address-family
```

# MAC アドレス関連パラメータ

MAC アドレス テーブルには、既知の MAC アドレスおよび転送情報のリストが含まれます。現在の VPLS の仕様では、MAC アドレス テーブルとその管理がルート プロセッサ（RP）カードで維持されます。

次のトピックでは、MAC アドレス関連パラメータについて説明します。

（注） ブリッジドメインレベルで MAC の制限またはアクションを修正した後で、アクションを有効にするために、ブリッジドメインを非アクティブにしてアクティブにしてください。（トラフィックが通過する）接続回線での MAC の制限またはアクションを変更した後で、アクションを有効にするために、接続回線を非アクティブにしてアクティブにする必要があります。

## MAC アドレス フラッディング

イーサネット サービスでは、ブロードキャスト アドレスおよび不明な宛先アドレスに送信されるフレームをすべてのポートにフラッディングする必要があります。VPLS ブロードキャスト モデル内のフラッディングを取得するために、すべての不明ユニキャスト、ブロードキャスト、およびマルチキャスト フレームが、対応する疑似回線およびすべての接続回線にフラッディングされます。したがって、PE は、接続回線および疑似回線の両方にパケットを複製する必要があります。

## MAC アドレスベース転送

フレームを転送するには、PE は、宛先 MAC アドレスを疑似回線または接続回線に関連付ける必要があります。このタイプのアソシエーションは、各 PE で静的設定によって行われるか、すべてのブリッジ ポートにフラッディングされるダイナミック学習によって行われます。

（注） たとえば接続回線または疑似回線で着信するフレームが同じ疑似回線で送信されるような場合、スプリット ホライズンの転送が適用されます。1 つの疑似回線で受信される疑似回線フレームは、同じ仮想転送インスタンス（VFI）の他の疑似回線には複製されません。

## MAC アドレスの送信元ベースの学習

フレームがブリッジ ポート（たとえば、疑似回線または接続回路）に到達し、受信側 PE ルータが送信元 MAC アドレスを認識していない場合、送信元 MAC アドレスは、疑似回線または接続回線に関連付けられます。MAC アドレスへの送信フレームは、適切な疑似回線または接続回線に転送されます。

MAC アドレスの送信元ベースの学習は、ハードウェア転送パスで学習される MAC アドレス情報を使用します。更新された MAC テーブルはルータのハードウェアに伝達され、それによってルータのハードウェアがプログラミングされます。

（注） スタティック MAC 移動は、1 つのポート、インターフェイス、または AC から別のポート、インターフェイス、または AC に対してはサポートされていません。たとえば、スタティック MAC が AC1（ポート 1）で設定されていて、AC2（ポート 2）の送信元 MAC と同じ MAC を持つパケットを送信しようとした場合、その MAC をダイナミック MAC として AC2 に接続することはできません。したがって、MAC を持つパケットは、設定したどのスタティック MAC アドレスとしても送信しないでください。

学習される MAC アドレスの数は、設定可能なポート単位およびブリッジ ドメイン単位の MAC アドレス制限によって制限されます。

## MAC アドレス エージング

MAC テーブルの MAC アドレスは、MAC アドレス エージング タイムの間だけ有効と見なされます。期限切れになると、関連する MAC エントリが再度読み込まれます。MAC エージング タイムをブリッジ ドメインだけで設定すると、ブリッジ ドメインのすべての疑似回線と接続回線において、設定したその MAC エージング タイムが使用されます。

ブリッジは、ブリッジテーブルに基づいてパケットの転送、フラッディング、ドロップを行います。ブリッジ テーブルは、スタティック エントリとダイナミック エントリの両方を保持します。スタティック エントリは、ネットワーク マネージャまたはブリッジ自体によって入力されます。ダイナミック エントリはブリッジ学習プロセスによって入力されます。ダイナミック エントリは、エントリが作成された時点か最後に更新された時点から、「エージング タイム」と呼ばれる指定された期間が経過すると、自動的に削除されます。

ブリッジ型ネットワークのホストが移動する可能性が高い場合、ブリッジが変更に迅速に適応できるようにエージング タイムを小さくします。ホストが連続して送信しない場合は、より長い時間ダイナミック エントリを記録するようにエージング タイムを長くして、ホストが再度送信する場合よりフラッディングの可能性を低減できます。

## MAC アドレス制限

MACアドレス制限は、学習されるMACアドレスの数を制限するために使用されます。ブリッジ ドメイン レベルの制限は常に設定され、ディセーブルにできません。ブリッジ ドメイン レベルの制限のデフォルト値は 4000 で、1 ～ 512000 の範囲で変更できます。

制限を超えると、これらの通知を行うようシステムが設定されています。

- syslog（デフォルト）
- 簡易ネットワーク管理プロトコル（SNMP）トラップ
- syslog および SNMP トラップ
- なし（通知なし）

## MAC アドレス取り消し

高速な VPLS コンバージェンスでは、ダイナミックに学習された MAC アドレスを削除または学習解除できます。ラベル配布プロトコル（LDP）アドレス取り消しメッセージが MAC アドレスのリストと一緒に送信されます。これらのアドレスは、対応する VPLS サービスに参加する他のすべての PE で取り消す必要があります。

Cisco IOS XR VPLS の実装では、ダイナミックに学習された MAC アドレスの部分は、デフォルトで MAC アドレス エージング メカニズムを使用してクリアされます。MAC アドレス取り消し機能は、LDP アドレス取り消しメッセージによって追加されます。MAC アドレス取り消し機能をイネーブルにするには、l2vpn ブリッジ グループ ブリッジ ドメイン MAC コンフィギュレーション モードで **withdrawal** コマンドを使用します。MACアドレス取り消しがイネーブルであることを確認するには、**detail** キーワードとともに **show l2vpn bridge-domain** コマンドを使用します。

（注） デフォルトでは、Cisco IOS XR で LDP MAC 取り消し機能がイネーブルになっています。

LDP MAC 取り消し機能は、次のイベントが原因で生成されます。

- 接続回線がダウンした。CLI から接続回線を削除または追加できます。
- MAC 取り消しメッセージを VFI 擬似回線経由で受信した。RFC 4762 では、ワイルドカード（空のタイプ、長さ、および値（TLV）による方法）と、特定の MAC アドレス取り消しの両方が規定されています。Cisco IOS XR ソフトウェアは、ワイルドカードによる MAC アドレス取り消しだけをサポートしています。

## MAC アドレスのセキュリティ

インターフェイス レベルとブリッジアクセス ポート（サブインターフェイス）レベルでMACアドレス セキュリティを設定できます。ただし、インターフェイスで設定されたMACセキュリティは、ブリッジ ドメイン レベルで設定された MAC セキュリティよりも優先されます。

MAC セキュリティで設定された EFP で MAC アドレスを最初に学習して、次に同じ MAC アドレスを別の EFP で学習すると、次のイベントが発生します。

- パケットはドロップされます。
- 2 番めの EFP はシャットダウンされます。
- パケットが学習され、元の EFP からの MAC はフラッシュされます。

## MAC アドレス移動およびユニキャストトラフィックのカウンタ

MAC アドレス移動とユニキャストトラフィックのカウンタが、ASR9K プラットフォームの VPLSブリッジポートに導入されています。これらのカウンタは本質的には、L2VPNブリッジポートの統計カウンタです。MAC 移動とユニキャストトラフィックのカウンタは、トラブルシューティングのために導入されています。Cisco ASR 9000 高密度 100GE イーサネットラインカードとCisco ASR 9000 拡張イーサネットラインカードが、これらのカウンタをサポートしています。

MAC 移動とユニキャストトラフィックのカウンタの詳細については、AC ブリッジ、PW ブリッジ、PBB エッジ、および VXLAN ブリッジポートで、**detail** キーワードを指定して **show l2vpn bridge-domain** コマンドを使用してください。

（注） すべてまたは一部のブリッジポートトラフィックが ASR 9000 イーサネットラインカードに転送される場合、MAC アドレス移動とユニキャストトラフィックのカウンタは正確ではない可能性があります。

# LSP Ping over VPWS および VPLS

Cisco IOS XR ソフトウェアでは、（LDP FEC128 を使用してシグナリングされる）ポイントツーポイント疑似回線のラベルスイッチドパス（LSP）ping とトレースルート検証メカニズムの既存のサポートが、VFI（VPLS）に関連付けられている疑似回線をカバーするために拡張されています。現在、LDP シグナリング FEC128 疑似回線の LSP ping とトレースルートのサポートは、手動で設定された VPLS 疑似回線に制限されています。また、Cisco IOS XR ソフトウェアは、VPWS に適用可能な LDP FEC129 AII タイプ 2 を使用してシグナリングされる、または VPLS に適用可能な LDP FEC129 AII タイプ 1 を使用してシグナリングされる、ポイントツーポイント単一セグメント疑似回線の LSP ping をサポートしています。仮想回線接続検証（VCCV）のサポートと **ping mpls pseudowire** コマンドの詳細については、『*MPLS Command Reference for Cisco ASR 9000 Series Routers*』を参照してください。

# スプリット ホライズン グループ

IOS XR ブリッジ ドメインは、スプリット ホライズン グループと呼ばれる 3 つのグループの 1 つに接続回線（AC）と疑似回線（PW）を集約します。ブリッジ ドメインに適用した場合、スプリット ホライズンは、スプリット ホライズン グループのメンバー間のフラッディングと転

送動作を示します。次の表では、スプリット ホライズン グループの 1 つのメンバーで受信したフレームがどのように処理されるかを示し、トラフィックが同じスプリット ホライズン グループの他のメンバーに転送される場合について説明します。

ブリッジ ドメイン トラフィックは、ユニキャストまたはマルチキャストのいずれかです。

フラッディングトラフィックは、不明なユニキャスト宛先 MAC アドレスフレームで構成されます。これは、イーサネット マルチキャスト アドレス（スパニングツリー BPDU など）に送信されるフレームです。イーサネット ブロードキャスト フレーム （MAC アドレス FF-FF-FF-FF-FF-FF）。

既知のユニキャスト トラフィックは、MAC 学習を使用するポートから学習されたブリッジ ポートに送信されるフレームで構成されます。

トラフィック フラッディングは、ブロードキャスト、マルチキャスト、不明なユニキャスト宛先アドレスに対して実行されます。

**表 *2 : Cisco IOS XR* でサポートされているスプリット ホライズン グループ**

| スプリット ホライズン グループ | このグループに属しているメンバー | グループ内のマルチキャスト | グループ内のユニキャスト |
|---|---|---|---|
| 0 | デフォルト：グループ 1 または 2 でカバーされないメンバー。 | 対応 | 対応 |
| 1 | VFI で設定されるすべての PW。 | なし | なし |
| 2 | split-horizon キーワードで設定された AC または PW。 | なし | なし |

スプリット ホライズン グループに関する重要事項：

- ブリッジ ドメインのメンバーであるすべてのブリッジ ポートまたは PW が、3 つのグループのうちの 1 つに属している必要があります。
- デフォルトでは、すべてのブリッジ ポートまたは PW がグループ 0 のメンバーです。
- ブリッジ ドメイン設定の VFI コンフィギュレーション サブモードは、このドメインのメンバーがグループ 1 に含まれていることを示しています。
- グループ 0 で設定された PW はアクセス疑似回線と呼ばれます。
- **split-horizon group** コマンドは、グループ 2 のメンバーとしてブリッジポートまたは PW を指定するために使用されます。
- ASR9000 は 1 個の VFI グループだけをサポートします。

# レイヤ 2 セキュリティ

次のトピックでは、レイヤ 2 セキュリティをサポートするレイヤ 2 VPN の拡張について説明します。

## ポート セキュリティ

ポートへのトラフィック送信を許可する MAC アドレスを制限することによって、ダイナミックに学習される MAC アドレス、およびスタティック MAC アドレスを使用したポートセキュリティを使用して、ポートの入力トラフィックを制限します。セキュアポートにセキュア MAC アドレスを割り当てると、ポートは、定義されたアドレスのグループ外に送信元アドレスがある入力トラフィックを転送しません。セキュア MAC アドレスの数を 1 つに制限し、単一のセキュア MAC アドレスを割り当てると、そのポートに接続されているデバイスはそのポートの全帯域を使用できます。

次のポート セキュリティ機能がサポートされます。

- ブリッジまたはポートの MAC テーブルのサイズを制限します。
- MAC アドレスの処理と通知を容易にします。
- ブリッジまたはポートの MAC エージング タイムとモードをイネーブルにします。
- ブリッジまたはポートのスタティック MAC アドレスをフィルタリングします。
- セキュアまたは非セキュアとしてポートをマークします。
- ブリッジまたはポートでフラッディングをイネーブルまたはディセーブルにします。

ポートにセキュア MAC アドレスの最大数を設定した後で、次のいずれかの方法でアドレステーブルにセキュア アドレスを組み込むようポート セキュリティを設定できます。

- **static-address** コマンドを使用して、すべてのセキュア MAC アドレスをスタティックに設定します。
- 接続されているデバイスの MAC アドレスで、ポートがセキュア MAC アドレスをダイナミックに設定できるようにします。
- アドレス数をいくつかスタティックに設定し、残りのアドレスがダイナミックに設定されるようします。

## Dynamic Host Configuration Protocol スヌーピング

Dynamic Host Configuration Protocol（DHCP）スヌーピングは、信頼できないホストと信頼済み DHCP サーバとの間のファイアウォールのように機能するセキュリティ機能です。DHCP スヌーピング機能は次のアクティビティを実行します。

- 信頼できないソースからの DHCP メッセージを検証し、無効なメッセージをフィルタ処理して除外する。

- 信頼できるソースおよび信頼できないソースからのDHCPトラフィックのレートを制限する。
- DHCPスヌーピングのバインディングデータベースを構築し、管理する。このデータベースには、リースIPアドレスがある信頼できないホストに関する情報が保存されています。
- 信頼できないホストからの以降の要求を検証するためにDHCPスヌーピングのバインディング データベースを使用する。

DHCP に関する追加情報については、『Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Configuration Guide』を参照してください。

# G.8032 イーサネット リング保護

ITU-T G.8032 で定義されているイーサネット リング保護（ERP）プロトコルは、リング トポロジでイーサネット トラフィックを保護し、イーサネット レイヤのリング内でループが発生しないようにします。ループは、事前設定されたリンクまたは障害リンクのいずれかをブロックすることで防止されます。

## 概要

各イーサネット リング ノードは、2 個の独立したリンクを使用してイーサネット リングに参加する隣接イーサネット リング ノードに接続されます。リング リンクは、ネットワークに影響を及ぼすループの編成を許可しません。イーサネット リングは、イーサネット リングを保護するために特定のリンクを使用します。この特定のリンクは、リング予備リンク（RPL）と呼ばれます。リング リンクは、リング リンク（別名リング ポート）の 2 個の隣接するイーサネット リング ノードとポートで区切られます。

（注） イーサネット リングでのイーサネット リング ノードの最小数は 2 です。

リング保護スイッチングの基礎は次のとおりです。

- ループ回避の原則
- 学習、転送、およびフィルタリング データベース（FDB）メカニズムの使用

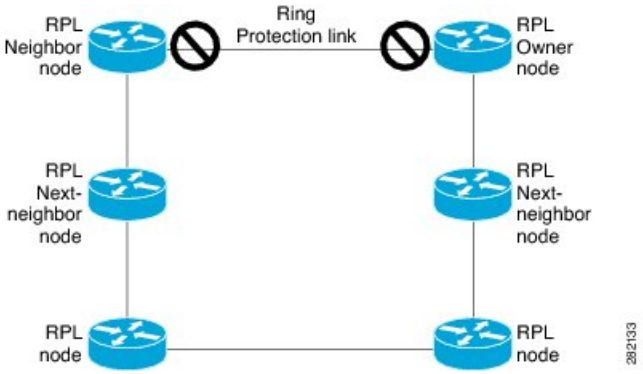
イーサネット リングでのループ回避は、RPL である 1 つのリング リンクを除くすべてで常にトラフィック フローを確保することで行います。複数のノードが、リングの形成に使用されます。

- RPL オーナー：ループがイーサネット トラフィックで形成されないように、RPL を介してトラフィックをブロックします。リングには RPL オーナーは 1 つだけ存在します。
- RPL ネイバー ノード：RPL ネイバー ノードは、RPL に隣接するイーサネット リング ノードです。通常の状態で RPL の終了をブロックします。このノード タイプはオプションであり、保護されている場合 RPL の使用を防止します。

- RPL の次のネイバー ノード：RPL の次のネイバー ノードは、RPL オーナー ノードまたは RPL ネイバー ノードに隣接するイーサネット リング ノードです。これは、主にリングでの FDB フラッシュ最適化に使用されます。このノードはオプションです。

次の図は G.8032 イーサネット リングの例です。

**図 *25 : G.8032* イーサネット リング**




リングのノードは、RAPS と呼ばれる制御メッセージを使用して、RPL リンクのオンとオフを切り替えるアクティビティを調整します。リンクの障害によって、障害が発生したリンクに面するポートをノードがブロックした後で、障害が発生したリンクに隣接するノードから両方の方向に RAPS 信号障害（RAPS SF）メッセージがトリガーされます。このメッセージの取得時に、RPL オーナーは、RPL ポートのブロックを解除します。

（注） リングの単一のリンク障害によって、ループフリー トポロジが確保されます。

リング リンクおよびノードの障害を検出するために、回線ステータスおよび接続障害管理プロトコルが使用されます。回復フェーズ中に、障害が発生したリンクが復元されると、復元されたリンクに隣接するノードは、RAPS no request（RAPS NR）メッセージを送信します。このメッセージの取得時に、RPL オーナーは RPL ポートをブロックし、RAPS no request, root blocked（RAPS NR, RB）メッセージを送信します。これにより、リング内の RPL オーナー以外のその他すべてのノードが、すべてのブロックされたポートのブロックを解除します。ERP プロトコルは、リング トポロジの単方向障害と複数のリンク障害シナリオの両方で機能するために十分に強力です。

G.8032 リングは、次の基本的なオペレータ管理コマンドをサポートします。

- Force switch（FS）：オペレータは、特定のリング ポートを強制的にブロックできます。
  - 既存の SF 状態がある場合でも有効です。
  - サポートされるリング用の複数の FS コマンド。
  - 即時のメンテナンス操作を可能にするために使用できます。
- Manual switch（MS）：オペレータは、特定のリング ポートを手動でブロックできます。

- 既存の FS または SF 状態では無効です。
- 新しい FS または SF 状態によって上書きされます。
- 複数の MS コマンドは、すべての MS コマンドを取り消します。

- Clear：リング ポートで既存の FS または MS コマンドを取り消します。
  - 非リバーティブ モードをクリアするために（RPL オーナーで）使用されます。

G.8032 リングは、複数のインスタンスをサポートできます。インスタンスは、物理的なリングに実行される論理リングです。そのようなインスタンスは、リング上のロード バランシング VLAN などのさまざまな理由で使用されます。たとえば、奇数の VLAN はリングの 1 方向に送信され、偶数の VLAN は他の方向に送信されることがあります。特定の VLAN は 1 つのインスタンスだけで設定できます。これらは複数のインスタンスと重複できません。重複すると、データ トラフィックまたはRAPSパケットは論理リングを通過する可能性があるため、望ましくありません。

G.8032 ERP は、リンク障害の検出に回線ステータスと接続障害管理（CFM）に依存する新しいテクノロジーを提供します。100ms の間隔で CFM Continuity Check Message（CCM）メッセージを実行することにより、SONET のようなスイッチング時間パフォーマンスとループ フリー トラフィックを実現できます。

イーサネット接続障害管理（CFM）とイーサネット障害検出（EFD）の設定の詳細については、『Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Configuration Guide』の「Configuring Ethernet OAM on the Cisco ASR 9000 Series Router」モジュールを参照してください。

### タイマー

G.8032 は、競合状態および不要なスイッチング操作を回避するために異なる ERP タイマーを使用することを指定します。

- 遅延タイマー：RPL をブロックする前にネットワークが安定していることを確認するために RPL オーナーによって使用されます。
  - SF 状態の後で、SF が断続的に中断していないことを確認するために、Wait-to-Restore（WTR）タイマーが使用されます。WTR タイマーはオペレータが設定できます。デフォルトの時間間隔は 5 分です。時間間隔の範囲は 1 ～ 12 分です。
  - FS/MS コマンドの後で、バックグラウンド状態でないことを確認するために、Wait-to-Block タイマーが使用されます。

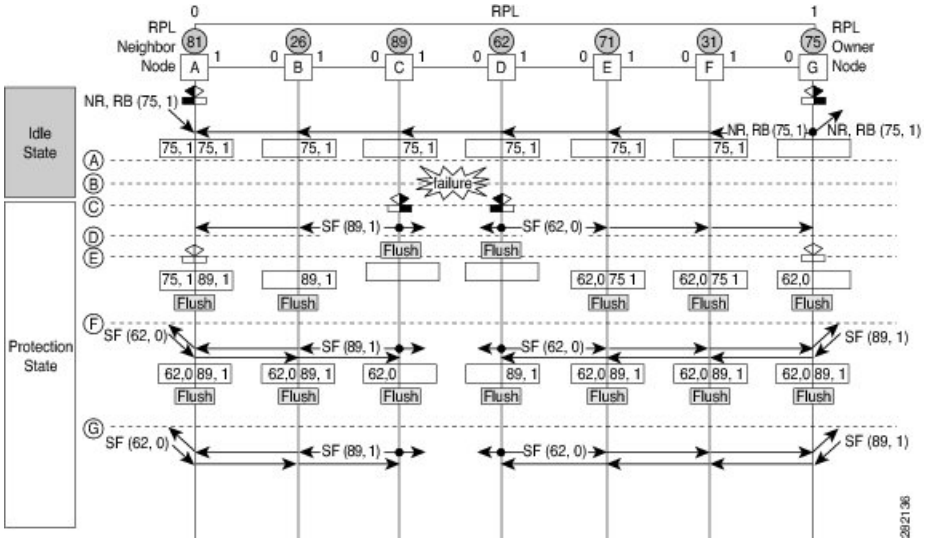
（注） Wait-to-Block タイマーは、Wait-to-Restore タイマーよりも短くなることがあります。

- ガードタイマー：状態の変更時にすべてのノードで使用されます。これは、潜在的な古いメッセージが不要な状態変更を引き起こさないようにします。ガードタイマーは設定可能であり、デフォルトの時間間隔は 500 ミリ秒です。時間間隔の範囲は 10 ～ 2000 ミリ秒です。
- hold-off タイマー：断続的なリンク障害をフィルタリングするために、基盤となるイーサネット レイヤによって使用されます。hold-off タイマーは設定可能であり、デフォルトの時間間隔は 0 秒です。時間間隔の範囲は 0 ～ 10 秒です。
  - 障害は、このタイマーの有効期限が切れた場合だけリング保護メカニズムに報告されます。

### 単一のリンク障害

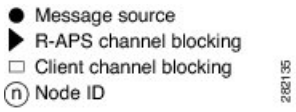
次の図は、単一のリンク障害が発生した場合の保護スイッチングを表しています。

**図 *26 : G.8032* の単一のリンク障害**




次の図は、7つのイーサネットリングノードで構成されたイーサネットリングを表しています。RPL は、イーサネット リング ノード A と G の間のリング リンクです。このようなシナリオでは、RPL の両端がブロックされます。イーサネット リング ノード G は RPL オーナー ノードで、イーサネット リング ノード A は RPL ネイバー ノードです。

次の記号が使用されます。



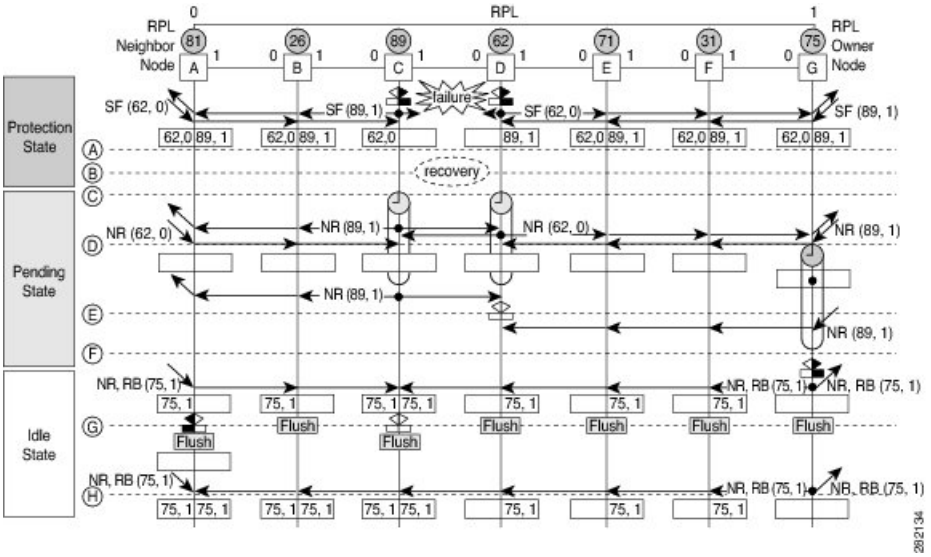

次の手順では、図 8 で表されている単一のリンク障害でのステップを説明します。

1. リンクは正常な状態で動作しています。
2. 障害が発生します。

3. イーサネット リング ノード C と D は、ローカルの信号障害を検出し、ホールドオフ時間間隔後に障害が発生したリング ポートをブロックし、FDB フラッシュを実行します。
4. イーサネット リング ノード C と D は、SF 状態が続いている間、両方のリング ポートの（ノード ID、BPR）ペアとともに RAPS（SF）メッセージの定期的な送信を開始します。
5. RAPS（SF）メッセージを受信するすべてのイーサネット リング ノードが FDB フラッシュを実行します。RPL オーナー ノード G と RPL ネイバー ノード A が RAPS（SF）メッセージを受信すると、イーサネット リング ノードは自身の RPL の終端をブロック解除し、FDB フラッシュを実行します。
6. 2 番目の RAPS（SF）メッセージを受信するすべてのイーサネット リング ノードは、FDB フラッシュを再度実行します。これは、ノード ID と BPR ベース メカニズムが原因です。
7. 安定した SF 状態：イーサネット リングの RAPS メッセージ（SF）。これ以上の RAPS（SF）メッセージは、さらなるアクションをトリガーしません。

次の図は、単一のリンク障害が発生した場合の復帰を表しています。

**図 *27*：単一のリンク障害回復（リバーティブ操作）**




次の手順では、図 9 で表されている単一のリンク障害回復でのステップを説明します。

1. リンクが安定した SF 状態で動作しています。
2. リンク障害回復が行われます。
3. イーサネットリングノード C と D は、信号障害（SF）状態のクリアを検出し、ガードタイマーを開始し、両方のリングポートの RAPS（NR）メッセージの定期的な送信を開始します（ガードタイマーは、RAPS メッセージの受信を防止します）。
4. イーサネットリングノードが RAPS（NR）メッセージを受信すると、受信側リングポートのノード ID および BPR のペアが削除され、RPL オーナーノードは WTR タイマーを開始します。

5. イーサネットリングノードCとDでガードタイマーの有効期限が切れると、受信する新しいRAPSメッセージを受け入れることがあります。イーサネットリングノードDは、イーサネットリングノードCから上位のノードIDを持つRAPS（NR）メッセージを受信し、障害が発生していないリングポートのブロックを解除します。
6. WTRタイマーの有効期限が切れると、RPLオーナーノードは、RPLの終端をブロックし、（ノードID、BPR）ペアを持つRAPS（NR、RB）メッセージを送信し、FDBフラッシュを実行します。
7. イーサネットリングノードCがRAPS（NR、RB）メッセージを受信すると、ブロックされたリングポートのブロックを解除し、RAPS（NR）メッセージの送信を停止します。一方、RPLネイバーノードAがRAPS（NR、RB）メッセージを受信すると、RPLの終了をブロックします。さらに、イーサネットリングノードA～Fは、ノードIDとBPRベースメカニズムが存在することが原因で、RAPS（NR、RB）メッセージを受信するとFDBフラッシュを実行します。

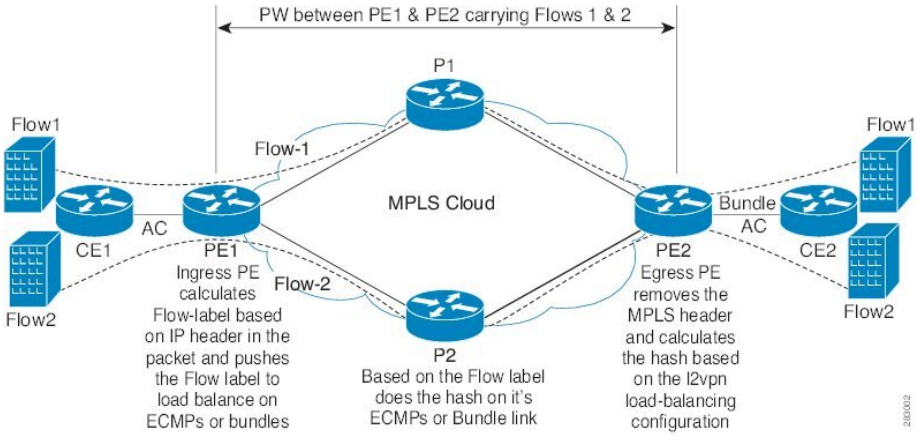
# Flow Aware Transport 疑似回線（FAT PW）

ルータは通常、ラベルスタックの最低ラベル（特定の疑似回線のすべてのフローに対して同じラベル）に基づいてトラフィックをロードバランスします。このとき、非対称ロードバランシングが発生することがあります。このコンテキストでは、フローは同じ送信元/宛先ペアを持つパケットのシーケンスを示します。パケットは、送信元プロバイダー エッジ（PE）から宛先 PE に転送されます。

Flow-Aware Transport 疑似回線（FAT PW）は、疑似回線内の個々のフローを識別する機能を提供します。また、ルータに対してこれらのフローを使用してトラフィックをロードバランスする機能を提供します。等価コスト マルチパス（ECMP）が使用されている場合は、FAT PW はコア内のトラフィックのロードバランスに使用されます。疑似回線に伝送される個々のパケット フローに基づいてフロー ラベルが作成され、最低ラベルとしてパケットに挿入されます。ルータは、フローラベルをロードバランシングに使用できます。これにより、コア内のECMPパスまたはリンクがバンドルされたパスでより適切なトラフィックの分配が実現します。

次の図に、FAT PW と、ECMP およびバンドルされたリンクに分配される 2 つのフローの例を示します。

図 *28 : FAT PW* と *ECMP* およびバンドルされたリンクへ分配される *2* つのフロー




追加ラベルは、仮想回線（VC）のフロー情報を含むスタック（フロー ラベルと呼ばれる）に追加されます。フロー ラベルは、PW 内のフローを区別する一意の ID で、送信元/宛先 MAC アドレスと送信元/宛先 IP アドレスから取得されます。フロー ラベルにはラベル スタック（EOS）ビット セットの末尾が含まれ、VC ラベルの後ろや、コントロール ワード（存在する場合）の前に挿入されます。入力 PE は、フロー ラベルを計算し、転送します。FAT PW コンフィギュレーションは、フロー ラベルをイネーブルにします。出力 PE は、決定が行われないように、フロー ラベルを廃棄します。

すべてのコア ルータが、FAT PW でフローラベルに基づいてロード バランシングを実行します。これにより、ECMP とリンク バンドルへのフローの分配が可能になります。

# 疑似回線ヘッドエンド

疑似回線（PW）は、IP/MPLS パケットスイッチド ネットワーク（PSN）でのペイロードの透過的な伝送を可能にします。PWは、コア ネットワークに戻るカスタマー トラフィックのための、簡単で管理可能な軽いトンネルと見なされます。サービス プロバイダーは、PW 接続をネットワークのアクセスおよび集約の領域に拡張しています。

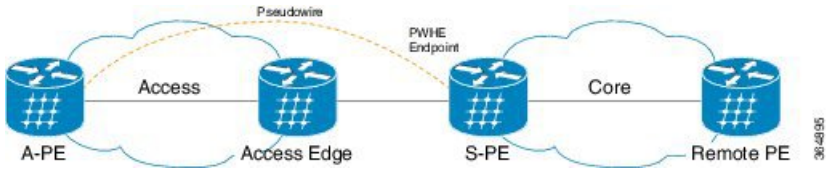
疑似回線ヘッドエンド（PWHE）は、レイヤ 3（VRF またはグローバル）ドメインまたはレイヤ2 ドメインへのアクセス疑似回線（PW）の終端を可能にするテクノロジーです。PWは、共通の IP/MPLS ネットワーク インフラストラクチャへのカスタマー トラフィックのトンネリングのために、簡単でスケーラブルなメカニズムを提供します。PWHEにより、カスタマーは、サービス プロバイダー エッジ（PE）ルータ上で、QoS アクセス リスト（ACL）、L3VPN などの機能を PWHE インターフェイス単位でプロビジョニングできます。

（注） カプセル化のデフォルトは PWHE ではサポートされていません。

PWHEは、再帰的プレフィックスと非再帰的プレフィックスを介して到達可能な疑似回線ネイバーにクロスコネクトします。再帰的プレフィックスによる到達可能性は、Cisco ASR 9000 シ

リーズ ルータでの BGP RFC3107 サポートの導入によって実現されています。シナリオ例として、次のネットワークトポロジについて考えてみましょう。

**図 *29*：疑似回線のネットワーク**




PWHE クロスコネクトの設定では、A-PE（アクセスプロバイダーエッジ）と S-PE 間の相互接続は、MPLS ラベルを IP プレフィックスとともに配布する BGP RFC3107 を介して行われます。顧客のネットワークでは、IGP を使用することで、顧客の自律システムの外部にある S-PE デバイスへの接続の提供を回避できます。

すべての実際の用途で、PWHE インターフェイスは他の既存の L3 インターフェイスと同様に扱われます。PW は次のいずれかのモードで動作します。

- ブリッジ型インターワーキング（VC タイプ 5 または VC タイプ 4）
- IP インターワーキングモード（VC タイプ 11）

VC タイプ 4 と VC タイプ 5 では、PW は IP ペイロードを使用して顧客のイーサネットフレーム（タグ付きまたはタグなし）を伝送します。そのため、S-PE デバイスは、PWHE を介して学習された顧客の IP アドレスに対して ARP 解決を実行する必要があります。VC タイプ 4（VLAN タグ付き）および VC タイプ 5（イーサネットポート/raw）では、PWHE はブロードキャストインターフェイスとして機能します。一方、VC タイプ 11（IP インターワーキング）では、PWHE はポイントツーポイントインターフェイスとして機能します。そのため、PWHE インターフェイスには、PW-Ether（VC タイプ 4 および 5 用）と PW-IW（VC タイプ 11 用）の 2 つのタイプがあります。これらの PW は S-PE で VRF または IP グローバルテーブルに終端できます。

## PWHE の利点

PWHE の実装には次のような利点があります。

- アクセスまたは集約ネットワークの基礎となる物理転送メディアからの、サービス PE のカスタマー側インターフェイス（CFI）の分離
- アクセスまたは集約ネットワークおよびサービス PE の CapEx の削減
- カスタマー側レイヤ 2 UNI インターフェイス セットの分配および拡大
- OAM 機能の統一方法の実装
- プロバイダーによるレイヤ 3 サービスのフットプリントの延長または拡張が可能
- 次世代ネットワーク（NGN）にカスタマー トラフィックの終端方法を提供

## 機能制限

- PWHE では、FAT（フロー認識型トランスポート）ラベルまたはエントロピーラベルはサポートされていません。
- システムは、汎用インターフェイスリスト内のインターフェイス経由の発信側 PE ルータの PWID に基づいて、および後続の P ルータの PWID に基づいて PWHE をロードバランシングします。
- PWHE は、2 番目のレベルの BGP ラベル付きユニキャスト（LU）再帰ではサポートされていません。

## 汎用インターフェイスリスト

汎用インターフェイスリストには、PW-HE 接続で使用される物理インターフェイスまたはバンドルインターフェイスのリストが含まれています。

汎用インターフェイスリストでは、メインインターフェイスのみがサポートされ、サブインターフェイスはサポートされません。汎用インターフェイスリストは双方向であり、アクセス側ラインカードの受信と送信インターフェイスの両方を制限します。汎用インターフェイスリストは、コア方向には影響を与えません。

汎用インターフェイスリストは、PWHEインターフェイスに割り当てられるリソースをこのリストで指定した一連のインターフェイスに制限します。

S-PE のみで汎用インターフェイスリストが認識されます。正常な場合 PWHE パケットは汎用インターフェイスリストのメンバーを設定したラインカードのみに着信すると見なされます。汎用インターフェイスリストのメンバーを設定していないラインカードにパケットが着信した場合は、ドロップされます。

## 疑似回線ヘッドエンドを介した LFA

リリース 5.1.1 からは、ループフリー代替（LFA）ルートで PW-HE がサポートされています。

PW-HE インターフェイスで LFA を有効にするには、すべてのルーティングパス（保護およびバックアップ）をその PW-HE インターフェイスの汎用インターフェイスリストに含める必要があります。すべてのルーティングパスが汎用インターフェイスリストに含まれていない場合は、LFA が有効になっている場合でもトラフィックが失われる可能性があります。これは、LFA ルートが汎用インターフェイスリストに含まれていないものである可能性があるためです。

PW-HE インターフェイスで LFA を設定するには、次の手順を実行します。

1. 汎用インターフェイスリストの設定
2. IP/LDP 高速再ルーティングの設定

IP 高速再ルーティングループフリー代替の設定の詳細については、『Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide』の「Implementing IS-IS on Cisco IOS XR Software」モジュールを参照してください。

## PW-HE マルチキャスト

疑似回線ヘッドエンド（PW-HE）インターフェイスのマルチキャストサポートは、拡張イーサネットカードでのみ使用できます。

PW-HE マルチキャスト機能の詳細については、『Cisco ASR 9000 Series Aggregation Services Router Multicast Configuration Guide, Release 5.1.x』の「Implementing Layer 3 Multicast Routing」の章を参照してください。

## PW-HE over MPLS-TE トンネル

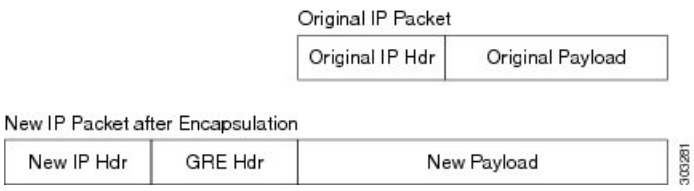
PW-HE over MPLS-TE トンネル機能は、TE トンネルを介した（疑似回線ヘッドエンドを使用した）疑似回線トラフィックの転送をサポートしています。

PW-HE over MPLS TE トンネルの詳細については、『Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide, Release 5.1.x』の「Implementing MPLS Traffic Engineering」の章を参照してください。

# L2VPN over GRE

システムは、総称ルーティング カプセル化（GRE）トンネル上で IP パケットを転送するために、最初に GRE ヘッダーで元の IP パケットをカプセル化します。カプセル化された GRE パケットは、パケットを宛先に転送するために使用する外部の転送ヘッダーによって再びカプセル化されます。次の図に、IP 転送ネットワークでの GRE のカプセル化の例を示します。

**図 *30 : GRE* のカプセル化**




（注） 新しい IP パケットでは、新しいペイロードは元の IP パケットに似ています。また、新しい IP ヘッダー（新しい IP Hdr）は、トンネル IP ヘッダーに似ており、転送ヘッダーにも似ています。

GRE トンネルエンドポイントで GRE パケットのカプセル化が解除されると、ペイロードタイプに基づいてそのパケットが転送されます。たとえば、ペイロードがラベル付きパケットの場合は、仮想回線（VC）ラベルまたは VPN ラベルに基づいて、L2VPN および L3VPN にそれぞれ転送されます。

## L2VPN over GRE の制限事項

L2VPN over GRE を設定する際に考慮する必要があるいくつかの制限事項を次に示します。

- VPLS フローベースのロードバランシングシナリオの場合、GRE トンネルは、トンネルの送信元または宛先の Cyclic Redundancy Check（CRC; 巡回冗長検査）に基づいて発信パスに固定されます。ユニキャストトラフィックとフラッディングトラフィックは、特定の GRE トンネルに対して常に同じ物理パスを使用します。
- 入力接続回線は、L2VPN over GRE 用の ASR 9000 拡張イーサネットラインカードである必要があります。さらに、GRE トンネルの宛先は、ASR 9000 拡張イーサネットラインカードでのみ到達可能である必要があります。
- L2VPN over GRE 機能は、ASR 9000 イーサネットラインカードまたは Cisco ASR 9000 シリーズ SPA インターフェイスプロセッサ 700 ラインカードではサポートされていません。これは、入力接続回線および GRE 宛先が GRE を介して到達可能であるためです。
- TE over GRE を介した疑似回線のシナリオはサポートされていません。
- 優先パスの制限事項：
  - GRE を優先パスとして設定する場合、GRE トンネル（出力 ACL）では出力機能がサポートされません。
  - 優先パスでは、VCCV ping またはトレースルートはサポートされません。
  - 優先パスは、PE（プロバイダーエッジ）to PE トポロジで設定された疑似回線でのみサポートされます。

## GRE 配置シナリオ

L2VPN ネットワークでは、次のシナリオで GRE を配置できます。

- プロバイダー エッジ（PE）と PE ルータ間に GRE トンネルを設定
- P ルータと P ルータ間に GRE トンネルを設定
- P ルータと PE ルータ間に GRE トンネルを設定

次の図は、さまざまなシナリオを示します。

**図 *31 : PE* ルータと *PE* ルータ間に設定された *GRE* トンネル**

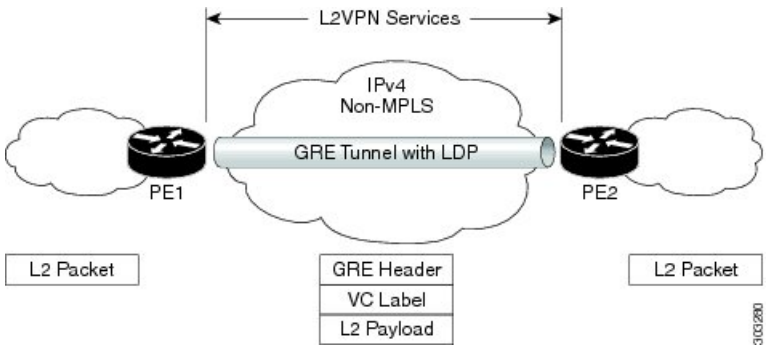


図 *32 : P* ルータと *P* ルータ間に設定された *GRE* トンネル

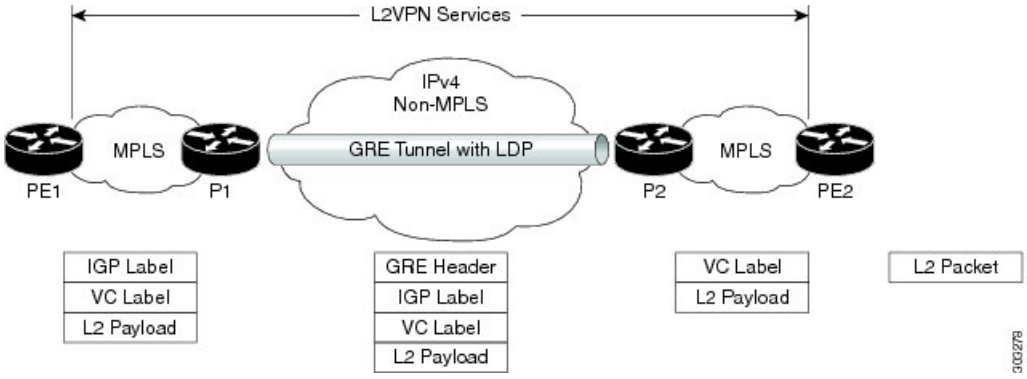
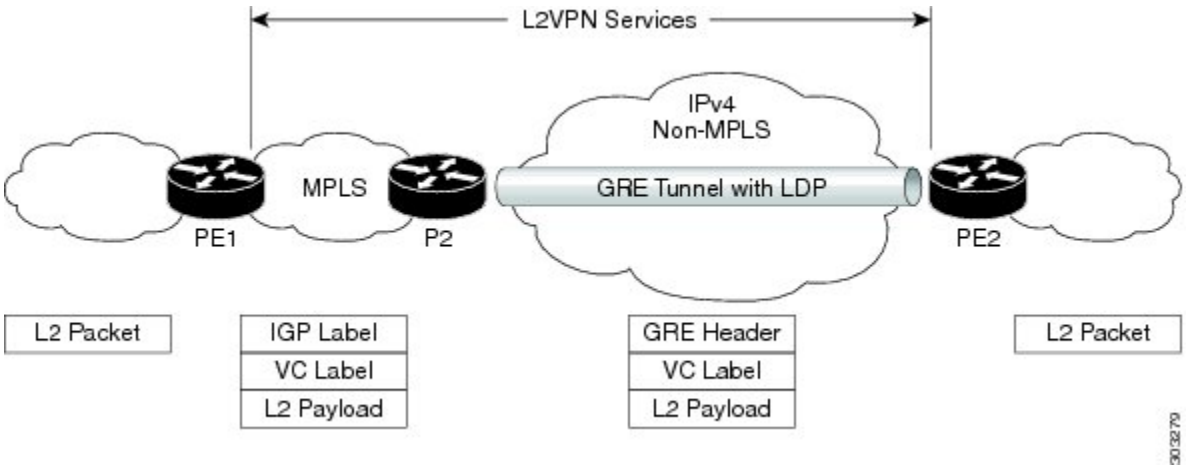



図 *33 : P* ルータと *PE* ルータ間に設定された *GRE* トンネル




（注） これらの配置シナリオは、VPWS および VPLS に適用されます。

## 優先パスとしての GRE トンネル

優先トンネル パス機能により、疑似回線を特定の GRE トンネルにマッピングできます。接続回線は、リモート PE ルータの IP アドレス（IGP または LDP を使用して到達可能）ではなく、GRE トンネル インターフェイスに相互接続されます。優先トンネル パスを使用する場合、L2 トラフィックを転送する GRE トンネルが 2 台の PE ルータ間で動作することが常に想定されます（つまり、始端はインポジション PE ルータで、終端はディスポジション PE ルータです）。

# マルチポイントレイヤ2サービスのラベルスイッチドマルチキャスト

マルチポイントレイヤ 2 サービスのラベル スイッチド マルチキャスト（LSM）は、マルチプロトコル ラベル スイッチング（MPLS）ネットワークを介してマルチキャストトラフィックを送信する、レイヤ 2 ベースのソリューションです。マルチポイントレイヤ 2 サービスでは、

イーサネット LAN エミュレーションを提供するために、マルチポイントレイヤ 2 サービスドメインに参加している PE ルータでポイントツーポイント（P2P）疑似回線（PW）が設定されます。ブロードキャスト、マルチキャスト、および不明なユニキャストのトラフィックを、マルチポイントレイヤ 2 サービスドメインの入力複製またはラベル スイッチド マルチキャストを介して送信できます。

## 入力複製とその制限事項

入力複製では、ブロードキャスト、マルチキャスト、および不明なユニキャストのトラフィックが、入力 PE ルータで複製されます。同じパケットの個々のコピーが、同じ VPLS ドメインに参加しているリモート PE ルータに送信されます。ただし、入力複製には次の制限事項があります。

- ブロードキャストおよびマルチキャストの VPLS トラフィックが多い場合に、リンク帯域幅が大量に浪費されます
- 入力 PE ルータが複製によって最も影響を受けるため、リソースが大量に消費されます

## ソリューションとしての VPLS LSM

VPLS ラベル スイッチドマルチキャスト（LSM）は、入力複製の制限に対応するための効果的なマルチキャスト ソリューションです。このソリューションは、MPLS ネットワーク内のポイントツーマルチポイント（P2MP）ラベルスイッチドパス（LSP）を使用し、VPLS ドメインを介してマルチキャストトラフィックを転送します。

（注） ブロードキャスト、マルチキャスト、または未知のユニキャストトラフィックのみが P2MP LSP を介して送信されます。

VPLS LSM ソリューションでは、BGP ベースの P2MP PW シグナリングがサポートされています。そのため、VPLS ドメインに参加しているリモート PE は、BGP 自動検出メカニズムを使用して自動的に検出されます。VPLS ドメインごとに P2MP PW を作成すると、VPLS ドメイン内の PW の VPLS P2MP サービスをエミュレートできます。

VPLS に使用されるマルチキャストツリーには、次の 2 つのタイプがあります。

- 包含ツリー
- 選択ツリー

包含ツリー：このオプションでは、SP ネットワーク内で単一マルチキャスト配信ツリーを使用して、特定の PE に接続されている指定された一連の VPLS サイトからのすべてのマルチキャストトラフィックを伝送できます。特定のマルチキャストツリーは、単一 VPLS インスタンスに属しているサイトまたは個別の VPLS インスタンスに属するサイトによって発生したトラフィックを伝送するように設定できます。同じツリー上で複数の VPLS インスタンスのトラフィックを伝送する機能は、集約包含と呼ばれます。ツリーには、ツリーを使用している VPLS インスタンスのいずれかのメンバーであるすべての PE が含まれている必要があります。PE は、そのストリームのトラフィックの受信に関心のある受信者がいなくても、マルチキャスト

ストリームのマルチキャストトラフィックを受信する場合があります。VPLS LSM の包含マルチキャストツリーは、P2MP ツリーです。

選択ツリー：PE は選択マルチキャストツリーを使用して、1 または複数の特定 IP マルチキャストストリーム（同じまたは異なる VPLS インスタンスに属する PE-CE インターフェイスを介して PE が受信）の IP マルチキャストトラフィックを、これらの VPLS インスタンスに属する PE のサブセットに送信します。これは、特定のマルチキャストストリーム（高帯域幅マルチキャストストリームなど）用に個別の SP マルチキャストツリーを作成する機能を PE に提供するためです。したがって、これらのマルチキャストストリームのトラフィックは、ストリームの受信者がいる PE ルータにのみ到達できます。これにより、VPLS インスタンス内の他の PE ルータのフラッディングが回避されます。

## VPLS LSM に関する制限事項

VPLS LSM には次の制限があります。

- RSVP-TE マルチキャストツリーを使用した BGP-AD シグナリングのみがサポートされます。
- 静的に設定された PW はサポートされません。
- 包含ツリーのみがサポートされます。
- 選択マルチキャストツリーはサポートされません。
- LDP シグナリング P2P PW はサポートされません。BGP シグナリング PW のみがサポートされます。
- RSVP-TE マルチキャストツリーのみがサポートされます。
- P2MP マルチキャストツリールートに参加しているブリッジドメインで IGMP スヌーピングが有効になっている場合、IGMP スヌーピングトラフィックは入力複製を使用して転送されます。P2MP マルチキャストツリーは使用されません。
- P2MP PW シグナリングは、VPLS ドメインで P2MP が有効になっている場合に開始されます。したがって、1 つまたは複数のリーフ PE がマルチキャストツリーに参加できない可能性があります。このシナリオでは、リーフ PE は P2MP ツリーを介して送信されたトラフィックを受信しません。
- リーフ PE がマルチキャストツリーに正常に参加するまで、トラフィックは Blackhole 状態になります。自動リカバリはサポートされません。
- MAC ラーニングは、不明なユニキャストトラフィックが P2MP マルチキャストツリーで送信された場合に発生します。これにより、トラフィックは P2P PW に切り替えられます。パケットの並べ替えは P2P として発生する可能性があり、P2MP PW はネットワークを介して異なるパスを選択する可能性があります。P2MP を介してすでに伝送されているトラフィックは、同じフローの P2P PW 上の新しいトラフィックよりも後に到着する可能性があります。
- VPLS LSM に関しては、A9K-SIP-700 ラインカードには次のような固有の制限があります。

- ISSU はサポートされていません。
- QoS は、このラインカードのアクセス PW ではサポートされません。
- MPLS-TE は、シリアルインターフェイスではサポートされません。
- TE か RSVP または両方が設定されている場合、RSVP はインターフェイスとネイバーを削除しません。
- FRR リンク保護のみがサポートされます。FRR ノード保護はサポートされません。
- サポートされている P2MP 対応 BD-VFI の最大数は 1000 です。
- Bud から Mid ノードに移行するには、 **no l2vpn** コマンドを使用して l2vpn 設定全体を削除します。**no multicast p2mp** コマンドを設定するだけでは十分ではありません。

# マルチポイント レイヤ 2 サービスの実装方法

ここでは、マルチポイント レイヤ 2 サービスの実装に必要なタスクについて説明します。

## ブリッジ ドメインの設定

次のトピックでは、ブリッジ ドメインの設定方法について説明します。

### ブリッジ ドメインの作成

ブリッジ ドメインを作成するには、次の作業を実行します。

#### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **commit** コマンドまたは **end** コマンドを使用します。

#### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを含めることができるブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain** *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## 疑似回線の設定

ブリッジ ドメインで疑似回線を設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge group name*
4. **bridge-domain** *bridge-domain name*
5. **vfi** { *vfi-name* }
6. **exit**
7. **neighbor** { *A.B.C.D* } { **pw-id** *value* }

8. **dhcp ipv4 snoop profile** *{ dhcp_snoop_profile_name }*
9. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

**例**：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

**例**：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge group name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain** *bridge-domain name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **vfi** { *vfi-name* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# vfi v1
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)#
```

仮想転送インターフェイス（VFI）パラメータを設定し、L2VPN ブリッジ グループ ブリッジ ドメイン VFI コンフィギュレーション モードを開始します。

- 指定した仮想転送インターフェイス名を設定するには、*vfi-name* 引数を使用します。

**ステップ 6** **exit**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)# exit
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

現在のコンフィギュレーション モードを終了します。

**ステップ 7** **neighbor** { *A.B.C.D* } { **pw-id** *value* }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# neighbor 10.1.1.2 pw-id 1000
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-pw)#
```

アクセス疑似回線ポートをブリッジ ドメインに追加するか、または疑似回線を仮想転送インターフェイス（VFI）に追加します。

- 相互接続ピアの IP アドレスを指定するには、*A.B.C.D* 引数を使用します。
- 疑似回線 ID および ID 値を設定するには、 **pw-id** キーワードを使用します。指定できる範囲は 1 ～ 4294967295 です。

**ステップ 8** **dhcp ipv4 snoop profile** *{ dhcp_snoop_profile_name }*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-pw)#dhcp ipv4 snoop profile profile1
```

ブリッジ上で DHCP スヌーピングをイネーブルにして、DHCP スヌーピング プロファイルを対応付けます。

**ステップ 9** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## メンバのブリッジ ドメインへの関連付け

ブリッジ ドメインの作成後、ブリッジ ドメインにインターフェイスを割り当てるには、この作業を実行します。次のタイプのブリッジ ポートは、ブリッジ ドメインに関連付けられています。

- イーサネットおよび VLAN
- VFI

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge group name*
4. **bridge-domain** *bridge-domain name*
5. **interface** *type interface-path-id*
6. （任意）**static-mac-address** { *MAC-address* }
7. **routed interface** *BVI-id*
8. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge group name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain** *bridge-domain name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **interface** *type interface-path-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# interface GigabitEthernet 0/4/0/0
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)#
```

インターフェイス コンフィギュレーション モードを開始し、同じブリッジ ドメインに属する他のインターフェイスからパケットを転送および受信できるブリッジ ドメインにインターフェイスを追加します。

**ステップ 6** （任意）**static-mac-address** { *MAC-address* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)# static-mac-address 1.1.1
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)# exit
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

スタティック MAC アドレスを設定してリモート MAC アドレスを疑似回線またはその他のブリッジ インターフェイスに関連付けます。

**ステップ 7** **routed interface** *BVI-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# routed interface BVI100
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)#
```

VPLS 疑似回線トラフィックを、統合ルーティングおよびブリッジング（IRB）を介してルーティングする必要がある場合は、この手順を実行します。このコマンドは、ブリッジグループ仮想インターフェイス コンフィギュレーション モードを開始し、同じブリッジ ドメインに属する他のインターフェイスからパケットを転送および受信できるブリッジ ドメインにブリッジグループ仮想インターフェイス（BVI）を追加します。BVI のステータスをアップにするには、この手順が不可欠です。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# ブリッジ ドメイン パラメータの設定

ブリッジ ドメイン パラメータを設定するには、ブリッジ ドメインに次のパラメータを関連付けます。

- 最大伝送ユニット（MTU）：ブリッジ ドメインのすべてのメンバーに同じMTUがあることを指定します。MTUサイズが異なるブリッジ ドメイン メンバーは、まだブリッジ ドメインに関連付けられている場合でもブリッジ ドメインによって使用されません。
- フラッディング：フラッディングは常に有効になります。
- ダイナミック ARP インスペクション（DAI）：有効な ARP 要求と応答だけが中継されるようにします。
- IP SourceGuard（IPSG）：レイヤ 2 ポートで送信元 IP アドレス フィルタリングをイネーブルにします。

（注） DAI および IPSG 機能が正常に動作していることを確認するには、DAI および IPSG 違反についてパケット ドロップ統計情報を調べます。パケット ドロップ統計情報は **show l2vpn bridge-domain** *bd-name* <> **detail** コマンドの出力で確認できます。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **flooding disable**
6. **mtu** *bytes*
7. **dynamic-arp-inspection** { **address-validation** | **disable** | **logging** }
8. **ip-source-guard logging**
9. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

### ステップ 1 configure

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

### ステップ 2 l2vpn

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

```
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

l2vpn コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain** *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、l2vpn ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **flooding disable**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# flooding disable
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

フラッディングを無効にします。

**ステップ 6** **mtu** *bytes*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# mtu 1000
```

ブリッジ ドメインの最大パケット サイズまたは最大伝送ユニット（MTU）サイズを調整します。

- バイト単位で MTU サイズを指定するには、*bytes* 引数を使用します。範囲は 64 ～ 65535 です。

**ステップ 7** **dynamic-arp-inspection** { **address-validation** | **disable** | **logging** }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# dynamic-arp-inspection
```

ダイナミック ARP インスペクション コンフィギュレーション サブモードを開始します。有効な ARP 要求および応答だけがリレーされるようになります。

（注） ブリッジ ドメインまたはブリッジ ポートのダイナミック ARP インスペクションを設定できます。

**ステップ 8** **ip-source-guard logging**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# ip-source-guard logging
```

IP ソース ガード コンフィギュレーション サブモードを開始し、レイヤ 2 ポート上で送信元 IP アドレス フィルタリングをイネーブルにします。

ブリッジ ドメインまたはブリッジ ポートで IP ソース ガードをイネーブルにできます。デフォルトでは、ブリッジの下のブリッジ ポートは親ブリッジから IP ソース ガード設定を継承します。

デフォルトでは、すべてのブリッジに対して IP ソース ガードがディセーブルです。

**ステップ 9** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## ブリッジ ドメインのディセーブル化

ブリッジ ドメインをディセーブルにするには、次の作業を実行します。ブリッジ ドメインをディセーブルにすると、ブリッジ ドメインに関連付けられているすべての VFI がディセーブルになります。引き続き、ブリッジ ドメインに関連付けられたブリッジ ドメインと VFI にメンバーを接続するか、または取り外すことができます。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge group name*
4. **bridge-domain** *bridge-domain name*
5. **shutdown**
6. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge group name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain** *bridge-domain name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、l2vpn ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **shutdown**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインをシャットダウンし、ブリッジと、ブリッジ下のすべての接続回線と疑似回線を管理ダウン状態に戻します。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。

• [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## 不明なユニキャスト フラッディングのブロック

ブリッジ ドメイン レベルで不明なユニキャスト トラフィックのフラッディングをディセーブルにするには、次の作業を実行します。

ブリッジ ドメイン、ブリッジ ポート、またはアクセス疑似回線レベルで不明なユニキャスト トラフィックのフラッディングをディセーブルにできます。デフォルトでは、不明なユニキャスト トラフィックは、ブリッジ ドメインのすべてのポートにフラッディングされます。

（注） ブリッジ ドメインで不明なユニキャスト トラフィックのフラッディングをディセーブルにすると、ブリッジ ドメイン内のすべてのポートがこの設定を継承します。ブリッジ ドメイン設定を上書きするように、ブリッジ ポートを設定できます。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **flooding unknown-unicast disable**
6. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain** *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、l2vpn ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **flooding unknown-unicast disable**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
flooding unknown-unicast disable
```

ブリッジ ドメイン レベルで不明なユニキャスト トラフィックのフラッディングをディセーブルにします。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## フラッディング最適化モードの変更

ブリッジ ドメインでフラッディング最適化モードを変更するには、次の作業を行います。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **flood mode convergence-optimized**
6. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain** *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、l2vpn ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **flood mode convergence-optimized**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# flood mode convergence-optimized
```

デフォルトのフラッディング最適化モードを帯域幅最適化モードからコンバージェンス モードに変更します。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# レイヤ 2 セキュリティの設定

次のトピックでは、レイヤ 2 セキュリティの設定方法について説明します。

## レイヤ 2 セキュリティのイネーブル化

ブリッジのレイヤ 2 ポート セキュリティをイネーブルにするには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge domain** *bridge-domain-name*
5. **security**
6. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

各ネットワーク インターフェイスをブリッジ グループに割り当てて、L2VPN ブリッジ グループ コンフィギュレーション モードを開始します。

**ステップ 4** **bridge domain** *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **security**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# security
```

ブリッジのレイヤ 2 ポート セキュリティをイネーブルにします。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## Dynamic Host Configuration Protocol（DHCP）プロファイルの対応付け

ブリッジ上でDHCP スヌーピングをイネーブルにし、ブリッジにDHCP スヌーピング プロファイルを対応付けるには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **dhcp ipv4 snoop** { **profile** *profile-name* }
6. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

各ネットワーク インターフェイスをブリッジ グループに割り当てて、L2VPN ブリッジ グループ コンフィギュレーション モードを開始します。

**ステップ 4** **bridge-domain** *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **dhcp ipv4 snoop** { **profile** *profile-name* }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# dhcp ipv4 snoop profile attach
```

ブリッジ上で DHCP スヌーピングをイネーブルにし、ブリッジに DHCP スヌーピング プロファイルを対応付けます。

- DHCP プロファイルを対応付けるには、profile キーワードを使用します。profile-name 引数は、DHCPv4 スヌーピングのプロファイル名です。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# レイヤ 2 仮想転送インスタンスの設定

次のトピックでは、レイヤ 2 仮想転送インスタンス（VFI）の設定方法について説明します。

## 仮想転送インスタンスの作成

ブリッジ ドメインのすべてのプロバイダー エッジ（PE）デバイスでレイヤ 2 仮想転送インスタンス（VFI）を作成するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge group name*
4. **bridge-domain** *bridge-domain name*
5. **vfi** {*vfi-name*}
6. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

**例**：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

**例**：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge group name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain** *bridge-domain name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **vfi** {*vfi-name*}

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# vfi v1
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)#
```

仮想転送インターフェイス（VFI）パラメータを設定し、L2VPN ブリッジ グループ ブリッジ ドメイン VFI コンフィギュレーション モードを開始します。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## 疑似回線の仮想転送インスタンスへの関連付け

VFI を作成した後、1 つ以上の疑似回線を VFI に関連付けるには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **vfi** { *vfi name* }
6. **neighbor** { *A.B.C.D* } { **pw-id** *value* }
7. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain** *bridge-domain-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **vfi** { *vfi name* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# vfi v1
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)#
```

仮想転送インターフェイス（VFI）パラメータを設定し、L2VPN ブリッジ グループ ブリッジ ドメイン VFI コンフィギュレーション モードを開始します。

**ステップ 6** **neighbor** { *A.B.C.D* } { **pw-id** *value* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)# neighbor 10.1.1.2 pw-id 1000
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-pw)#
```

疑似回線ポートをブリッジ ドメインに追加するか、または疑似回線を仮想転送インターフェイス（VFI）に追加します。

- 相互接続ピアの IP アドレスを指定するには、*A.B.C.D* 引数を使用します。
- 疑似回線 ID および ID 値を設定するには、**pw-id** キーワードを使用します。指定できる範囲は 1 ～ 4294967295 です。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## ブリッジ ドメインへの仮想転送インスタンスの関連付け

VFI をブリッジ ドメインのメンバーになるように関連付けるには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge group name*
4. **bridge-domain** *bridge-domain name*
5. **vfi** { *vfi name* }
6. **neighbor** { *A.B.C.D* } { **pw-id** *value* }
7. **static-mac-address** { *MAC-address* }
8. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge group name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain** *bridge-domain name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **vfi** { *vfi name* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# vfi v1
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)#
```

仮想転送インターフェイス（VFI）パラメータを設定し、L2VPN ブリッジ グループ ブリッジ ドメイン VFI コンフィギュレーション モードを開始します。

**ステップ 6** **neighbor** { *A.B.C.D* } { **pw-id** *value* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)# neighbor 10.1.1.2 pw-id 1000
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-pw)#
```

疑似回線ポートをブリッジ ドメインに追加するか、または疑似回線を仮想転送インターフェイス（VFI）に追加します。

- 相互接続ピアの IP アドレスを指定するには、*A.B.C.D* 引数を使用します。
- 疑似回線 ID および ID 値を設定するには、**pw-id** キーワードを使用します。指定できる範囲は 1 ～ 4294967295 です。

**ステップ 7** **static-mac-address** { *MAC-address* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-pw)# static-mac-address 1.1.1
```

スタティック MAC アドレスを設定してリモート MAC アドレスを疑似回線またはその他のブリッジ インターフェイスに関連付けます。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。

- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## 疑似回線への疑似回線クラスの接続

疑似回線に疑似回線クラスを接続するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge group name*
4. **bridge-domain** *bridge-domain name*
5. **vfi** { *vfi-name* }
6. **neighbor** { *A.B.C.D* } { **pw-id** *value* }
7. **pw-class** { *class-name* }
8. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge group name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain** *bridge-domain name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **vfi** { *vfi-name* }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# vfi v1
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)#
```

仮想転送インターフェイス（VFI）パラメータを設定し、L2VPN ブリッジ グループ ブリッジ ドメイン VFI コンフィギュレーション モードを開始します。

**ステップ 6** **neighbor** { *A.B.C.D* } { **pw-id** *value* }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)# neighbor 10.1.1.2 pw-id 1000
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-pw)#
```

疑似回線ポートをブリッジ ドメインに追加するか、または疑似回線を仮想転送インターフェイス（VFI）に追加します。

- 相互接続ピアの IP アドレスを指定するには、*A.B.C.D* 引数を使用します。
- 疑似回線 ID および ID 値を設定するには、 **pw-id** キーワードを使用します。指定できる範囲は 1 ～ 4294967295 です。

**ステップ 7** **pw-class** { *class-name* }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-pw)# pw-class canada
```

疑似回線に使用する疑似回線クラス テンプレート名を設定します。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# スタティックラベルを使用した疑似回線の設定

スタティックラベルを使用して Any Transport over Multiprotocol（AToM）疑似回線を設定するには、次の作業を実行します。疑似回線は、ローカルとリモートにMPLSスタティックラベルを設定することでスタティック AToM 疑似回線になります。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **vfi** { *vfi-name* }
6. **neighbor** { *A.B.C.D* } { **pw-id** *value* }
7. **mpls static label** { **local** *value* } { **remote** *value* }
8. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

### ステップ 1 configure

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

### ステップ 2 l2vpn

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

### ステップ 3 bridge group *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

### ステップ 4 bridge-domain *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **vfi** { *vfi-name* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# vfi v1
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)#
```

仮想転送インターフェイス（VFI）パラメータを設定し、L2VPN ブリッジ グループ ブリッジ ドメイン VFI コンフィギュレーション モードを開始します。

**ステップ 6** **neighbor** { *A.B.C.D* } { **pw-id** *value* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)# neighbor 10.1.1.2 pw-id 1000
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-pw)#
```

疑似回線ポートをブリッジ ドメインに追加するか、または疑似回線を仮想転送インターフェイス（VFI）に追加します。

- 相互接続ピアの IP アドレスを指定するには、*A.B.C.D* 引数を使用します。
- 疑似回線 ID および ID 値を設定するには、 **pw-id** キーワードを使用します。指定できる範囲は 1 ～ 4294967295 です。

**ステップ 7** **mpls static label** { **local** *value* } { **remote** *value* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-pw)# mpls static label local 800 remote 500
```

MPLS スタティック ラベルおよび疑似回線コンフィギュレーションのスタティック ラベルを設定します。ローカルおよびリモートの疑似回線ラベルを設定できます。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# 仮想転送インスタンスのディセーブル化

VFI をディセーブルにするには、次の作業を実行します。VFI がディセーブルの場合、VFI に関連付けられた、以前に確立された疑似回線はすべて切断されます。LDP アドバタイズメントは、VFI に関連付けられた MAC アドレスを回収するために送信されます。ただし、シャットダウン後にも引き続き接続回線を VFI に接続したり切断したりできます。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge group name*
4. **bridge-domain** *bridge-domain name*
5. **vfi** { *vfi-name* }
6. **shutdown**
7. **commit** コマンドまたは **end** コマンドを使用します。
8. **show l2vpn bridge-domain** [ **detail** ]

## 手順の詳細

### ステップ 1 configure

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

### ステップ 2 l2vpn

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

### ステップ 3 bridge group *bridge group name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

### ステップ 4 bridge-domain *bridge-domain name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **vfi** { *vfi-name* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# vfi v1
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)#
```

仮想転送インターフェイス（VFI）パラメータを設定し、L2VPN ブリッジ グループ ブリッジ ドメイン VFI コンフィギュレーション モードを開始します。

**ステップ 6** **shutdown**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)# shutdown
```

仮想転送インターフェイス（VFI）をディセーブルにします。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

**ステップ 8** **show l2vpn bridge-domain** [ **detail** ]

**例**：

```
RP/0/RSP0/cpu 0: router# show l2vpn bridge-domain detail
```

VFI の状態を表示します。たとえば、VFI をシャットダウンすると、VFI はブリッジ ドメインでシャットダウンされていると示されています。

# MAC アドレス関連パラメータの設定

次のトピックでは、MAC アドレス関連パラメータの設定方法について説明します。

MAC テーブル属性は、ブリッジ ドメインについて設定されます。

# MAC アドレスの送信元ベースの学習の設定

MAC アドレスの送信元ベースの学習を設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge group name*
4. **bridge-domain** *bridge-domain-name*
5. **mac**
6. **learning disable**
7. **commit** コマンドまたは **end** コマンドを使用します。
8. **show l2vpn bridge-domain** [ **detail** ]

## 手順の詳細

### ステップ 1 configure

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

### ステップ 2 l2vpn

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

### ステップ 3 bridge group *bridge group name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

### ステップ 4 bridge-domain *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
```

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **mac**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# mac
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-mac)#
```

L2VPN ブリッジ グループ ブリッジ ドメイン MAC コンフィギュレーション モードを開始します。

**ステップ 6** **learning disable**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-mac)# learning disable
```

ブリッジ ドメイン レベルで MAC 学習をディセーブルにします。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

**ステップ 8** **show l2vpn bridge-domain** [ **detail** ]

例：

```
RP/0/RSP0/cpu 0: router# show l2vpn bridge-domain detail
```

MAC アドレスの送信元ベースの学習がブリッジでディセーブルになったことの詳細が表示されます。

## MAC アドレス取り消しの有効化

指定されたブリッジ ドメインの MAC アドレス取り消しを有効にするには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*

4. **bridge-domain** *bridge-domain-name*
5. **mac**
6. **withdrawal**
7. **commit** コマンドまたは **end** コマンドを使用します。
8. **show l2vpn bridge-domain [detail]**

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain** *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、L2VPN l2vpn ブリッジ グループ ブリッジ ドメイン設定モードを開始します。

**ステップ 5** **mac**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# mac
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-mac)#
```

L2VPN l2vpn ブリッジ グループ ブリッジ ドメイン MAC 設定モードを開始します。

**ステップ 6** **withdrawal**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-mac)# withdrawal
```

特定のブリッジ ドメインについて MAC アドレス取り消しを有効にします。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

**ステップ 8** **show l2vpn bridge-domain [detail]**

例：

```
RP/0/RSP0/cpu 0: router# show l2vpn bridge-domain detail
```

MACアドレス取り消しを有効にすることを指定する詳細な出力例が表示されます。また、出力例には、疑似回線から送信または受信した MAC 回収メッセージの数が表示されます。

## MAC アドレス制限の設定

MAC アドレス制限のパラメータを設定するには、次の作業を実行します。

（注） MAC アドレス制限のアクションは AC でのみサポートされ、コア疑似回線ではサポートされません。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge group name*
4. **bridge-domain** *bridge-domain name*
5. （任意） **interface** *type interface_id*
6. **mac**
7. **limit**
8. **maximum** { *value* }

9. **action** { **flood** | **no-flood** | **shutdown** }
10. **notification** { **both** | **none** | **trap** }
11. **mac limit threshold** *80*
12. **commit** コマンドまたは **end** コマンドを使用します。
13. **show l2vpn bridge-domain** [ **detail** ]

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge group name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain** *bridge-domain name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** （任意） **interface** *type interface_id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# interface gigabitEthernet 0/2/0/1
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)#
```

指定したインターフェイスのインターフェイス コンフィギュレーション モードを開始し、このインターフェイスをブリッジ ドメイン メンバー インターフェイスとして追加します。

（注） 特定のインターフェイスに対してのみ MAC アドレス制限を設定する場合は、この手順を実行します。以降の手順では、MACアドレス制限をブリッジ ドメイン レベルで設定するためのルータ プロンプトを示します。ルータ プロンプトはこの手順をスキップした場合に表示されます。

**ステップ 6** **mac**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# mac
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-mac)#
```

L2VPN ブリッジ グループ ブリッジ ドメイン MAC コンフィギュレーション モードを開始します。

**ステップ 7** **limit**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-mac)# limit
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-mac-limit)#
```

アクション、最大、通知の MAC アドレス制限を設定し、L2VPN ブリッジ グループ ブリッジ ドメイン MAC 制限コンフィギュレーション モードを開始します。

**ステップ 8** **maximum** { *value* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-mac-limit)# maximum 5000
```

ブリッジで学習される MAC アドレスの数が制限に到達したときの特定のアクションを設定します。

**ステップ 9** **action** { **flood** | **no-flood** | **shutdown** }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-mac-limit)# action flood
```

学習される MAC アドレスの数が設定された MAC 制限を超えたときのブリッジの動作を設定します。

**ステップ 10** **notification** { **both** | **none** | **trap** }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-mac-limit)# notification both
```

学習されるMACアドレスの数が設定された制限を超えたときに送信される通知のタイプを指定します。

**ステップ 11** **mac limit threshold** *80*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# mac limit threshold 80
```

MAC 制限のしきい値を設定します。デフォルトは、ステップ 8 で設定した MAC アドレス制限の 75% です。

**ステップ 12** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

**ステップ 13** **show l2vpn bridge-domain** [ **detail** ]

例：

```
RP/0/RSP0/cpu 0: router# show l2vpn bridge-domain detail
```

MAC アドレス制限の詳細が表示されます。

## MAC アドレス エージングの設定

MAC アドレス エージングのパラメータを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **mac**
6. **aging**
7. **time** { *seconds* }
8. **commit** コマンドまたは **end** コマンドを使用します。
9. **show l2vpn bridge-domain** [ **detail** ]

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

**例**：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain** *bridge-domain-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **mac**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# mac
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-mac)#
```

L2VPN ブリッジ グループ ブリッジ ドメイン MAC コンフィギュレーション モードを開始します。

**ステップ 6** **aging**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-mac)# aging
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-mac-aging)#
```

MAC エージング コンフィギュレーション サブモードを開始し、時間やタイプなどのエージング パラメータを設定します。

ASR 9000 イーサネットおよび ASR 9000 拡張イーサネット ライン カードの最大 MAC 経過時間は 2 時間です。

**ステップ 7** **time** { *seconds* }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-mac-aging)# time 300
```

最大エージング タイムを設定します。

- MAC アドレス テーブル エントリの最大経過時間を指定するには、*seconds* 引数を使用します。範囲は 300 ～ 30000 秒です。エージング タイムは最後にスイッチが MAC アドレスを検出した時点からカウントされます。デフォルト値は 300 秒です。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

**ステップ 9** **show l2vpn bridge-domain** [ **detail** ]

例：

```
RP/0/RSP0/cpu 0: router# show l2vpn bridge-domain detail
```

エージング フィールドに関する詳細を表示します。

## ブリッジ ポート レベルでの MAC フラッシュのディセーブル化

ブリッジ ドメイン レベルで MAC フラッシュをディセーブルにするには、次の作業を実行します。

ブリッジ ドメインまたはブリッジ ポート レベルで MAC フラッシュをディセーブルにできます。デフォルトでは、そのポートが機能しなくなると、特定のポートで学習される MAC はただちにフラッシュされます。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **mac**

6. **port-down flush disable**
7. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain** *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、l2vpn ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **mac**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# mac
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-mac)#
```

l2vpn ブリッジ グループ ブリッジ ドメイン MAC コンフィギュレーション モードを開始します。

**ステップ 6** **port-down flush disable**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-mac)#
port-down flush disable
```

ブリッジ ポートが機能しなくなったら、MAC フラッシュをディセーブルにします。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## MAC アドレスのセキュリティの設定

MAC アドレスのセキュリティを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group***bridge-group-name*
4. **bridge-domain***bridge-domain-name*
5. **neighbor** { *A.B.C.D* } { **pw-id** *value* }
6. **mac**
7. **secure [action | disable | logging]**
8. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group***bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain***bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジドメインを確立し、L2VPN l2vpn ブリッジグループブリッジドメイン設定モードを開始します。

**ステップ 5** **neighbor** { *A.B.C.D* } { **pw-id** *value* }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# neighbor 10.1.1.2 pw-id 1000
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-pw)#
```

アクセス疑似回線ポートをブリッジ ドメインに追加するか、または疑似回線をブリッジ仮想転送インターフェイス（VFI）に追加します。

- 相互接続ピアの IP アドレスを指定するには、A.B.C.D 引数を使用します。
- 疑似回線 ID および ID 値を設定するには、pw-id キーワードを使用します。指定できる範囲は 1 ～ 4294967295 です。

**ステップ 6** **mac**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-pw)# mac
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-pw-mac)#
```

L2VPN l2vpn ブリッジ グループ ブリッジ ドメイン MAC 設定モードを開始します。

**ステップ 7** **secure [action | disable | logging]**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-pw-mac)#
secure
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-pw-mac-
secure)#
```

MAC セキュア コンフィギュレーション モードを開始します。

デフォルトでは、ブリッジの下のブリッジ ポート（インターフェイスおよびアクセス疑似回線）は親ブリッジからセキュリティ設定を継承します。

（注） ブリッジポートがダウンした後に、ブリッジポートを起動するには **clear** コマンドを実行する必要があります。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# AC スプリット ホライズン グループへの接続回線の設定

次の手順では、ブリッジドメインの接続回線（AC）のスプリットホライズン グループにインターフェイスを追加する方法を示します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group***bridge-group-name*
4. **bridge-domain***bridge-domain-name*
5. **interface** *type instance*
6. **split-horizon group**
7. **commit**
8. **end**
9. **show l2vpn bridge-domain detail**

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group***bridge-group-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group metroA
```

名前付きブリッジ グループのコンフィギュレーション モードを開始します。

**ステップ 4** **bridge-domain***bridge-domain-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain east
```

名前付きブリッジ ドメインのコンフィギュレーション モードを開始します。

**ステップ 5** **interface** *type instance*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# interface GigabitEthernet0/1/0/6
```

指定されたインターフェイスのコンフィギュレーション モードを開始します。

**ステップ 6** **split-horizon group**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)# split-horizon group
```

指定されたインターフェイスのコンフィギュレーション モードを開始します。

**ステップ 7** **commit**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)# commit
```

設定変更を保存します。

**ステップ 8** **end**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)# end
```

EXEC モードに戻ります。

**ステップ 9** **show l2vpn bridge-domain detail**

**例**：

```
RP/0/RSP0/cpu 0: router show l2vpn bridge-domain detail
```

各 AC が AC スプリット ホライズン グループに属しているかどうかを含め、ブリッジに関する情報を表示します。

# AC スプリット ホライズン グループへのアクセス疑似回線の追加

次の手順では、ブリッジドメインの接続回線（AC）のスプリット ホライズン グループのメンバーとしてアクセス疑似回線を追加する方法を示します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **neighbor** *A.B.C.D* **pw-id** *pseudowire-id*
6. **split-horizon group**
7. **commit**
8. **end**
9. **show l2vpn bridge-domain detail**

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group metroA
```

名前付きブリッジ グループのコンフィギュレーション モードを開始します。

**ステップ 4** **bridge-domain** *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain east
```

名前付きブリッジ ドメインのコンフィギュレーション モードを開始します。

**ステップ 5** **neighbor** *A.B.C.D* **pw-id** *pseudowire-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# neighbor 10.2.2.2 pw-id 2000
```

疑似回線セグメントを設定します。

**ステップ 6** **split-horizon group**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)# split-horizon group
```

AC のスプリット ホライズン グループにこのインターフェイスを追加します。AC の唯一のスプリット ホライズン グループ

（注） ブリッジドメインごとに AC とアクセス疑似回線のスプリット ホライズン グループは 1 つだけサポートされます

。

**ステップ 7** **commit**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-pw) commit
```

設定変更を保存します。

**ステップ 8** **end**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)# end
```

EXEC モードに戻ります。

**ステップ 9** **show l2vpn bridge-domain detail**

**例**：

```
RP/0/RSP0/cpu 0: router # show l2vpn bridge-domain detail
```

各 AC が AC スプリット ホライズン グループに属しているかどうかを含め、ブリッジに関する情報を表示します。

# BGP オートディスカバリおよびシグナリングでの VPLS の設定

BGP ベースのオートディスカバリとシグナリングを設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge group name*
4. **bridge-domain** *bridge-domain name*
5. **vfi** { *vfi-name* }
6. **vpn-id** *vpn-id*
7. **autodiscovery bgp**
8. **rd** { *as-number:nn* | *ip-address:nn* | **auto** }
9. **route-target** { *as-number:nn* | *ip-address:nn* | **export** | **import** }
10. **route-target import** { *as-number:nn* | *ip-address:nn* }
11. **route-target export** { *as-number:nn* | *ip-address:nn* }
12. **signaling-protocol bgp**
13. **ve-id** { *number*}
14. **ve-range** { *number* }
15. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge group name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group metroA
```

名前付きブリッジ グループのコンフィギュレーション モードを開始します。

**ステップ 4** **bridge-domain** *bridge-domain name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain east
```

名前付きブリッジ ドメインのコンフィギュレーション モードを開始します。

**ステップ 5** **vfi** { *vfi-name* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# vfi vfi-east
```

仮想転送インスタンス（VFI）コンフィギュレーション モードを開始します。

**ステップ 6** **vpn-id** *vpn-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)# vpn-id 100
```

VPLS サービスの ID を指定します。VPN ID は、PE ルータ内でグローバルに一意である必要があります。つまり、同じ PE ルータ上の複数の VFI に同じ VPN ID を存在させることはできません。また、VFI に指定できる VPN ID は 1 つだけです。

**ステップ 7** **autodiscovery bgp**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)# autodiscovery bgp
```

すべての BGP オートディスカバリ パラメータが設定される BGP オートディスカバリ コンフィギュレーション モードを開始します。

このコマンドは、少なくとも VPN ID とシグナリング プロトコルが設定されるまで、BGP にプロビジョニングされません。

**ステップ 8** **rd** { *as-number:nn* | *ip-address:nn* | **auto** }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad)# rd auto
```

VFI でルート識別子（RD）を指定します。

RD は、VFI を識別するために BGP NLRI で使用されます。VFI ごとに RD を 1 つだけ設定できます。**rd auto** を除き、RD は同じ PE の複数の VFI で設定できません。

**rd auto** が設定されている場合、RD 値は、{BGP ルータ ID}:{自動生成の一意の 16 ビットインデックス} の形式になります。

**ステップ 9** **route-target** { *as-number:nn* | *ip-address:nn* | **export** | **import** }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad)# route-target 500:99
```

VFI のルート ターゲット（RT）を指定します。

PE 間の BGP オートディスカバリを設定するには、少なくとも 1 つのインポートと 1 つのエクスポート ルート ターゲット（または両方のロールを持つ 1 つのルート ターゲットだけ）を各 PE で設定する必要があります。

export または import キーワードが指定されていない場合、RT はインポートおよびエクスポートの両方であることを意味します。VFI には、複数のエクスポートまたはインポート RT を設定できます。ただし、同じ PE の複数の VFI で、同じ RT を使用することはできません。

**ステップ 10** **route-target import** { *as-number:nn* | *ip-address:nn* }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad)# route-target import 200:20
```

VFI のインポート ルート ターゲットを指定します。

インポート ルート ターゲットは、PE が受信した NLRI の RT と比較する項目です。RT が同じ VPLS サービスに属することを判断するには、受信した NLRI の RT がインポート RT と一致する必要があります。

**ステップ 11** **route-target export** { *as-number:nn* | *ip-address:nn* }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad)# route-target export 100:10
```

VFI のエクスポート ルート ターゲットを指定します。

エクスポート ルート ターゲットは、他の PE にアドバタイズされる NLRI 内に含められる RT です。

**ステップ 12** **signaling-protocol bgp**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad)# signaling-protocol bgp
```

BGP シグナリングをイネーブルにして、BGP シグナリング パラメータが設定される BGP シグナリング コンフィギュレーション サブモードを開始します。

このコマンドは、VE ID と VE ID の範囲が設定されるまで BGP にプロビジョニングされません。

**ステップ 13** **ve-id** { *number*}

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad-sig)# ve-id 10
```

VPLS を設定するために VFI のローカル PE ID を指定します。

VE ID は、VPLS サービス内の VFI を識別します。これは、同じ VPLS サービスの VFI が同じ VE ID を共有できないことを意味します。VE ID のスコープは、ブリッジ ドメイン内だけに存在します。したがって、PE 内の異なるブリッジ ドメインの VFI は、同じ VE ID を使用できます。

**ステップ 14** **ve-range** { *number* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad-sig)# ve-range 40
```

VPLS エッジ（VE）ブロックの最小サイズを上書きします。

デフォルトの最小サイズは 10 です。設定する VE の範囲は、10 よりも高い必要があります。

**ステップ 15** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# BGP オートディスカバリおよび LDP シグナリングでの VPLS の設定

BGP ベースのオートディスカバリとシグナリングを設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **router-id** *ip-address*
4. **bridge group** *bridge-group-name*
5. **bridge-domain** *bridge-domain-name*
6. **transport-mode vlan passthrough**
7. **vfi** {*vfi-name*}
8. **autodiscovery bgp**
9. **vpn-id** *vpn-id*
10. **rd** {*as-number:nn* | *ip-address:nn* | **auto**}
11. **route-target** {*as-number:nn* | *ip-address:nn* | **export** | **import** }
12. **route-target import** {*as-number:nn* | *ip-address:nn*}
13. **route-target export** {*as-number:nn* | *ip-address:nn*}

14. **signaling-protocol ldp**
15. **vpls-id** {*as-number:nn* | *ip-address:nn*}
16. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/CPU0:router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **router-id** *ip-address*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# router-id 1.1.1.1
```

プロバイダー エッジ（PE）ルータの一意のレイヤ 2（L2）ルータ ID を指定します。

ルータ IDは、LDPシグナリング用に設定する必要があり、BGPNLRI、SAII（ローカルL2ルータ ID）、およびTAII（リモートL2ルータ ID）でL2ルータ IDとして使用されます。IPv4 アドレス形式の任意の値を使用できます。

（注） 各 PE には一意の L2 ルータ ID が必要です。PE が LDP ルータ ID を使用して自動的に L2 ルータ ID を生成するため、この CLI はオプションです。

**ステップ 4** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group metroA
```

名前付きブリッジ グループのコンフィギュレーション モードを開始します。

**ステップ 5** **bridge-domain** *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain east
```

名前付きブリッジ ドメインのコンフィギュレーション モードを開始します。

**ステップ 6** **transport-mode vlan passthrough**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# transport-mode vlan passthrough
```

BGP 自動検出のために VC タイプ 4 を有効にします。

**ステップ 7** **vfi** {*vfi-name*}

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# vfi vfi-east
```

仮想転送インスタンス（VFI）コンフィギュレーション モードを開始します。

**ステップ 8** **autodiscovery bgp**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)# autodiscovery bgp
```

すべての BGP オートディスカバリ パラメータが設定される BGP オートディスカバリ コンフィギュレーション モードを開始します。

このコマンドは、少なくとも VPN ID とシグナリング プロトコルが設定されるまで、BGP にプロビジョニングされません。

**ステップ 9** **vpn-id** *vpn-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)# vpn-id 100
```

VPLS サービスの ID を指定します。VPN ID は、PE ルータ内でグローバルに一意である必要があります。つまり、同じ PE ルータ上の複数の VFI に同じ VPN ID を存在させることはできません。また、VFI に指定できる VPN ID は 1 つだけです。

**ステップ 10** **rd** {*as-number:nn* | *ip-address:nn* | **auto**}

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad)# rd auto
```

VFI でルート識別子（RD）を指定します。

RD は、VFI を識別するために BGP NLRI で使用されます。VFI ごとに RD を 1 つだけ設定できます。**rd auto** を除き、RD は同じ PE の複数の VFI で設定できません。

**rd auto** が設定されている場合、RD 値は、{BGP ルータ ID}:{自動生成の一意の 16 ビットインデックス} の形式になります。

**ステップ 11** **route-target** {*as-number:nn* | *ip-address:nn* | **export** | **import** }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad)# route-target 500:99
```

VFI のルート ターゲット（RT）を指定します。

PE 間の BGP オートディスカバリを設定するには、少なくとも 1 つのインポートと 1 つのエクスポート ルート ターゲット（または両方のロールを持つ 1 つのルート ターゲットだけ）を各 PE で設定する必要があります。

export または import キーワードが指定されていない場合、RT はインポートおよびエクスポートの両方であることを意味します。VFI には、複数のエクスポートまたはインポート RT を設定できます。ただし、同じ PE の複数の VFI で、同じ RT を使用することはできません。

**ステップ 12** **route-target import** {*as-number:nn* | *ip-address:nn*}

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad)# route-target import 200:20
```

VFI のインポート ルート ターゲットを指定します。

インポート ルート ターゲットは、PE が受信した NLRI の RT と比較する項目です。RT が同じ VPLS サービスに属することを判断するには、受信した NLRI の RT がインポート RT と一致する必要があります。

**ステップ 13** **route-target export** {*as-number:nn* | *ip-address:nn*}

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad)# route-target export 100:10
```

VFI のエクスポート ルート ターゲットを指定します。

エクスポート ルート ターゲットは、他の PE にアドバタイズされる NLRI 内に含められる RT です。

**ステップ 14** **signaling-protocol ldp**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad)# signaling-protocol ldp
```

LDP シグナリングをイネーブルにします。

**ステップ 15** **vpls-id** {*as-number:nn* | *ip-address:nn*}

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad-sig)# vpls-id 10:20
```

シグナリング中に VPLS ドメインを識別する VPLS ID を指定します。

デフォルトの VPLS ID は BGP の ASN および設定済みの VPN ID を使用して自動的に生成されるため、同じ自律システム内にある（同じ ASN を共有する）すべての PE ではこのコマンドはオプションです（つまり、デフォルトの VPLS ID は ASN:VPN-ID です）。4 バイトの ASN を使用する場合は、VPLS ID を作成するために、ASN の下位 2 バイトが使用されます。InterAS の場合、VPLS ID を明示的に設定する必要があります。VFI ごとに 1 つの VPLS ID だけを設定でき、同じ VPLS ID を複数の VFI には使用できません。

**ステップ 16** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# サービスパス設定の設定

サービスパス設定を設定するには、次のタスクを実行します。

## ルートポリシーの転送クラスの設定

次に、ルートポリシーで転送クラスを設定する方法を示します。

```
route-policy  fwd1
      set forward-class 1
end-policy
!
route-policy fwd2
      set forward-class 2
 end-policy
!
```

## テーブルポリシー付加ポイントでのルートポリシーの付加

次に、VPLS ブリッジドメイン VFI のテーブルポリシー付加ポイントにルートポリシーを付加する設定を示します。

```
config
  l2vpn
    bridge group bg1
     bridge-domain bd1
      vfi v1
       autodiscovery bgp
        table-policy fwd1
!
```

次に、EVPN EVI のテーブルポリシー付加ポイントにルートポリシーを付加する設定を示します。

```
config
  l2vpn
    bridge group pbb
     bridge-domain core1
      pbb core
       evi 1
!
 bridge group edge
  bridge-domain edge1
   pbb edge i-sid 256 core-bridge core1
!
 evpn
  evi 1
    bgp
     table-policy fwd2
!
```

## TE トンネルと転送クラスインデックスの関連付け

次に、TE トンネルを転送クラスインデックスに関連付ける設定を示します。

```
 config
  interface tunnel-te1
   ipv4 unnumbered Loopback0
   autoroute announce
   destination 10.10.10.10
   forward-class 1
   path-option 10 explicit name PATH1
!
```

# BGP 自動検出を使用した L2VPN VPLS のルートポリシーの有効化

BGP 自動検出設定を使用して L2VPN VPLS のルートポリシーを有効にするには、次の作業を実行します。ルートポリシーエクスポートのみがサポートされています。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge group name*
4. **bridge-domain** *bridge-domain name*
5. **vfi** { *vfi-name* }
6. **autodiscovery bgp**
7. **route-policy export** *policy-name*
8. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

**例**：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

**例**：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge group name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group bg1
```

名前付きブリッジ グループのコンフィギュレーション モードを開始します。

**ステップ 4** **bridge-domain** *bridge-domain name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain bd1
```

名前付きブリッジ ドメインのコンフィギュレーション モードを開始します。

**ステップ 5** **vfi** { *vfi-name* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# vfi vfi-east
```

仮想転送インスタンス（VFI）コンフィギュレーション モードを開始します。

**ステップ 6** **autodiscovery bgp**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)# autodiscovery bgp
```

すべての BGP オートディスカバリ パラメータが設定される BGP オートディスカバリ コンフィギュレーション モードを開始します。

このコマンドは、少なくとも VPN ID とシグナリング プロトコルが設定されるまで、BGP にプロビジョニングされません。

**ステップ 7** **route-policy export** *policy-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad)# route-policy export RPL_1
```

**route-policy export** 付加ポイントにルートポリシーを付加します。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

例

# BGP 自動検出を使用した L2VPN VPWS のルートポリシーの有効化

BGP 自動検出設定を使用して L2VPN VPWS のルートポリシーを有効にするには、次の作業を実行します。ルートポリシーエクスポートのみがサポートされています。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group** *xconnect group name*
4. **mp2mp** *mp2mp instance name*
5. **autodiscovery bgp**
6. **route-policy export** *policy-name*
7. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

**例**：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **xconnect group** *xconnect group name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group xg1
```

名前付きクロスコネクトグループの設定モードを開始します。

**ステップ 4** **mp2mp** *mp2mp instance name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# mp2mp mp1
```

名前付き mp2mp インスタンスを作成します。

**ステップ 5** **autodiscovery bgp**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-mp2mp)# autodiscovery bgp
```

すべての BGP オートディスカバリ パラメータが設定される BGP オートディスカバリ コンフィギュレーション モードを開始します。

このコマンドは、少なくとも VPN ID とシグナリング プロトコルが設定されるまで、BGP にプロビジョニングされません。

**ステップ 6** **route-policy export** *policy-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-mp2mp-ad)# route-policy export RPL_2
```

**route-policy export** 付加ポイントにルートポリシーを付加します。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。

• [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
• [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

例

# G.8032 イーサネット リング保護の設定

G.8032 動作を設定するには、次のものを別個に設定します。

- 次のものを示す ERP インスタンス：
  - APS チャネルとして使用する（サブ）インターフェイス
  - CFM によって監視する（サブ）インターフェイス）
  - インターフェイスが RPL リンクであるかどうか、RPL リンクである場合は RPL ノード タイプ
- リング リンクを監視する EFD による CFM

（注） 各モニタ リンクの MEP は、別のメンテナンス アソシエーションで設定する必要があります。

- レイヤ 2 トポロジを作成するブリッジ ドメイン。RAPS チャネルは、データ ブリッジ ドメインから分離した専用の管理ブリッジ ドメインで設定されます。
- デフォルト値と異なる場合は、ERP インスタンスに適用される動作の特性。これは任意です。

この項では、次の内容について説明します。

## ERP プロファイルの設定

イーサネット リング保護（ERP）プロファイルを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **Ethernet ring g8032 profile** *profile-name*
3. **timer** { **wtr** | **guard** | **hold-off** } *seconds*
4. **non-revertive**
5. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

**例**：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **Ethernet ring g8032 profile** *profile-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config)# Ethernet ring g8032 profile p1
```

G.8032 リング モードをイネーブルにし、G.8032 コンフィギュレーション サブモードを開始します。

**ステップ 3** **timer** { **wtr** | **guard** | **hold-off** } *seconds*

**例**：

```
RP/0/RSP0/cpu 0: router(config-g8032-ring-profile)# timer hold-off 5
```

ガード、hold-off、および wait-to-restore タイマーの間隔（秒単位）を指定します。

**ステップ 4** **non-revertive**

**例**：

```
RP/0/RSP0/cpu 0: router(config-g8032-ring-profile)# non-revertive
```

非リバーティブ リング インスタンスを指定します。

**ステップ 5** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## CFM MEP の設定

イーサネット接続障害管理（CFM）の詳細については、『Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Configuration Guide』の「Configuring Ethernet OAM on the Cisco ASR 9000 Series Router」モジュールを参照してください。

# ERP インスタンスの設定

ERP インスタンスを設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *domain-name*
5. **interface** *type port0-interface-path-id.subinterface*
6. **interface** *type port1-interface-path-id.subinterface*
7. **bridge-domain** *domain-name*
8. **interface** *type interface-path-id.subinterface*
9. **ethernet ring g8032** *ring-name*
10. **instance** *number*
11. **description** *string*
12. **profile** *profile-name*
13. **rpl { port0 | port1 } { owner | neighbor | next-neighbor }**
14. **inclusion-list vlan-ids** *vlan-id*
15. **aps-channel**
16. **level** *number*
17. **port0 interface** *type path-id*
18. **port1** { **interface** *type interface-path-id* | **bridge-domain** *bridge-domain-name* | **xconnect** *xconnect-name* | **none** }
19. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを含めることができるブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain** *domain-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain bd1
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

R-APS チャネルのブリッジ ドメインを設定し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **interface** *type port0-interface-path-id.subinterface*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# interface GigabitEthernet 0/0/0/0.1
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)#
```

インターフェイス コンフィギュレーション モードを開始し、同じブリッジ ドメインに属する他のインターフェイスからパケットを転送および受信できるブリッジ ドメインにインターフェイスを追加します。

**ステップ 6** **interface** *type port1-interface-path-id.subinterface*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# interface GigabitEthernet 0/0/0/1.1
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)#
```

インターフェイス コンフィギュレーション モードを開始し、同じブリッジ ドメインに属する他のインターフェイスからパケットを転送および受信できるブリッジ ドメインにインターフェイスを追加します。

**ステップ 7** **bridge-domain** *domain-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain bd2
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

データ トラフィックのブリッジ ドメインを設定し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 8** **interface** *type interface-path-id.subinterface*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# interface GigabitEthernet 0/0/0/0.10
```

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)#
```

インターフェイス コンフィギュレーション モードを開始し、同じブリッジ ドメインに属する他のインターフェイスからパケットを転送および受信できるブリッジドメインにインターフェイスを追加します。

**ステップ 9** **ethernet ring g8032** *ring-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# ethernet ring g8032 r1
```

G.8032 リング モードをイネーブルにし、G.8032 コンフィギュレーション サブモードを開始します。

**ステップ 10** **instance** *number*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-erp)# instance 1
```

イーサネット リング G.8032 インスタンス コンフィギュレーション サブモードを開始します。

**ステップ 11** **description** *string*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-erp-instance)# description test
```

このインスタンスの説明として機能するストリングを指定します。

**ステップ 12** **profile** *profile-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-erp-instance)#profile p1
```

関連するイーサネット リング G.8032 プロファイルを指定します。

**ステップ 13** **rpl { port0 | port1 } { owner | neighbor | next-neighbor }**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-erp-instance)#rpl port0 neighbor
```

RPL オーナー、ネイバー、または次のネイバーとしてローカル ノードのリング ポートを 1 つ指定します。

**ステップ 14** **inclusion-list vlan-ids** *vlan-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-erp-instance)# inclusion-list vlan-ids e-g
```

現在のインスタンスと一連の VLAN ID を関連付けます。

**ステップ 15** **aps-channel**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-erp-instance)# aps-channel
```

イーサネット リング G.8032 インスタンス aps-channel コンフィギュレーション サブモードを開始します。

**ステップ 16** **level** *number*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-erp-instance-aps)# level 5
```

APS メッセージ レベルを指定します。範囲は 0 ～ 7 です。

**ステップ 17** **port0 interface** *type path-id*

**例**：

```
RP/0/RSP0/cpu 0: router(configl2vpn-erp-instance-aps)# port0 interface GigabitEthernet 0/0/0/0.1
```

G.8032 APS チャネル インターフェイスを port0 に関連付けます。

**ステップ 18** **port1** { **interface** *type interface-path-id* | **bridge-domain** *bridge-domain-name* | **xconnect** *xconnect-name* | **none** }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-erp-instance-aps)# port1 interface GigabitEthernet 0/0/0/1.1
```

G.8032 APS チャネル インターフェイスを port1 に関連付けます。

**ステップ 19** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## ERP パラメータの設定

ERP パラメータを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **ethernet ring g8032** *ring-name*
4. **port0 interface***type interface-path-id*
5. **monitor port0 interface***type interface-path-id*
6. **exit**
7. **port1** { **interface** *type interface-path-id* | **virtual** | **none**}
8. **monitor port1 interface***type interface-path-id*
9. **exit**
10. **exclusion-list vlan-ids** *vlan-id*
11. **open-ring**
12. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **ethernet ring g8032** *ring-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# ethernet ring g8032 r1
```

G.8032 リング モードをイネーブルにし、G.8032 コンフィギュレーション サブモードを開始します。

**ステップ 4** **port0 interface***type interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-erp)# port0 interface GigabitEthernet 0/1/0/6
```

指定したポート（リング ポート）の G.8032 ERP をイネーブルにします。

**ステップ 5** **monitor port0 interface***type interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-erp-port0)# monitor port0 interface 0/1/0/2
```

リング ポートごとにリング リンク障害を検出するために監視するポートを指定します。モニタ対象インターフェイスは、メイン インターフェイスのサブインターフェイスでなければなりません。

**ステップ 6** **exit**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-erp-port0)# exit
```

port0 コンフィギュレーション サブモードを終了します。

**ステップ 7** **port1** { **interface** *type interface-path-id* | **virtual** | **none**}

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-erp)# port1 interface GigabitEthernet 0/1/0/8
```

指定したポート（リング ポート）の G.8032 ERP をイネーブルにします。

**ステップ 8** **monitor port1 interface***type interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-erp-port1)# monitor port1 interface 0/1/0/3
```

リング ポートごとにリング リンク障害を検出するために監視するポートを指定します。モニタ対象インターフェイスは、メイン インターフェイスのサブインターフェイスでなければなりません。

**ステップ 9** **exit**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-erp-port1)# exit
```

port1 コンフィギュレーション サブモードを終了します。

**ステップ 10** **exclusion-list vlan-ids** *vlan-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-erp)# exclusion-list vlan-ids a-d
```

イーサネット リング保護メカニズムによって保護されていない一連の VLAN ID を指定します。

**ステップ 11** **open-ring**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-erp)# open-ring
```

開いたリングとしてイーサネット リング G.8032 を指定します。

**ステップ 12** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# TCN 伝播の設定

トポロジ変更通知（TCN）の伝播を設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **tcn-propagation**
4. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **tcn-propagation**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# tcn-propagation
```

マイナー リングからメイン リング、および MSTP から G.8032 への TCN 伝播を許可します。

**ステップ 4** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# Flow Aware Transport 疑似回線の設定

この項では、次の内容について説明します。

## VPWS の ECMP および FAT PW によるロード バランシングのイネーブル化

VPWS の ECMP および FAT PW によるロード バランシングをイネーブルにするには、次の作業を実行します。L2VPN設定でPWクラスを作成すると、ロードバランシングが発生します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **pw-class** { *name* }
4. **encapsulation mpls**
5. **load-balancing flow-label** { **both** | **code** | **receive** | **transmit** } [ **static** ]
6. **exit**
7. **exit**
8. **xconnect group** *group-name*
9. **p2p** *xconnect-name*
10. **interface type** *interface-path-id*
11. **neighbor** *A.B.C.D* **pw-id** pseudowire-id
12. **pw-class** *class-name*
13. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **pw-class** { *name* }

例：

```
RP/0/RSP0/CPU0:router(config-l2vpn)# pw-class path1
```

疑似回線に使用する疑似回線クラス テンプレート名を設定します。

**ステップ 4** **encapsulation mpls**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc)# encapsulation mpls
```

MPLS に疑似回線カプセル化を設定します。

**ステップ 5** **load-balancing flow-label** { **both** | **code** | **receive** | **transmit** } [ **static** ]

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc-mpls)# load-balancing flow-label both
```

ECMP のロード バランシングをイネーブルにします。また、疑似回線のフロー ラベルのインポジションおよびディスポジションをイネーブルにします。

（注） static キーワードを指定しない場合は、FAT PW のエンドツーエンド ネゴシエーションが有効になります。

**ステップ 6** **exit**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc-mpls)#exit
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc)#
```

疑似回線カプセル化サブモードを終了し、ルータを親コンフィギュレーション モードに戻します。

**ステップ 7** **exit**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc)#exit
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

疑似回線サブモードを終了し、ルータを l2vpn 設定モードに戻します。

**ステップ 8** **xconnect group** *group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group grp1
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)#
```

相互接続グループの名前を指定します。

**ステップ 9** **p2p** *xconnect-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p vlan1
```

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)#
```

ポイントツーポイント クロスコネクトの名前を指定します。

**ステップ 10** **interface type** *interface-path-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface TenGigE 0/0/0/0.1
```

インターフェイス タイプとインスタンスを指定します。

**ステップ 11** **neighbor** *A.B.C.D* **pw-id** pseudowire-id

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor 10.2.2.2 pw-id 2000
```

クロスコネクトの疑似回線セグメントを設定します。

相互接続ピアの IP アドレスを指定するには、A.B.C.D 引数を使用します。

（注） A.B.C.D は再帰的または非再帰的プレフィックスです。

**ステップ 12** **pw-class** *class-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pw)# pw-class path1
```

この疑似回線を疑似回線クラスと関連付けます。

**ステップ 13** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## VPLS の ECMP および FAT PW によるロード バランシングのイネーブル化

VPLS の ECMP および FAT PW によるロードバランシングをイネーブルにするには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**

3. **load-balancing flow {src-dst-mac | src-dst-ip}**
4. **pw-class** { *class - name* }
5. **encapsulation mpls**
6. **load-balancing flow-label** { **both** | **code** | **receive** | **transmit** } [ **static** ]
7. **exit**
8. **bridge group** *bridge-group-name*
9. **bridge-domain** *bridge-domain-name*
10. **vfi** { *vfi-name* }
11. **autodiscovery bgp**
12. **signaling-protocol bgp**
13. **load-balancing flow-label** { **both** | **code** | **receive** | **transmit** } [ **static** ]
14. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **load-balancing flow {src-dst-mac | src-dst-ip}**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# load-balancing flow src-dst-ip
```

フローに基づくロード バランシングをイネーブルにします。

- **src-dst-mac**：ハッシュ用の送信元/宛先 MAC アドレスを使用します。
- **src-dst-ip**：ハッシュ用の送信元/宛先 IP アドレスを使用します。

（注） **load-balancing flow** コマンドは、**src-dst-ip** キーワードとともに使用することを推奨します。

**ステップ 4** **pw-class** { *class - name* }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# pw-class class1
```

この疑似回線を疑似回線クラスと関連付けます。

**ステップ 5** **encapsulation mpls**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc)# encapsulation mpls
```

MPLS に疑似回線カプセル化を設定します。

**ステップ 6** **load-balancing flow-label** { **both** | **code** | **receive** | **transmit** } [ **static** ]

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc-mpls)# load-balancing flow-label both
```

ECMP のロード バランシングをイネーブルにします。また、疑似回線のフロー ラベルのインポジションおよびディスポジションをイネーブルにします。

（注） **static** キーワードを指定しない場合は、**FAT PW** のエンドツーエンド ネゴシエーションがイネーブルになります。

**ステップ 7** **exit**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc-mpls)# exit
```

疑似回線カプセル化サブモードを終了し、ルータを親コンフィギュレーション モードに戻します。

**ステップ 8** **bridge group** *bridge-group-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group group1
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 9** **bridge-domain** *bridge-domain-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain domain1
```

ブリッジ ドメインを確立し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 10** **vfi** { *vfi-name* }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# vfi my_vfi
```

仮想転送インスタンス（VFI）コンフィギュレーション モードを開始します。

**ステップ 11** **autodiscovery bgp**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)# autodiscovery bgp
```

すべての BGP オートディスカバリ パラメータが設定される BGP オートディスカバリ コンフィギュレーション モードを開始します。

**ステップ 12** **signaling-protocol bgp**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad)# signaling-protocol bgp
```

BGP シグナリングをイネーブルにして、BGP シグナリング パラメータが設定される BGP シグナリング コンフィギュレーション サブモードを開始します。

**ステップ 13** **load-balancing flow-label** { **both** | **code** | **receive** | **transmit** } [ **static** ]

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad-sig)# load-balancing flow-label both static
```

ECMP のロードバランシングをイネーブルにします。また、疑似回線のフロー ラベルのインポジションおよびディスポジションをイネーブルにします。

**ステップ 14** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# 疑似回線ヘッドエンドの設定

疑似回線ヘッドエンド（PWHE）は、pw-ether メインインターフェイス、pw-ether サブインターフェイス、または pw-iw インターフェイスを設定することで作成されます。使用可能な PWHE タイプは、pw-ether メインインターフェイス、サブインターフェイス、および pw-iw インター

フェイスです。特に指定のない限り、インターフェイスという用語は、pw-ether メインインターフェイス、サブインターフェイス、および pw-iw インターフェイスに適用されます。

（注） PWHE イーサネット サブインターフェイスおよびインターワーキング インターフェイスは、リリース 5.1.1 以降でサポートされています。

PWHE を機能させるには、相互接続を完全に設定する必要があります。PWHE を機能させるための、その他のレイヤ 3（L3）パラメータ（VRF および IP アドレスなど）の設定は、任意で行います。ただし、レイヤ 3 サービスを動作可能にするには L3 機能が必要です（PW L3 の終端用）。

PWHE は IPv4 と IPv6 の両方のアドレスをサポートしています。

ここでは、次の内容について説明します。

## PWHE 設定の制限事項

PWHE 設定に関する制限事項は、次のとおりです。

1. 汎用インターフェイスリストのメンバーは、A-PE への ECMP パスリストのスーパーセットである必要があります。
2. 各 A-PE ネイバーアドレスでサポートされるのは、8 つの汎用インターフェイスリストだけです。
3. 汎用インターフェイスリストごとに 8 つのレイヤ 3 リンクがサポートされます。
4. PW-Ether インターフェイスのみを PWHE L2 または L3 サブインターフェイスとして設定できます。
5. PW-Ether メインインターフェイスを含むクロスコネクトは、VC タイプ 5 または VC タイプ 4 として設定できます。デフォルトでは、クロスコネクトは VC タイプ 5 として設定されています（または、pw-class transport-mode ethernet コマンドを使用して設定されます）。クロスコネクトを VC タイプ 4 として設定するには、**p2p neighbor tag-impose vlan id** コマンドと **pw-class transport-mode vlan** コマンドを使用します。
6. L3 PW-Ether サブインターフェイスが関連付けられている PW-Ether メインインターフェイスを含むクロスコネクトは、VC タイプ 5 でのみサポートされています。
7. PW-IW インターフェイスを含むクロスコネクトは、IPv4 および VC タイプ 11 でのみサポートされています。PW-IW インターフェイスは、IP インターワーキングに使用される L3 仮想インターフェイスです。クロスコネクトを VC タイプ 11 として設定するには、interworking ipv4 コマンドを使用します。
8. PW-Ether インターフェイスおよびサブインターフェイスは、IPv4 と IPv6 の両方で設定できます。
9. PW-IW インターフェイスは、IPv4 でのみ設定できます。
10. 疑似回線の冗長性、優先パス、ローカルスイッチングまたは L2TP は、PWHE で設定されたクロスコネクトに対してはサポートされません。
11. TE および LDP などのアプリケーションはインターフェイス タイプのチェックを行うため、PWHE を設定することはできません。

12. PWHE インターフェイス上では、アドレスファミリ、CDP、および MPLS は設定できません。
13. pw-iw インターフェイスでは、MAC アドレスはサポートされません。

## 汎用インターフェイスリストの設定

汎用インターフェイスリストを設定するには、次の作業を実行します。

（注） 各 A-PE ネイバーアドレスでサポートされるのは、8 つの汎用インターフェイスリストだけです。汎用インターフェイスリストごとに 8 つのレイヤ 3 リンクがサポートされます。ステップ 3 またはステップ 4 を繰り返して、インターフェイスを汎用インターフェイスリストに追加します。

### 手順の概要

1. **configure**
2. **generic-interface-list** *list-name*
3. **interface** *type interface-path-id*
4. **interface** *type interface-path-id*
5. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **generic-interface-list** *list-name*

例：

```
RP/0/RSP0/cpu 0: router(config)# generic-interface-list list1
```

汎用インターフェイスリストを設定します。

汎用インターフェイスリストを削除するには、このコマンドの no 形式（**no generic-interface-list** *list-name*）を使用します。

**ステップ 3** **interface** *type interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-if-list)# interface Bundle-Ether 100
```

指定されたインターフェイスを設定します。

**ステップ 4** **interface** *type interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-if-list)# interface Bundle-Ether 200
```

指定されたインターフェイスを設定します。

**ステップ 5** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## PWHE インターフェイスの設定

PWHE イーサネットおよびインターワーキング インターフェイスを設定する（つまり PWHE イーサネットおよびインターワーキングインターフェイスに汎用インターフェイスリストを付加する）には、次の作業を実行します。

### 手順の概要

1. **configure**
2. **interface pw-ether** *id* または **interface pw-iw** *id*
3. **attach generic-interface-list** *interface_list_name*
4. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface pw-ether** *id* または **interface pw-iw** *id*

例：

```
RP/0/RSP0/cpu 0: router(config)# interface pw-ether <id>
or
RP/0/RSP0/cpu 0: router(config)# interface pw-iw <id>
```

（**interface pw-ether** *id*）PWHE 疑似回線のメインまたはサブインターフェイスを設定し、インターフェイス設定モードを開始します。

（**interface pw-iw** *id*）PWHE 疑似回線インターワーキング インターフェイスを設定し、インターフェイス設定モードを開始します。

**ステップ 3** **attach generic-interface-list** *interface_list_name*

例：

```
RP/0/RSP0/cpu 0: router(config-if)# attach generic-interface-list interfacelist1
```

PW-Ether または PW-IW インターフェイスに汎用インターフェイスリストを付加します。PW-Ether または PW-IW インターフェイスから汎用インターフェイスリストを削除するには、このコマンドの **no** 形式（**no generic-interface-list** *list-name*）を使用します。

**ステップ 4** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## PWHE 相互接続の設定

PWHE 相互接続を設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **xconnect group** *group-name*
4. **p2p** *xconnect-name*
5. **interface pw-ether** *id* または **interface pw-iw** *id*
6. **neighbor** *ip-address* **pw-id** *value*
7. **pw-class** *class-name*
8. （*PW-IW のみ*）**interworking ipv4**
9. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

**例**：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 3** **xconnect group** *group-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group pw-he1
```

自由形式の 32 文字ストリングを使用して、相互接続グループ名を設定します。

**ステップ 4** **p2p** *xconnect-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p pw-hexconnect
```

P2P コンフィギュレーション サブモードを開始します。

**ステップ 5** **interface pw-ether** *id* または **interface pw-iw** *id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface pw-ether 100
or
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p )# interface pw-iw 100
```

PWHE インターフェイスを設定します。

**ステップ 6** **neighbor** *ip-address* **pw-id** *value*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor 10.165.200.25 pw-id 100
```

相互接続の疑似回線を設定します。

IP アドレスは、該当する PE ノードの IP アドレスです。

**pw-id** は PE ノードの **pw-id** と一致する必要があります。

**ステップ 7** **pw-class** *class-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pw)# pw-class dynamic_mpls
```

疑似回線クラス サブモードになり、疑似回線クラス テンプレートを定義できます。

（注） 疑似回線クラスは、VC4 および VC5 の L2VPN の下で次のように定義する必要があります。

```
pw-class vc_type_4
encapsulation mpls
transport-mode vlan
!
!
pw-class vc_type_5
encapsulation mpls
transport-mode ethernet
!
!
```

**ステップ 8** （*PW-IW* のみ）**interworking ipv4**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pw)# interworking ipv4
```

疑似回線の確立時に、VC タイプ 11 または IP インターワーキングモードを使用するようにクロスコネクト p2p エンティティを設定します。

**ステップ 9** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## 送信元アドレスの設定

ローカル送信元アドレスを設定するには、次の作業を実行します

（注） pw-class 送信元アドレスとしてサポートされるのは IPv4 のみです。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **pw-class** *class-name*
4. **encapsulation mpls**
5. **ipv4 source** *source-address*
6. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

**例**：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

**例**：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 3** **pw-class** *class-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# pw-class class1
```

疑似回線クラス サブモードになり、疑似回線クラス テンプレートを定義できます。

**ステップ 4** **encapsulation mpls**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc)# encapsulation mpls
```

MPLS に疑似回線カプセル化を設定します。

**ステップ 5** **ipv4 source** *source-address*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc-mpls)# ipv4 source 10.1.1.1
```

ローカル送信元 IPv4 アドレスを設定します。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。

• [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## PWHE インターフェイスのパラメータの設定

PWHE インターフェイスのパラメータを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **interface pw-ether** *id*（または）**interface pw-iw** *id*
3. **ipv4 address** *ip-address subnet-mask*（または）（PW-Ether のみ）**ipv6 address** *ipv6-prefix/prefix-length*
4. **attach generic-interface-list** *interface_list_name*
5. **l2overhead** *bytes*
6. **load-interval** *seconds*
7. **dampening** *decay-life*
8. **logging events link-status**
9. （PW-Ether メインインターフェイスのみ） **mac-address** *MAC address*
10. **mtu** *interface_MTU*
11. **bandwidth** *kbps*
12. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface pw-ether** *id*（または）**interface pw-iw** *id*

例：

```
RP/0/RSP0/cpu 0: router(config)# interface pw-ether <id>
or
RP/0/RSP0/cpu 0: router(config)# interface pw-iw <id>
```

（**interface pw-ether** *id*）PWHE インターフェイスを設定し、インターフェイス設定モードを開始します。

（**interface pw-iw** *id*）PWHE インターフェイスを設定し、インターフェイス設定モードを開始します。

**ステップ 3** **ipv4 address** *ip-address subnet-mask*（または）（PW-Ether のみ）**ipv6 address** *ipv6-prefix/prefix-length*

例：

```
RP/0/RSP0/cpu 0: router(config-if)# ipv4 address 40.1.1.2 255.255.255.0
```

インターフェイスの IPv4 または IPv6 アドレスを設定します。

**ステップ 4** **attach generic-interface-list** *interface_list_name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# attach generic-interface-list interfacelist1
```

汎用インターフェイスリストを PW-Ether または PW-IW インターフェイスに付加します。

**ステップ 5** **l2overhead** *bytes*

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# l2overhead 20
```

レイヤ 2 オーバーヘッドのサイズを設定します。

**ステップ 6** **load-interval** *seconds*

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# load-interval 90
```

インターフェイスの負荷計算の間隔（秒単位）を指定します。

間隔は次のとおりです。

- 0 に設定できます（0 は負荷計算をディセーブルにします）。
- 0 以外の場合は、30 ～ 600 の範囲の 30 の倍数で指定する必要があります。

**ステップ 7** **dampening** *decay-life*

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# dampening 10
```

特定のインターフェイスでのステート ダンプニングを設定します（分単位）。

**ステップ 8** **logging events link-status**

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# logging events link-status
```

インターフェイス ロギングごとに設定します。

**ステップ 9** （PW-Ether メインインターフェイスのみ） **mac-address** *MAC address*

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# mac-address aaaa.bbbb.cccc
```

インターフェイスの MAC アドレス（xxxx.xxxx.xxxx）を設定します。

**ステップ 10** **mtu** *interface_MTU*

例：

```
RP/0/RSP0/cpu 0: router(config-if)# mtu 128
```

インターフェイスの MTU を設定します。

**ステップ 11** **bandwidth** *kbps*

例：

```
RP/0/RSP0/cpu 0: router(config-if)# bandwidth 200
```

帯域幅を設定します。範囲は 0 ～ 4294967295 です。

**ステップ 12** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## PWHE レイヤ 2 サブインターフェイスを設定し、ブリッジドメインに追加する

PWHE レイヤ 2 サブインターフェイスを設定し、ブリッジドメインに追加するには、次の作業を実行します。

（注） レイヤ 2 サブインターフェイスには IP アドレスが含まれていないため、レイヤ 2 トランスポートモードで動作するように設定する必要があります。

### 手順の概要

1. **configure**
2. **interface pw-ether** *id*
3. **ipv4 address** *ip-address subnet-mask*（または）**ipv6 address** *ipv6-prefix/prefix-length*
4. **attach generic-interface-list** *interface_list_name*
5. **interface pw-ether** *id.subintfid* **l2transport**
6. **encapsulation dot1q** *value*
7. **l2vpn**
8. **xconnect group** *group-name*
9. **p2p** *xconnect-name*
10. **interface pw-ether** *id*
11. **neighbor ipv4** *ip-address* **pw-id** *value*
12. **bridge group** *bridge-group-name*

13. **bridge-domain** *bridge-domain-name*
14. **interface pw-ether** *id.subintfid*
15. **interface** *type interface-path-id*
16. **neighbor** *ip-address* **pw-id** *value*
17. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

**例**：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface pw-ether** *id*

**例**：

```
RP/0/RSP0/cpu 0: router(config)# interface PW-Ether1
```

指定されたインターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。

**ステップ 3** **ipv4 address** *ip-address subnet-mask*（または）**ipv6 address** *ipv6-prefix/prefix-length*

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# ipv4 address 40.1.1.2 255.255.255.0
```

メインインターフェイスの IPv4 または IPv6 アドレスを設定します。

**ステップ 4** **attach generic-interface-list** *interface_list_name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# attach generic-interface-list pw_he
```

インターフェイスに汎用インターフェイスリストを付加します。

**ステップ 5** **interface pw-ether** *id.subintfid* **l2transport**

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# interface PW-Ether1.1 l2transport
```

PWHE サブインターフェイスを設定し、サブインターフェイス設定モードを開始します。

**ステップ 6** **encapsulation dot1q** *value*

例：

```
RP/0/RSP0/cpu 0: router(config-subif)# encapsulation dot1q 1
```

インターフェイスの 802.1Q フレーム入力を適切なサービス インスタンスにマップするための一致基準を定義します。

**ステップ 7** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config-subif)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 8** **xconnect group** *group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group xg
```

自由形式の 32 文字ストリングを使用して、相互接続グループ名を設定します。

**ステップ 9** **p2p** *xconnect-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p1
```

P2P コンフィギュレーション サブモードを開始します。

**ステップ 10** **interface pw-ether** *id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface PW-Ether1
```

PWHE インターフェイスを設定します。

**ステップ 11** **neighbor ipv4** *ip-address* **pw-id** *value*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor ipv4 1.1.1.1 pw-id 1
```

相互接続の疑似回線を設定します。

IP アドレスは、対応する A-PE ノードのアドレスです。

pw-id は A-PE ノードの pw-id と一致する必要があります。

**ステップ 12** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p-pw)# bridge group bg
```

ブリッジドメインを含めることができるブリッジグループを作成し、ブリッジドメインにネットワークインターフェイスを割り当てます。

**ステップ 13** **bridge-domain** *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain bd1
```

ブリッジドメインを確立し、L2VPN ブリッジグループ ブリッジドメイン コンフィギュレーション モードを開始します。

**ステップ 14** **interface pw-ether** *id.subintfid*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# interface PW-Ether1.1
```

ブリッジドメインにサブインターフェイスを追加します。

**ステップ 15** **interface** *type interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)# interface GigabitEthernet0/1/1/11.1
```

インターフェイス設定モードを開始し、同じブリッジドメインに属する他のインターフェイスからパケットを転送および受信できるブリッジドメインにサブインターフェイスを追加します。

**ステップ 16** **neighbor** *ip-address* **pw-id** *value*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)# neighbor 3.3.3.3 pw-id 101
```

ブリッジドメインの疑似回線を設定します。

**ステップ 17** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# PWHE レイヤ 3 サブインターフェイスの設定

PWHE レイヤ 3 サブインターフェイスを設定するには、次の作業を実行します。

（注） レイヤ 3 サブインターフェイスには IPv4 または IPv6 アドレスが含まれている必要があるため、レイヤ 2 トランスポートモードでは設定できません。

## 手順の概要

1. **configure**
2. **interface pw-ether** *id*
3. **ipv4 address** *ip-address subnet-mask*（または）**ipv6 address** *ipv6-prefix/prefix-length*
4. **attach generic-interface-list** *interface_list_name*
5. **interface pw-ether** *id.subintfid*
6. **ipv4 address** *ip-address subnet-mask*（または）**ipv6 address** *ipv6-prefix/prefix-length*
7. **encapsulation dot1q** *value*
8. **l2vpn**
9. **xconnect group** *group-name*
10. **p2p** *group-name*
11. **interface pw-ether** *id*
12. **neighbor ipv4** *ip-address* **pw-id** *value*

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface pw-ether** *id*

例：

```
RP/0/RSP0/cpu 0: router(config)# interface PW-Ether1
```

指定されたインターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。

**ステップ 3** **ipv4 address** *ip-address subnet-mask*（または）**ipv6 address** *ipv6-prefix/prefix-length*

例：

```
RP/0/RSP0/cpu 0: router(config-if)# ipv4 address 40.1.1.2 255.255.255.0
```

メインインターフェイスの IPv4 または IPv6 アドレスを設定します。

**ステップ 4** **attach generic-interface-list** *interface_list_name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# attach generic-interface-list pw_he
```

インターフェイスに汎用インターフェイスリストを付加します。

**ステップ 5** **interface pw-ether** *id.subintfid*

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# interface PW-Ether1.1
```

PWHE サブインターフェイスを設定し、サブインターフェイス設定モードを開始します。

**ステップ 6** **ipv4 address** *ip-address subnet-mask*（または）**ipv6 address** *ipv6-prefix/prefix-length*

**例**：

```
RP/0/RSP0/cpu 0: router(config-if)# ipv4 address 40.1.1.2 255.255.255.0
```

サブインターフェイスの IPv4 または IPv6 アドレスを設定します。

**ステップ 7** **encapsulation dot1q** *value*

**例**：

```
RP/0/RSP0/cpu 0: router(config-subif)# encapsulation dot1q 1
```

インターフェイスの 802.1Q フレーム入力を適切なサービス インスタンスにマップするための一致基準を定義します。

**ステップ 8** **l2vpn**

**例**：

```
RP/0/RSP0/cpu 0: router(config-subif)# l2vpn
```

レイヤ 2 VPN コンフィギュレーション モードを開始します。

**ステップ 9** **xconnect group** *group-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# xconnect group xg
```

自由形式の 32 文字ストリングを使用して、相互接続グループ名を設定します。

**ステップ 10** **p2p** *group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc)# p2p 1
```

P2P コンフィギュレーション サブモードを開始します。

**ステップ 11** **interface pw-ether** *id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# interface PW-Ether1
```

PWHE インターフェイスを設定します。

**ステップ 12** **neighbor ipv4** *ip-address* **pw-id** *value*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-xc-p2p)# neighbor ipv4 1.1.1.1 pw-id 1
```

相互接続の疑似回線を設定します。

IP アドレスは、対応する A-PE ノードのアドレスです。

**pw-id** は A-PE ノードの **pw-id** と一致する必要があります。

# L2VPN over GRE の設定

L2VPN over GRE を設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **interface type** *interface-path-id*
3. **l2transport**
4. **exit**
5. **interface loopback instance**
6. **ipv4 address ip-address**
7. **exit**
8. **interface loopback** *instance*
9. **ipv4 address** *ip-address*
10. **router ospf** *process-name*
11. **area** *area-id*
12. **interface loopback** *instance*
13. **interface tunnel-ip** *number*
14. **exit**
15. **interface tunnel-ip** *number*

16. **ipv4 address** *ipv4-address subnet-mask*
17. **tunnel source** *type path-id*
18. **tunnel destination** *ip-address*
19. **end**
20. **l2vpn**
21. **bridge group** *bridge-group-name*
22. **bridge-domain** *bridge-domain-name*
23. **interface type** *interface-path-id*
24. **neighbor** { *A.B.C.D* } { **pw-id** *value* }
25. **mpls ldp**
26. **router-id** { *router-id* }
27. **interface tunnel-ip** *number*
28. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface type** *interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router# interface TenGigE0/1/0/12
```

インターフェイス コンフィギュレーション モードを開始し、インターフェイスを設定します。

**ステップ 3** **l2transport**

例：

```
RP/0/RSP0/cpu 0: router# l2transport
```

選択したインターフェイスでレイヤ 2 トランスポートをイネーブルにします。

**ステップ 4** **exit**

例：

```
RP/0/RSP0/cpu 0: router# exit
```

現在のコンフィギュレーション モードを終了します。

**ステップ 5** **interface loopback instance**

例：

```
RP/0/RSP0/cpu 0: router# interface Loopback0
```

インターフェイス コンフィギュレーション モードを開始して、新しいループバック インターフェイスの名前を指定します。

**ステップ 6** **ipv4 address ip-address**

例：

```
RP/0/RSP0/cpu 0: router# ipv4 address 100.100.100.100 255.255.255.255
```

仮想ループバック インターフェイスに IP アドレスおよびサブネット マスクを割り当てます。

**ステップ 7** **exit**

例：

```
RP/0/RSP0/cpu 0: router# exit
```

現在のコンフィギュレーション モードを終了します。

**ステップ 8** **interface loopback** *instance*

例：

```
RP/0/RSP0/cpu 0: router# interface Loopback1
```

インターフェイス コンフィギュレーション モードを開始して、新しいループバック インターフェイスの名前を指定します。

**ステップ 9** **ipv4 address** *ip-address*

例：

```
RP/0/RSP0/cpu 0: router# ipv4 address 10.0.1.1 255.255.255.255
```

仮想ループバック インターフェイスに IP アドレスおよびサブネット マスクを割り当てます。

**ステップ 10** **router ospf** *process-name*

例：

```
RP/0/RSP0/cpu 0: router# router ospf 1
```

指定したルーティング プロセスに OSPF ルーティングを有効にし、ルータ コンフィギュレーション モードでルータを配置します。

**ステップ 11** **area** *area-id*

例：

```
RP/0/RSP0/cpu 0: router# area 0
```

エリア コンフィギュレーション モードを開始し、OSPF プロセスのエリアを設定します。

**ステップ 12** **interface loopback** *instance*

例：

```
RP/0/RSP0/cpu 0: router# interface Loopback0
```

インターフェイス コンフィギュレーション モードを開始して、新しいループバック インターフェイスの名前を指定します。

**ステップ 13** **interface tunnel-ip** *number*

例：

```
RP/0/RSP0/cpu 0: router# interface tunnel-ip1
```

トンネル インターフェイス コンフィギュレーション モードを開始します。

**ステップ 14** **exit**

例：

```
RP/0/RSP0/cpu 0: router# exit
```

現在のコンフィギュレーション モードを終了します。

**ステップ 15** **interface tunnel-ip** *number*

例：

```
RP/0/RSP0/cpu 0: router(config)# interface tunnel-ip1
```

トンネル インターフェイス コンフィギュレーション モードを開始します。

- 番号はトンネル インターフェイスに関連付けられた番号です。

**ステップ 16** **ipv4 address** *ipv4-address subnet-mask*

例：

```
RP/0/RSP0/cpu 0: router(config-if)# ipv4 address 12.0.0.1 255.255.255.0
```

インターフェイスの IPv4 アドレスおよびサブネット マスクを指定します。

- ipv4-address は、インターフェイスの IP アドレスを指定します。
- subnet-mask は、インターフェイスのサブネット マスクを指定します。

**ステップ 17** **tunnel source** *type path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-if)# tunnel source Loopback1
```

トンネル インターフェイスの送信元を指定します。

**ステップ 18** **tunnel destination** *ip-address*

例：

```
RP/0/RSP0/cpu 0: router(config-if)# tunnel destination 100.100.100.20
```

トンネルの宛先を指定します。

**ステップ 19** **end**

例：

```
RP/0/RSP0/cpu 0: router(config-if)# end
```

設定変更を保存します。

- **end** コマンドを実行すると、変更をコミットするように要求されます。

  ```
  Uncommitted changes found, commit them before exiting(yes/no/cancel)?[cancel]:
  ```

  - **yes** と入力すると、実行設定ファイルに変更が保存され、設定セッションが終了して、ルータが **EXEC** モードに戻ります。
  - **no** と入力すると、設定セッションが終了して、ルータが **EXEC** モードに戻ります。設定の変更はコミットされません。
  - **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。

**ステップ 20** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 21** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router# bridge group access-pw
```

ブリッジ ドメインを含めることができるブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 22** **bridge-domain** *bridge-domain-name*

**例**：

```
RP/0/RSP0/cpu 0: router# bridge-domain test
```

ブリッジ ドメインを確立し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 23** **interface type** *interface-path-id*

**例**：

```
RP/0/RSP0/cpu 0: router# interface TenGigE0/1/0/12
```

インターフェイス コンフィギュレーション モードを開始し、同じブリッジ ドメインに属する他のインターフェイスからパケットを転送および受信できるブリッジ ドメインにインターフェイスを追加します。

**ステップ 24** **neighbor** { *A.B.C.D* } { **pw-id** *value* }

**例**：

```
RP/0/RSP0/cpu 0: router# neighbor 125.125.125.125 pw-id 100
```

アクセス疑似回線ポートをブリッジ ドメインに追加するか、または疑似回線を仮想転送インターフェイス（VFI）に追加します。

- クロスコネクトピアの IP アドレスを指定するには、A.B.C.D 引数を使用します。

  （注） A.B.C.D は再帰的または非再帰的プレフィックスです。

- 疑似回線 ID および ID 値を設定するには、pw-id キーワードを使用します。指定できる範囲は 1 ～ 4294967295 です。

**ステップ 25** **mpls ldp**

**例**：

```
RP/0/RSP0/cpu 0: router# mpls ldp
```

MPLS LDP コンフィギュレーション モードをイネーブルにします。

**ステップ 26** **router-id** { *router-id* }

**例**：

```
RP/0/RSP0/cpu 0: router# router-id 100.100.100.100
```

OSPF プロセスのルータ ID を設定します。

（注） 固定 IP アドレスをルータ ID として使用することを推奨します。

**ステップ 27** **interface tunnel-ip** *number*

例：

```
RP/0/RSP0/cpu 0: router# interface tunnel-ip1
```

トンネル インターフェイス コンフィギュレーション モードを開始します。

（注） number 引数は、トンネル インターフェイスに関連付けられた番号を示します。

**ステップ 28** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## 疑似回線の優先パスとしての GRE トンネルの設定

疑似回線の優先パスとして GRE トンネルを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **pw-class** { *name* }
4. **encapsulation mpls**
5. **preferred-path** { **interface** } { **tunnel-ip** *value* | **tunnel-te** *value* | **tunnel-tp** *value* } [ **fallback disable** ]
6. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **pw-class** { *name* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# pw-class gre
```

疑似回線クラス名を設定します。

**ステップ 4** **encapsulation mpls**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc)# encapsulation mpls
```

MPLS に疑似回線カプセル化を設定します。

**ステップ 5** **preferred-path** { **interface** } { **tunnel-ip** *value* | **tunnel-te** *value* | **tunnel-tp** *value* } [ **fallback disable** ]

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc-encap-
mpls)# preferred-path interface tunnel-ip 1 fallback disable
```

優先パス トンネルを設定します。フォールバックのディセーブル化の設定が使用されており、優先パスとして設定されている TE/TP トンネルがダウン状態になると、対応する疑似回線もダウン状態になることがあります。

（注） フォールバックがサポートされていることを確認します。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# VPLS LSM の設定：例

このセクションでは、VPLS LSM ソリューションを設定する方法の例を示します。

# VFI で RSVP-TE を使用する P2MP PW を有効化する

VFI で RSVP-TE を使用する P2MP 疑似回線を有効にするには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **vfi** { *vfi-name* }
6. **multicast p2mp**
7. **signaling protocol bgp**
8. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

### ステップ 1 configure

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

### ステップ 2 l2vpn

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

### ステップ 3 bridge group *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group bg1
```

レイヤ 2 VPN VPLS ブリッジグループ設定モードを開始します。

### ステップ 4 bridge-domain *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain bd1
```

レイヤ 2 VPLS VPN ブリッジ グループ ブリッジ ドメイン設定モードを開始します。

**ステップ 5** **vfi** { *vfi-name* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# vfi v1
```

仮想転送インターフェイス（VFI）パラメータを設定し、L2VPN ブリッジ グループ ブリッジ ドメイン VFI コンフィギュレーション モードを開始します。

- 指定した仮想転送インターフェイス名を設定するには、*vfi-name* 引数を使用します。

**ステップ 6** **multicast p2mp**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)# multicast p2mp
```

ポイントツーマルチポイント疑似回線を設定し、この VFI で疑似回線を有効にします。

**ステップ 7** **signaling protocol bgp**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-p2mp)# signaling protocol bgp
```

シグナリングプロトコルとして BGP を有効にします。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## VFI で P2MP PW の BGP 自動検出シグナリングを有効化する

VFI で P2MP 疑似回線の BGP 自動検出シグナリングを有効にするには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **vfi** { *vfi-name* }

6. **multicast p2mp**
7. **signaling protocol bgp**
8. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
RP/0/RSP0/cpu 0: router(config-l2vpn)#
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group bg1
```

レイヤ 2 VPN VPLS ブリッジグループ設定モードを開始します。

**ステップ 4** **bridge-domain** *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain bd1
```

レイヤ 2 VPLS VPN ブリッジ グループ ブリッジ ドメイン設定モードを開始します。

**ステップ 5** **vfi** { *vfi-name* }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# vfi v1
```

仮想転送インターフェイス（VFI）パラメータを設定し、L2VPN ブリッジ グループ ブリッジ ドメイン VFI コンフィギュレーション モードを開始します。

- 指定した仮想転送インターフェイス名を設定するには、vfi-name 引数を使用します。

**ステップ 6** **multicast p2mp**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)# multicast p2mp
```

ポイントツーマルチポイント疑似回線を設定し、この VFI で疑似回線を有効にします。

**ステップ 7** **signaling protocol bgp**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-p2mp)# signaling protocol bgp
```

シグナリングプロトコルとして BGP を有効にします。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## VPN ID の設定

VPN ID を設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **vfi** { *vfi-name* }
6. **vpn-id** *vpn-id*
7. **autodiscovery bgp**
8. **rd** { *as-number:nn* | *ip-address:nn* | **auto** }
9. **route-target export** { *as-number:nn* | *ip-address:nn* }
10. **signaling-protocol bgp**
11. **ve-id** { *number* }
12. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group metroA
```

レイヤ 2 VPN VPLS ブリッジグループ設定モードを開始します。

**ステップ 4** **bridge-domain** *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain east
```

レイヤ 2 VPLS VPN ブリッジ グループ ブリッジ ドメイン設定モードを開始します。

**ステップ 5** **vfi** { *vfi-name* }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# vfi vfi-east
```

仮想転送インスタンス（VFI）コンフィギュレーション モードを開始します。

**ステップ 6** **vpn-id** *vpn-id*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)# vpn-id 100
```

VPLS サービスの ID を指定します。VPN ID は、PE ルータ内でグローバルに一意である必要があります。つまり、同じ PE ルータ上の複数の VFI に同じ VPN ID を存在させることはできません。また、VFI に指定できる VPN ID は 1 つだけです。

有効な範囲は 1 ～ 65535 です。

**ステップ 7** **autodiscovery bgp**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi)# autodiscovery bgp
```

すべての BGP オートディスカバリ パラメータが設定される BGP オートディスカバリ コンフィギュレーション モードを開始します。

このコマンドは、少なくとも VPN ID とシグナリング プロトコルが設定されるまで、BGP にプロビジョニングされません。

**ステップ 8** **rd** { *as-number:nn* | *ip-address:nn* | **auto** }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad)# rd auto
```

VFI でルート識別子（RD）を指定します。

RD は、VFI を識別するために BGP NLRI で使用されます。VFI ごとに RD を 1 つだけ設定できます。**rd auto** を除き、RD は同じ PE の複数の VFI で設定できません。

**rd auto** が設定されている場合、RD 値は、{BGP ルータ ID}:{自動生成の一意の 16 ビットインデックス} の形式になります。

**ステップ 9** **route-target export** { *as-number:nn* | *ip-address:nn* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad)# route-target export 100:10
```

VFI のエクスポート ルート ターゲットを指定します。

エクスポート ルート ターゲットは、他の PE にアドバタイズされる NLRI 内に含められる RT です。

**ステップ 10** **signaling-protocol bgp**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad)# signaling-protocol bgp
```

BGP シグナリングをイネーブルにして、BGP シグナリング パラメータが設定される BGP シグナリング コンフィギュレーション サブモードを開始します。

このコマンドは、VE ID と VE ID の範囲が設定されるまで BGP にプロビジョニングされません。

**ステップ 11** **ve-id** { *number* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-vfi-ad-sig)# ve-id 10
```

VPLS を設定するために VFI のローカル PE ID を指定します。

VE ID は、VPLS サービス内の VFI を識別します。これは、同じ VPLS サービスの VFI が同じ VE ID を共有できないことを意味します。VE ID のスコープは、ブリッジ ドメイン内だけに存在します。したがって、PE 内の異なるブリッジ ドメインの VFI は、同じ VE ID を使用できます。

**ステップ 12** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## IGMP スヌーピングの設定

IGMP スヌーピングを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **igmp snooping profile** *profile_name*
3. **system-ip-address** *ip-address*
4. **internal-querier**
5. **exit**
6. **l2vpn**
7. **bridge group** *bridge-group-name*
8. **bridge-domain** *bridge-domain-name*
9. **igmp snooping disable**
10. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **igmp snooping profile** *profile_name*

例：

```
RP/0/RSP0/cpu 0: router(config)# igmp snooping profile default-bd-profile
```

IGMP スヌーピング プロファイル コンフィギュレーション モードを開始し、名前付きプロファイルを作成します。

**ステップ 3** **system-ip-address** *ip-address*

**例**：

```
RP/0/RSP0/cpu 0: router(config-igmp-snooping-profile)# system-ip-address 1.1.1.1
```

生成された IGMP メッセージの送信元アドレスを設定します。

**ステップ 4** **internal-querier**

**例**：

```
RP/0/RSP0/cpu 0: router(config-igmp-snooping-profile)#
```

IGMP 内部クエリ元を設定します。

（注） リーフ PE の場合、ブリッジドメインで IGMPSN を有効にするには、IGMPSN プロファイル内で内部クエリ元を設定していることを確認します。

**ステップ 5** **exit**

**例**：

```
RP/0/RSP0/cpu 0: router(config-igmp-snooping-
profile)# exit
```

グローバル コンフィギュレーション モードに戻ります。

**ステップ 6** **l2vpn**

**例**：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 7** **bridge group** *bridge-group-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group bg1
```

名前付きブリッジ グループのレイヤ 2 VPN VPLS ブリッジ グループ コンフィギュレーション モードを開始します。

**ステップ 8** **bridge-domain** *bridge-domain-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain bd1
```

名前付きブリッジドメインのレイヤ2 VPN VPLS ブリッジグループ ブリッジドメイン コンフィギュレーション モードを開始します。

**ステップ 9** **igmp snooping disable**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# igmp snooping disable
```

現在のブリッジドメインの IGMP スヌーピングを無効にします。

**ステップ 10** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# マルチポイント レイヤ 2 サービスの設定例

ここで示す設定例は、次のとおりです。

## プロバイダーエッジ間のマルチポイント レイヤ2サービスの設定：例

これらの設定は、参加しているマルチポイント レイヤ2サービスのプロバイダーエッジ（PE）ノードのフル メッシュでレイヤ 2 VFI を作成する例を示しています。

この設定は、PE 1 を設定する例を示しています。

```
configure
 l2vpn
  bridge group 1
   bridge-domain PE1-VPLS-A
    interface TenGigE0/0/0/0
    vfi 1
     neighbor 10.2.2.2 pw-id 1
     neighbor 10.3.3.3 pw-id 1
     !
   !
 interface loopback 0
  ipv4 address 10.1.1.1 255.255.255.25
```

この設定は、PE 2 を設定する例を示しています。

```
configure
 l2vpn
  bridge group 1
   bridge-domain PE2-VPLS-A
    interface TenGigE0/0/0/1

    vfi 1
     neighbor 10.1.1.1 pw-id 1
     neighbor 10.3.3.3 pw-id 1
     !
   !
 interface loopback 0
  ipv4 address 10.2.2.2 255.255.255.25
```

この設定は、PE 3 を設定する例を示しています。

```
configure
 l2vpn
  bridge group 1
   bridge-domain PE3-VPLS-A
    interface TenGigE0/0/0/2
    vfi 1
     neighbor 10.1.1.1 pw-id 1
     neighbor 10.2.2.2 pw-id 1
     !
   !
 interface loopback 0
  ipv4 address 10.3.3.3 255.255.255.25
```

# プロバイダー エッジとカスタマー エッジ間のマルチポイント レイヤ 2 サービスの設定：例

この設定は、PE-to-CE ノードのマルチポイント レイヤ 2 サービスの設定方法を示しています。

```
configure
 interface TenGigE0/0/0/0
  l2transport---AC interface

  no ipv4 address
  no ipv4 directed-broadcast
  negotiation auto
  no cdp enable
```

# MAC アドレス取り消しフィールドの表示：例

この出力は、MAC アドレス取り消しフィールドの例を示しています。

```
RP/0/RSP0/CPU0:router# show l2vpn bridge-domain detail

Legend: pp = Partially Programmed.
Bridge group: 222, bridge-domain: 222, id: 0, state: up, ShgId: 0, MSTi: 0
  Coupled state: disabled
  MAC learning: enabled
  MAC withdraw: enabled
    MAC withdraw sent on: bridge port up
    MAC withdraw relaying (access to access): disabled
  Flooding:
    Broadcast & Multicast: enabled
```

```
  Unknown unicast: enabled
MAC aging time: 300 s, Type: inactivity
MAC limit: 4000, Action: none, Notification: syslog
MAC limit reached: no
MAC port down flush: enabled
MAC Secure: disabled, Logging: disabled
Split Horizon Group: none
Dynamic ARP Inspection: disabled, Logging: disabled
IP Source Guard: disabled, Logging: disabled
DHCPv4 snooping: disabled
IGMP Snooping: enabled
IGMP Snooping profile: none
MLD Snooping profile: none
Storm Control: disabled
Bridge MTU: 1500
MIB cvplsConfigIndex: 1
Filter MAC addresses:
P2MP PW: disabled
Create time: 01/03/2017 11:01:11 (00:21:33 ago)
No status change since creation
ACs: 1 (1 up), VFIs: 1, PWs: 1 (1 up), PBBs: 0 (0 up)
List of ACs:
  AC: TenGigE0/2/0/1.7, state is up
    Type VLAN; Num Ranges: 1
    Outer Tag: 21
    VLAN ranges: [22, 22]
    MTU 1508; XC ID 0x208000b; interworking none
    MAC learning: enabled
    Flooding:
      Broadcast & Multicast: enabled
      Unknown unicast: enabled
    MAC aging time: 300 s, Type: inactivity
    MAC limit: 4000, Action: none, Notification: syslog
    MAC limit reached: no
    MAC port down flush: enabled
    MAC Secure: disabled, Logging: disabled
    Split Horizon Group: none
    Dynamic ARP Inspection: disabled, Logging: disabled
    IP Source Guard: disabled, Logging: disabled
    DHCPv4 snooping: disabled
    IGMP Snooping: enabled
    IGMP Snooping profile: none
    MLD Snooping profile: none
    Storm Control: bridge-domain policer
    Static MAC addresses:
    Statistics:
     packets: received 714472608 (multicast 0, broadcast 0, unknown unicast 0, unicast
0), sent 97708776
      bytes: received 88594603392 (multicast 0, broadcast 0, unknown unicast 0, unicast
0), sent 12115888224
      MAC move: 0
    Storm control drop counters:
      packets: broadcast 0, multicast 0, unknown unicast 0
      bytes: broadcast 0, multicast 0, unknown unicast 0
    Dynamic ARP inspection drop counters:
      packets: 0, bytes: 0
    IP source guard drop counters:
      packets: 0, bytes: 0
List of VFIs:
  VFI 222 (up)
    PW: neighbor 1.1.1.1, PW ID 222, state is up ( established )
      PW class not set, XC ID 0xc000000a
      Encapsulation MPLS, protocol LDP
      Source address 21.21.21.21
```

```
  PW type Ethernet, control word disabled, interworking none
  Sequencing not set

  PW Status TLV in use
    MPLS         Local                          Remote
    ------------ ------------------------------ ------------------------
    Label        24017                          24010
    Group ID     0x0                            0x0
    Interface    222                            222
    MTU          1500                           1500
    Control word disabled                       disabled
    PW type      Ethernet                       Ethernet
    VCCV CV type 0x2                            0x2
                 (LSP ping verification)        (LSP ping verification)
    VCCV CC type 0x6                            0x6
                 (router alert label)           (router alert label)
                 (TTL expiry)                   (TTL expiry)
    ------------ ------------------------------ ------------------------
  Incoming Status (PW Status TLV):
    Status code: 0x0 (Up) in Notification message
  MIB cpwVcIndex: 3221225482
  Create time: 01/03/2017 11:01:11 (00:21:33 ago)
  Last time status changed: 01/03/2017 11:21:01 (00:01:43 ago)
  Last time PW went down: 01/03/2017 11:15:21 (00:07:23 ago)
  MAC withdraw messages: sent 0, received 0
  Forward-class: 0
  Static MAC addresses:
  Statistics:
    packets: received 95320440 (unicast 0), sent 425092569
    bytes: received 11819734560 (unicast 0), sent 52711478556
    MAC move: 0
  Storm control drop counters:
    packets: broadcast 0, multicast 0, unknown unicast 0
    bytes: broadcast 0, multicast 0, unknown unicast 0
DHCPv4 snooping: disabled
IGMP Snooping profile: none
MLD Snooping profile: none
VFI Statistics:
  drops: illegal VLAN 0, illegal length 0
```

# スプリット ホライズン グループ：例

次の例では、レイヤ2トランスポートのインターフェイスを設定し、ブリッジドメインに追加し、スプリット ホライズン グループに割り当てます。

```
RP/0/RSP0/CPU0:router(config)#l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)#bridge group examples
RP/0/RSP0/CPU0:router(config-l2vpn-bg)#bridge-domain all_three
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#interface GigabitEthernet 0/0/0/0.99
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)#exit
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#interface GigabitEthernet 0/0/0/0.101
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)#split-horizon group
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)#exit
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#neighbor 192.168.99.1 pw-id 1
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pw)#exit
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#neighbor 192.168.99.9 pw-id 1
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pw)#split-horizon group
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pw)#exit
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#vfi abc
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-vfi)#neighbor 192.168.99.17 pw-id 1
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-vfi-pw)#exit
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-vfi)#exit
```

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#show
Mon Oct 18 13:51:05.831 EDT
l2vpn
 bridge group examples
  bridge-domain all_three
   interface GigabitEthernet0/0/0/0.99
   !
   interface GigabitEthernet0/0/0/0.101
    split-horizon group
   !
   neighbor 192.168.99.1 pw-id 1
   !
   neighbor 192.168.99.9 pw-id 1
    split-horizon group
   !
   vfi abc
    neighbor 192.168.99.17 pw-id 1
    !
   !
  !
 !
!

RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#
```

この例に従って、ブリッジドメイン **all_three** のスプリット ホライズン グループの割り当ては、次のようになります。

| ブリッジ ポート/疑似回線 | スプリット ホライズン グループ |
|---|---|
| ブリッジ ポート：gig0/0/0/0.99 | 0 |
| ブリッジ ポート：gig0/0/0/0.101 | 2 |
| PW：192.168.99.1 pw-id 1 | 0 |
| PW: 192.168.99.9 pw-id 1 | 2 |
| PW: 192.168.99.17 pw-id 1 | 1 |

# 不明なユニキャスト フラッディングのブロック：例

不明なユニキャスト フラッディングは、次のレベルでブロックできます。

- ブリッジ ドメイン
- ブリッジ ポート（接続回線（AC））
- アクセス疑似回線（PW）

次に、ブリッジ ドメイン レベルで不明なユニキャスト フラッディングをブロックする例を示します。

```
configure
  l2vpn
    bridge-group group1
    bridge-domain domain1
```

```
    flooding unknown-unicast disable
  end
```

次に、ブリッジ ポート レベルで不明なユニキャスト フラッディングをブロックする例を示します。

```
configure
  l2vpn
    bridge-group group1
    bridge-domain domain1
    interface GigabitEthernet 0/1/0/1
    flooding unknown-unicast disable
  end
```

次に、アクセス疑似回線レベルで不明なユニキャストフラッディングをブロックする例を示します。

```
configure
  l2vpn
    bridge-group group1
    bridge-domain domain1
    neighbor 10.1.1.1 pw-id 1000
    flooding unknown-unicast disable
  end
```

# MAC フラッシュのディセーブル化：例

次のレベルで MAC フラッシュをディセーブルにできます。

- ブリッジ ドメイン
- ブリッジ ポート（接続回線（AC））
- アクセス疑似回線（PW）

次に、ブリッジドメインレベルで MAC フラッシュを無効にする例を示します。

```
configure
  l2vpn
    bridge-group group1
    bridge-domain domain1
    mac
    port-down flush disable
  end
```

次に、ブリッジポートレベルで MAC フラッシュを無効にする例を示します。

```
configure
  l2vpn
    bridge-group group1
    bridge-domain domain1
    interface TenGigE 0/0/0/0
    mac
    port-down flush disable
  end
```

# IOS XR トランク インターフェイスでのブリッジング：例

次に、を単純な L2 スイッチとして設定する例を示します。

**特記事項**：

4 本の接続回線（AC）があるブリッジ ドメインを作成します。各 AC は、IOS XR トランク インターフェイスです（つまり、サブインターフェイス/EFP ではありません）。

- 次の例では、実行コンフィギュレーションが空であり、すべてのコンポーネントが作成されていると想定します。
- この例では、インターフェイス間のスイッチングを実行するようにを設定するために必要なすべての手順を示しています。ただし、no shut、negotiation auto などのインターフェイスを準備するためのコマンドは除外されています。
- ブリッジ ドメインは、作成直後に no shut 状態になります。
- この例ではトランク（つまりメイン）インターフェイスだけが使用されます。
- トランク インターフェイスは、タグ付き（IEEE 802.1Q）またはタグなし（つまり VLAN ヘッダーなし）フレームを処理できます。
- ブリッジ ドメインは、MAC アドレスに基づいて学習、フラッディング、および転送を行います。この機能は、タグの設定に関係なくフレームで動作します。
- ブリッジ ドメイン エンティティはシステム全体にわたります。単一の LC にすべてのブリッジ ドメイン AC を配置する必要はありません。これは、ブリッジ ドメインの設定に適用されます。
- ルータが予期したとおりに設定されていること、およびコマンドによって新しい設定ステータスが表示されることを確認するには、show bundle および show l2vpn bridge-domain コマンドを使用します。
- 次の例の AC では、管理ダウン状態になっているインターフェイスを使用します。

設定例

```
RP/0/RSP0/CPU0:router#config
RP/0/RSP0/CPU0:router(config)#interface Bundle-ether10
RP/0/RSP0/CPU0:router(config-if)#l2transport
RP/0/RSP0/CPU0:router(config-if-l2)#interface GigabitEthernet0/2/0/5
RP/0/RSP0/CPU0:router(config-if)#bundle id 10 mode active
RP/0/RSP0/CPU0:router(config-if)#interface GigabitEthernet0/2/0/6
RP/0/RSP0/CPU0:router(config-if)#bundle id 10 mode active
RP/0/RSP0/CPU0:router(config-if)#interface GigabitEthernet0/2/0/0
RP/0/RSP0/CPU0:router(config-if)#l2transport
RP/0/RSP0/CPU0:router(config-if-l2)#interface GigabitEthernet0/2/0/1
RP/0/RSP0/CPU0:router(config-if)#l2transport
RP/0/RSP0/CPU0:router(config-if-l2)#interface TenGigE0/1/0/2
RP/0/RSP0/CPU0:router(config-if)#l2transport
RP/0/RSP0/CPU0:router(config-if-l2)#l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)#bridge group examples
RP/0/RSP0/CPU0:router(config-l2vpn-bg)#bridge-domain test-switch
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#interface Bundle-ether10
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)#exit
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#interface GigabitEthernet0/2/0/0
```

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)#exit
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#interface GigabitEthernet0/2/0/1
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)#exit
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#interface TenGigE0/1/0/2
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)#commit
RP/0/RSP0/CPU0:Jul 26 10:48:21.320 EDT: config[65751]: %MGBL-CONFIG-6-DB_COMMIT :
Configuration committed by user 'lab'. Use 'show configuration commit changes 1000000973'
 to view the changes.
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)#end
RP/0/RSP0/CPU0:Jul 26 10:48:21.342 EDT: config[65751]: %MGBL-SYS-5-CONFIG_I : Configured
 from console by lab
RP/0/RSP0/CPU0:router#show bundle Bundle-ether10

Bundle-Ether10
  Status:                                  Down
  Local links <active/standby/configured>:   0 / 0 / 2
  Local bandwidth <effective/available>:     0 (0) kbps
  MAC address (source):                    0024.f71e.22eb (Chassis pool)
  Minimum active links / bandwidth:        1 / 1 kbps
  Maximum active links:                    64
  Wait while timer:                        2000 ms
  LACP:                                    Operational
    Flap suppression timer:                Off
  mLACP:                                   Not configured
  IPv4 BFD:                                Not configured

  Port                  Device           State        Port ID         B/W, kbps
  --------------------  ---------------  -----------  --------------  ----------
  Gi0/2/0/5             Local            Configured   0x8000, 0x0001     1000000
      Link is down
  Gi0/2/0/6             Local            Configured   0x8000, 0x0002     1000000
      Link is down

RP/0/RSP0/CPU0:router#
RP/0/RSP0/CPU0:router#show l2vpn bridge-domain group examples
Bridge group: examples, bridge-domain: test-switch, id: 2000, state: up, ShgId: 0, MSTi:
 0
  Aging: 300 s, MAC limit: 4000, Action: none, Notification: syslog
  Filter MAC addresses: 0
  ACs: 4 (1 up), VFIs: 0, PWs: 0 (0 up), PBBs: 0 (0 up)
  List of ACs:
    BE10, state: down, Static MAC addresses: 0
    Gi0/2/0/0, state: up, Static MAC addresses: 0
    Gi0/2/0/1, state: down, Static MAC addresses: 0
    Te0/5/0/1, state: down, Static MAC addresses: 0
  List of VFIs:
RP/0/RSP0/CPU0:router#
```

次の表に、設定手順（アクション）およびこの例の対応する目的を示します。

## 手順の概要

1. **configure**
2. **interface Bundle-ether10**
3. **l2transport**
4. **interface GigabitEthernet0/2/0/5**
5. **bundle id 10 mode active**
6. **interface GigabitEthernet0/2/0/6**
7. **bundle id 10 mode active**

8. **interface GigabitEthernet0/2/0/0**
9. **l2transport**
10. **interface GigabitEthernet0/2/0/1**
11. **l2transport**
12. **interface TenGigE0/1/0/2**
13. **l2transport**
14. **l2vpn**
15. **bridge group examples**
16. **bridge-domain test-switch**
17. **interface Bundle-ether10**
18. **exit**
19. **interface GigabitEthernet0/2/0/0**
20. **exit**
21. **interface GigabitEthernet0/2/0/1**
22. **exit**
23. **interface TenGigE0/1/0/2**
24. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface Bundle-ether10**

新しいバンドル トランク インターフェイスを作成します。

**ステップ 3** **l2transport**

Bundle-ether10 を L3 インターフェイスから L2 インターフェイスに変更します。

**ステップ 4** **interface GigabitEthernet0/2/0/5**

インターフェイス設定モードを開始します。GigabitEthernet0/2/0/5 で機能するようコンフィギュレーション モードを変更します。

**ステップ 5** **bundle id 10 mode active**

GigabitEthernet0/2/0/5 を Bundle-ether10 のメンバーとして設定します。**mode active** キーワードは、LACP プロトコルを指定します。

**ステップ 6** **interface GigabitEthernet0/2/0/6**

インターフェイス設定モードを開始します。GigabitEthernet0/2/0/6 で機能するようコンフィギュレーション モードを変更します。

**ステップ 7** **bundle id 10 mode active**

GigabitEthernet0/2/0/6 を Bundle-ether10 のメンバーとして設定します。**mode active** キーワードは、LACP プロトコルを指定します。

**ステップ 8** **interface GigabitEthernet0/2/0/0**

インターフェイス設定モードを開始します。GigabitEthernet0/2/0/0 で機能するようコンフィギュレーション モードを変更します。

**ステップ 9** **l2transport**

GigabitEthernet0/2/0/0 を L3 インターフェイスから L2 インターフェイスに変更します。

**ステップ 10** **interface GigabitEthernet0/2/0/1**

インターフェイス設定モードを開始します。GigabitEthernet0/2/0/1 で機能するようコンフィギュレーション モードを変更します。

**ステップ 11** **l2transport**

GigabitEthernet0/2/0/1 を L3 インターフェイスから L2 インターフェイスに変更します。

**ステップ 12** **interface TenGigE0/1/0/2**

インターフェイス設定モードを開始します。TenGigE0/1/0/2 で機能するようコンフィギュレーション モードを変更します。

**ステップ 13** **l2transport**

TenGigE0/1/0/2 を L3 インターフェイスから L2 インターフェイスに変更します。

**ステップ 14** **l2vpn**

L2VPN コンフィギュレーション モードを開始します。

**ステップ 15** **bridge group examples**

ブリッジ グループ **examples** を作成します。

**ステップ 16** **bridge-domain test-switch**

ブリッジ ドメイン **test-switch** を作成します。これは、ブリッジ グループ **examples** のメンバーです。

**ステップ 17** **interface Bundle-ether10**

Bundle-ether10 をブリッジ ドメイン test-switch の AC として設定します。

**ステップ 18** **exit**

ブリッジ ドメイン AC コンフィギュレーション サブモードを終了し、次の AC を設定できるようにします。

**ステップ 19** **interface GigabitEthernet0/2/0/0**

GigabitEthernet0/2/0/0 をブリッジ ドメイン **test-switch** の AC として設定します。

**ステップ 20** **exit**

ブリッジ ドメイン AC コンフィギュレーション サブモードを終了し、次の AC を設定できるようにします。

**ステップ 21** **interface GigabitEthernet0/2/0/1**

GigabitEthernet0/2/0/1 をブリッジ ドメイン **test-switch** の AC として設定します。

**ステップ 22** **exit**

ブリッジ ドメイン AC コンフィギュレーション サブモードを終了し、次の AC を設定できるようにします。

**ステップ 23** **interface TenGigE0/1/0/2**

インターフェイス TenGigE0/1/0/2 をブリッジ ドメイン **test-switch** の AC として設定します。

**ステップ 24** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# イーサネット フロー ポイントでのブリッジング：例

次に、イーサネットフローポイント（EFP）を通過するトラフィックでレイヤ2スイッチングを実行するように を設定する例を示します。EFP トラフィックには通常、1 つ以上の VLAN ヘッダーがあります。IOS XR トランクと IOS-XR EFP の両方をブリッジ ドメインで接続回線として結合できますが、この例では EFP だけを使用します。

**特記事項**：

- EFPは、レイヤ2サブインターフェイスです。これは常に、トランク インターフェイスの下で作成されます。トランク インターフェイスは、EFPを作成する前に存在している必要があります。
- 空の設定では、バンドル インターフェイス トランクは存在しませんが、物理トランク インターフェイスは自動的に設定されます。したがって、バンドル トランクだけが作成されます。
- この例では、サブインターフェイス番号および VLAN ID は同じですが、これは便利ではなく、必要性はありません。同じ値である必要はありません。
- ブリッジ ドメイン test-efp には、3 本の接続回線（AC）があります。AC はすべて EFP です。

- VLAN ID が 999 のフレームだけが EFP に入ります。これによって、このブリッジ ドメインのすべてのトラフィックで同じ VLAN カプセル化を確保できます。
- 次の例の AC では、管理ダウン状態（「未解決」状態）になっているインターフェイスを使用します。AC として存在しないインターフェイスを使用するブリッジ ドメインは正常であり、このような設定のコミットは失敗しません。この場合、ブリッジ ドメインのステータスは、欠落しているインターフェイスを設定するまで **unresolved** と表示されます。

設定例

```
RP/0/RSP1/CPU0:router#configure
RP/0/RSP1/CPU0:router(config)#interface Bundle-ether10
RP/0/RSP1/CPU0:router(config-if)#interface Bundle-ether10.999 l2transport
RP/0/RSP1/CPU0:router(config-subif)#encapsulation dot1q 999
RP/0/RSP1/CPU0:router(config-subif)#interface GigabitEthernet0/6/0/5
RP/0/RSP1/CPU0:router(config-if)#bundle id 10 mode active
RP/0/RSP1/CPU0:router(config-if)#interface GigabitEthernet0/6/0/6
RP/0/RSP1/CPU0:router(config-if)#bundle id 10 mode active
RP/0/RSP1/CPU0:router(config-if)#interface GigabitEthernet0/6/0/7.999 l2transport
RP/0/RSP1/CPU0:router(config-subif)#encapsulation dot1q 999
RP/0/RSP1/CPU0:router(config-subif)#interface TenGigE0/1/0/2.999 l2transport
RP/0/RSP1/CPU0:router(config-subif)#encapsulation dot1q 999
RP/0/RSP1/CPU0:router(config-subif)#l2vpn
RP/0/RSP1/CPU0:router(config-l2vpn)#bridge group examples
RP/0/RSP1/CPU0:router(config-l2vpn-bg)#bridge-domain test-efp
RP/0/RSP1/CPU0:router(config-l2vpn-bg-bd)#interface Bundle-ether10.999
RP/0/RSP1/CPU0:router(config-l2vpn-bg-bd-ac)#exit
RP/0/RSP1/CPU0:router(config-l2vpn-bg-bd)#interface GigabitEthernet0/6/0/7.999
RP/0/RSP1/CPU0:router(config-l2vpn-bg-bd-ac)#exit
RP/0/RSP1/CPU0:router(config-l2vpn-bg-bd)#interface TenGigE0/1/0/2.999
RP/0/RSP1/CPU0:router(config-l2vpn-bg-bd-ac)#commit
RP/0/RSP1/CPU0:router(config-l2vpn-bg-bd-ac)#end
RP/0/RSP1/CPU0:router#
RP/0/RSP1/CPU0:router#show l2vpn bridge group examples
Fri Jul 23 21:56:34.473 UTC Bridge group: examples, bridge-domain: test-efp, id: 0,
state: up, ShgId: 0, MSTi: 0
Aging: 300 s, MAC limit: 4000, Action: none, Notification: syslog
  Filter MAC addresses: 0
  ACs: 3 (0 up), VFIs: 0, PWs: 0 (0 up), PBBs: 0 (0 up)
  List of ACs:
    BE10.999, state: down, Static MAC addresses: 0
    Gi0/6/0/7.999, state: unresolved, Static MAC addresses: 0
    Te0/1/0/2.999, state: down, Static MAC addresses: 0
  List of VFIs:
RP/0/RSP1/CPU0:router#
```

次の表に、設定手順（アクション）およびこの例の対応する目的を示します。

## 手順の概要

1. **configure**
2. **interface Bundle-ether10**
3. **interface Bundle-ether10.999 l2transport**
4. **encapsulation dot1q 999**
5. **interface GigabitEthernet0/6/0/5**
6. **bundle id 10 mode active**

7. **interface GigabitEthernet0/6/0/6**
8. **bundle id 10 mode active**
9. **interface GigabitEthernet0/6/0/7.999 l2transport**
10. **encapsulation dot1q 999**
11. **interface TenGigE0/1/0/2.999 l2transport**
12. **encapsulation dot1q 999**
13. **l2vpn**
14. **bridge group examples**
15. **bridge-domain test-efp**
16. **interface Bundle-ether10.999**
17. **exit**
18. **interface GigabitEthernet0/6/0/7.999**
19. **exit**
20. **interface TenGigE0/1/0/2.999**
21. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface Bundle-ether10**

新しいバンドル トランク インターフェイスを作成します。

**ステップ 3** **interface Bundle-ether10.999 l2transport**

新しいバンドル トランクに EFP を作成します。

**ステップ 4** **encapsulation dot1q 999**

この EFP に VLAN ID 999 を割り当てます。

**ステップ 5** **interface GigabitEthernet0/6/0/5**

インターフェイス設定モードを開始します。GigabitEthernet0/6/0/5 で機能するようコンフィギュレーション モードを変更します。

**ステップ 6** **bundle id 10 mode active**

GigabitEthernet0/6/0/5 を Bundle-ether10 のメンバーとして設定します。**mode active** キーワードは、LACP プロトコルを指定します。

**ステップ 7** **interface GigabitEthernet0/6/0/6**

インターフェイス設定モードを開始します。GigabitEthernet0/6/0/6 で機能するようコンフィギュレーション モードを変更します。

**ステップ 8** **bundle id 10 mode active**

GigabitEthernet0/6/0/6 を Bundle-ether10 のメンバーとして設定します。**mode active** キーワードは、LACP プロトコルを指定します。

**ステップ 9** **interface GigabitEthernet0/6/0/7.999 l2transport**

GigabitEthernet0/6/0/7 に EFP を作成します。

**ステップ 10** **encapsulation dot1q 999**

この EFP に VLAN ID 999 を割り当てます。

**ステップ 11** **interface TenGigE0/1/0/2.999 l2transport**

TenGigE0/1/0/2 に EFP を作成します。

**ステップ 12** **encapsulation dot1q 999**

この EFP に VLAN ID 999 を割り当てます。

**ステップ 13** **l2vpn**

L2VPN コンフィギュレーション モードを開始します。

**ステップ 14** **bridge group examples**

**examples** という名前のブリッジ グループを作成します。

**ステップ 15** **bridge-domain test-efp**

**test-efp** という名前のブリッジ ドメインを作成します。これは、ブリッジ グループ **examples** のメンバーです。

**ステップ 16** **interface Bundle-ether10.999**

Bundle-ether10.999 を **test-efp** という名前のブリッジ ドメインの AC として設定します。

**ステップ 17** **exit**

ブリッジ ドメイン AC コンフィギュレーション サブモードを終了し、次の AC を設定できるようにします。

**ステップ 18** **interface GigabitEthernet0/6/0/7.999**

GigabitEthernet0/6/0/7.999 を **test-efp** という名前のブリッジ ドメインの AC として設定します。

**ステップ 19** **exit**

ブリッジ ドメイン AC コンフィギュレーション サブモードを終了し、次の AC を設定できるようにします。

**ステップ 20** **interface TenGigE0/1/0/2.999**

インターフェイス TenGigE0/1/0/2.999 を **test-efp** という名前のブリッジ ドメインの AC として設定します。

**ステップ 21** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# フラッディング最適化モードの変更

ブリッジ ドメインでフラッディング最適化モードを変更するには、次の作業を行います。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **flood mode convergence-optimized**
6. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group csco
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

**ステップ 4** **bridge-domain** *bridge-domain-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、l2vpn ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

**ステップ 5** **flood mode convergence-optimized**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# flood mode convergence-optimized
```

デフォルトのフラッディング最適化モードを帯域幅最適化モードからコンバージェンス モードに変更します。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

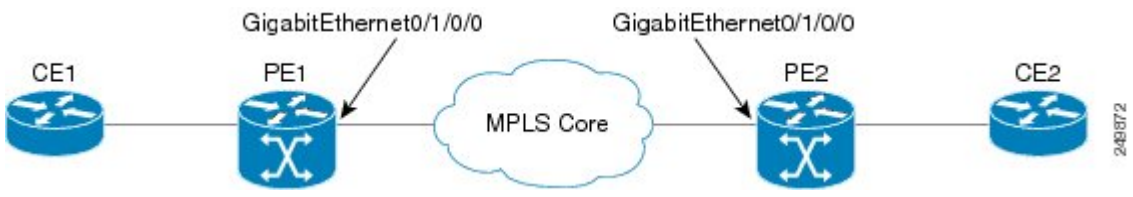
# BGP オートディスカバリおよびシグナリングでの VPLS の設定：例

ここでは、BGP オートディスカバリとシグナリング機能を設定するための例を示します。

## LDP および BGP の設定

次の図で、LDP および BGP の設定例について説明します。

**図 *34 : LDP* および *BGP* の設定**



**PE1** での設定：

```
interface Loopback0
 ipv4 address 1.1.1.100 255.255.255.255
!
interface Loopback1
 ipv4 address 1.1.1.10 255.255.255.255
!
mpls ldp
 router-id 1.1.1.1
 interface GigabitEthernt0/1/0/0
!
router bgp 120
 address-family l2vpn vpls-vpws
!
 neighbor 2.2.2.20
 remote-as 120
 update-source Loopback1
 address-family l2vpn vpls-vpws
 signaling bgp disable
```

**PE2** での設定：

```
interface Loopback0
 ipv4 address 2.2.2.200 255.255.255.255
!
interface Loopback1
 ipv4 address 2.2.2.20 255.255.255.255
!
mpls ldp
 router-id 2.2.2.2
 interface GigabitEthernt0/1/0/0
!
router bgp 120
 address-family l2vpn vpls-vpws
!
 neighbor 1.1.1.10
 remote-as 120
 update-source Loopback1
 address-family l2vpn vpls-vpws
```

## BGP シグナリングによる BGP オートディスカバリの最小の L2VPN 設定

次に、デフォルト値を持つパラメータが設定されていない BGP シグナリングを使用する BGP オートディスカバリに必要な最小の L2VPN 設定例を示します。

```
(config)# l2vpn
(config-l2vpn)# bridge group {bridge group name}
(config-l2vpn-bg)# bridge-domain {bridge domain name}
(config-l2vpn-bg-bd)# vfi {vfi name}
(config-l2vpn-bg-bd-vfi)# autodiscovery bgp
(config-l2vpn-bg-bd-vfi-ad)# vpn-id 10
(config-l2vpn-bg-bd-vfi-ad)# rd auto
(config-l2vpn-bg-bd-vfi-ad)# route-target 1.1.1.1:100
(config-l2vpn-bg-bd-vfi-ad-sig)# signaling-protocol bgp
(config-l2vpn-bg-bd-vfi-ad-sig)# ve-id 1
(config-l2vpn-bg-bd-vfi-ad-sig)# commit
```

# BGP オートディスカバリおよび BGP シグナリングでの VPLS

次の図 に、BGP オートディスカバリ（AD）および BGP シグナリングを使用して VPLS を設定する例を示します。

**図 *35 : BGP* オートディスカバリおよび *BGP* シグナリングを使用した *VPLS***


**PE1 での設定：**

```
l2vpn
  bridge group gr1
   bridge-domain bd1
    interface GigabitEthernet0/1/0/1.1
    vfi vf1
    ! AD independent VFI attributes
    vpn-id 100
    ! Auto-discovery attributes
    autodiscovery bgp
    rd auto
    route-target 2.2.2.2:100
    ! Signaling attributes
    signaling-protocol bgp
    ve-id 3
```

**PE2 での設定：**

```
l2vpn
  bridge group gr1
   bridge-domain bd1
    interface GigabitEthernet0/1/0/2.1
    vfi vf1
    ! AD independent VFI attributes
    vpn-id 100
    ! Auto-discovery attributes
    autodiscovery bgp
    rd auto
    route-target 2.2.2.2:100
    ! Signaling attributes
    signaling-protocol bgp
    ve-id 5
```

次に、BGP AD およびシグナリングを使用する VPLS の NLRI の例を示します。

ディスカバリ属性

**PE1** で送信される **NLRI**：

```
Length = 19
Router Distinguisher = 3.3.3.3:32770
VE ID = 3
VE Block Offset = 1
VE Block Size = 10
Label Base = 16015
```

**PE2** で送信される **NLRI**：

```
Length = 19
Router Distinguisher = 1.1.1.1:32775
VE ID = 5
VE Block Offset = 1
VE Block Size = 10
Label Base = 16120
```

## LDP シグナリングによる BGP オートディスカバリの最小設定

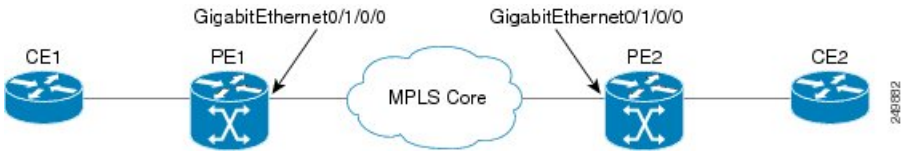
次に、デフォルト値を持つパラメータが設定されていない LDP シグナリングを使用する BGP オートディスカバリに必要な最小の L2VPN 設定例を示します。

```
(config)# l2vpn
(config-l2vpn)# bridge group {bridge group name}
(config-l2vpn-bg)# bridge-domain {bridge domain name}
(config-l2vpn-bg-bd)# vfi {vfi name}
(config-l2vpn-bg-bd-vfi)# autodiscovery bgp
(config-l2vpn-bg-bd-vfi-ad)# vpn-id 10
(config-l2vpn-bg-bd-vfi-ad)# rd auto
(config-l2vpn-bg-bd-vfi-ad)# route-target 1.1.1.1:100
(config-l2vpn-bg-bd-vfi-ad)# commit
```

## BGP オートディスカバリおよび LDP シグナリングでの VPLS

次の図に、BGP オートディスカバリ（AD）および LDP シグナリングを使用して VPLS を設定する例を示します。

**図 *36 : BGP* オートディスカバリおよび *LDP* シグナリングでの *VPLS***




**PE1** での設定：

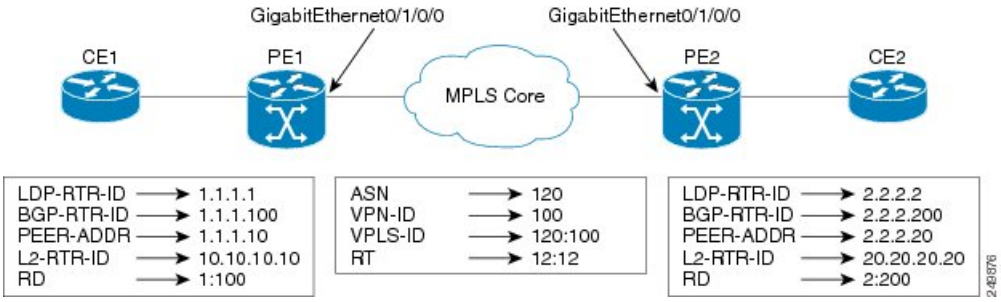
```
l2vpn
  router-id 10.10.10.10
  bridge group bg1
   bridge-domain bd1
    vfi vf1
```

```
    vpn-id 100
    autodiscovery bgp
    rd 1:100
    route-target 12:12
```

**PE2 での設定**：

```
l2vpn
  router-id 20.20.20.20
  bridge group bg1
   bridge-domain bd1
    vfi vf1
     vpn-id 100
     autodiscovery bgp
      rd 2:200
      route-target 12:12
      signaling-protocol ldp
       vpls-id 120:100
```

### ディスカバリおよびシグナリングの属性




**PE1 での設定**：

```
LDP Router ID - 1.1.1.1
BGP Router ID - 1.1.1.100
Peer Address - 1.1.1.10
L2VPN Router ID - 10.10.10.10
Route Distinguisher - 1:100
```

**PE1 と PE2 間の共通の設定**：

```
ASN - 120
VPN ID - 100
VPLS ID - 120:100
Route Target - 12:12
```

**PE2 での設定**：

```
LDP Router ID - 2.2.2.2
BGP Router ID - 2.2.2.200
Peer Address - 2.2.2.20
L2VPN Router ID - 20.20.20.20
Route Distinguisher - 2:200
```

ディスカバリ属性

**PE1** で送信される **NLRI**：

```
Source Address - 1.1.1.10
Destination Address - 2.2.2.20
Length - 14
Route Distinguisher - 1:100
L2VPN Router ID - 10.10.10.10
VPLS ID - 120:100
Route Target - 12:12
```

**PE2** で送信される **NLRI**：

```
Source Address - 2.2.2.20
Destination Address - 1.1.1.10
Length - 14
Route Distinguisher - 2:200
L2VPN Router ID - 20.20.20.20
VPLS ID - 120:100
Route Target - 12:12
```

### BGP オートディスカバリのための VC タイプ 4 の有効化

次に、BGP オートディスカバリを使用して VPLS で仮想接続タイプ 4 を設定する例を示します。

```
l2vpn
bridge group bg1
  bridge-domain bd1
   transport-mode vlan passthrough
   interface GigabitEthernet0/0/0/1.1
   !
   neighbor 2.2.2.2 pw-id 1
   !
   vfi vf1
    vpn-id 100
    autodiscovery bgp
     rd auto
     route-target 1:1
     signalining-protocol ldp
    !
   !
  !
!
```

## BGP オートディスカバリと VPLS ピアの手動プロビジョニングの両方を使用した VPLS

この例では、BGP オートディスカバリおよび BGP オートディスカバリプロセスに参加していない VPLS ピアの手動プロビジョニングを使用する VPLS 設定を示します。

```
!
l2vpn
 bridge group bg1
  bridge-domain bd1
   !
   vfi vfi1
```

```
    vpn-id 500
    autodiscovery bgp
     rd auto
     route-target 65533:12345678
     signaling-protocol ldp
      vpls-id 65533:12345678
    ! Manually provisioned peers
    neighbor 10.10.10.2 pw-id 102
!
```

# ダイナミック ARP インスペクションの設定：例

次に、ブリッジ ドメインで基本的なダイナミック ARP インスペクションを設定する例を示します。

```
config
l2vpn
 bridge group MyGroup
 bridge-domain MyDomain
 dynamic-arp-inspection logging
```

次に、ブリッジ ポートで基本的なダイナミック ARP インスペクションを設定する例を示します。

```
config
l2vpn
 bridge group MyGroup
 bridge-domain MyDomain
  interface gigabitEthernet 0/1/0/0.1
 dynamic-arp-inspection logging
```

次に、ブリッジ ドメインでオプションのダイナミック ARP インスペクションを設定する例を示します。

```
l2vpn
  bridge group SECURE
    bridge-domain SECURE-DAI
      dynamic-arp-inspection
        logging
        address-validation
          src-mac
          dst-mac
           ipv4
```

次に、ブリッジ ポートでオプションのダイナミック ARP インスペクションを設定する例を示します。

```
l2vpn
  bridge group SECURE
    bridge-domain SECURE-DAI
      interface GigabitEthernet0/0/0/1.10
        dynamic-arp-inspection
          logging
          address-validation
           src-mac
           dst-mac
           ipv4
```

次に、**show l2vpn bridge-domain** *bd-name* SECURE-DAI**detail** コマンドの出力例を示します。

```
#show l2vpn bridge-domain bd-name SECURE-DAI detail
Bridge group: SECURE, bridge-domain: SECURE-DAI, id: 2, state: up,
```

```
…
Dynamic ARP Inspection: enabled, Logging: enabled
Dynamic ARP Inspection Address Validation:
   IPv4 verification: enabled
   Source MAC verification: enabled
    Destination MAC verification: enabled
…
List of ACs:
 AC: GigabitEthernet0/0/0/1.10, state is up
…
    Dynamic ARP Inspection: enabled, Logging: enabled
    Dynamic ARP Inspection Address Validation:
        IPv4 verification: enabled
        Source MAC verification: enabled
        Destination MAC verification: enabled
        IP Source Guard: enabled, Logging: enabled

…
   Dynamic ARP inspection drop counters:
      packets: 1000, bytes: 64000
```

次に、**show l2vpn forwarding interface** *interface-name* **detail location** *location-name* コマンドの出力例を示します。

```
#show l2vpn forwarding interface g0/0/0/1.10 det location 0/0/CPU0
Local interface: GigabitEthernet0/0/0/1.10, Xconnect id: 0x40001, Status: up

…
    Dynamic ARP Inspection: enabled, Logging: enabled
    Dynamic ARP Inspection Address Validation:
        IPv4 verification: enabled
        Source MAC verification: enabled
        Destination MAC verification: enabled
        IP Source Guard: enabled, Logging: enabled
```

次に、ロギング表示の例を示します。

```
LC/0/0/CPU0:Jun 16 13:28:28.697 : l2fib[188]: %L2-L2FIB-5-SECURITY_DAI_VIOLATION_AC :
Dynamic ARP inspection in AC GigabitEthernet0_0_0_7.1000 detected violated packet -
source MAC: 0000.0000.0065, destination MAC: 0000.0040.0000, sender MAC: 0000.0000.0064,
 target MAC: 0000.0000.0000, sender IP: 5.6.6.6, target IP: 130.10.3.2

LC/0/5/CPU0:Jun 16 13:28:38.716 : l2fib[188]: %L2-L2FIB-5-SECURITY_DAI_VIOLATION_AC :
Dynamic ARP inspection in AC Bundle-Ether100.103 detected violated packet - source MAC:
 0000.0000.0067, destination MAC: 0000.2300.0000, sender MAC: 0000.7800.0034, target
MAC: 0000.0000.0000, sender IP: 130.2.5.1, target IP: 50.5.1.25
```

# IP ソース ガードの設定：例

次に、ブリッジ ドメインで基本的な IP ソース ガードを設定する例を示します。

```
config
l2vpn
 bridge group MyGroup
 bridge-domain MyDomain
 ip-source-guard logging
```

次に、ブリッジ ポートで基本的な IP ソース ガードを設定する例を示します。

```
config
l2vpn
```

```
 bridge group MyGroup
 bridge-domain MyDomain
  interface gigabitEthernet 0/1/0/0.1
 ip-source-guard  logging
```

次に、ブリッジ ドメインでオプションの IP ソース ガードを設定する例を示します。

```
l2vpn
  bridge group SECURE
    bridge-domain SECURE-IPSG
       ip-source-guard
         logging
```

次に、ブリッジ ポートでオプションの IP ソース ガードを設定する例を示します。

```
l2vpn
  bridge group SECURE
    bridge-domain SECURE-IPSG
      interface GigabitEthernet0/0/0/1.10
        ip-source-guard
          logging
```

次に、**show l2vpn bridge-domain** *bd-name ipsg-name* **detail** コマンドの出力例を示します。

```
# show l2vpn bridge-domain bd-name SECURE-IPSG detail
Bridge group: SECURE, bridge-domain: SECURE-IPSG, id: 2, state: up,
…
  IP Source Guard: enabled, Logging: enabled
…
List of ACs:
 AC: GigabitEthernet0/0/0/1.10, state is up
…

   IP Source Guard: enabled, Logging: enabled
…
   IP source guard drop counters:
      packets: 1000, bytes: 64000
```

次に、**show l2vpn forwarding interface***interface-name* **detail location***location-name* コマンドの出力例を示します。

```
# show l2vpn forwarding interface g0/0/0/1.10 detail location 0/0/CPU0
Local interface: GigabitEthernet0/0/0/1.10, Xconnect id: 0x40001, Status: up

…
    IP Source Guard: enabled, Logging: enabled
```

次に、ロギング表示の例を示します。

```
LC/0/0/CPU0:Jun 16 13:32:25.334 : l2fib[188]: %L2-L2FIB-5-SECURITY_IPSG_VIOLATION_AC :
IP source guard in AC  GigabitEthernet0_0_0_7.1001 detected violated packet - source
MAC: 0000.0000.0200, destination MAC: 0000.0003.0000, source IP: 130.0.0.1, destination
 IP: 125.34.2.5



LC/0/5/CPU0:Jun 16 13:33:25.530 : l2fib[188]: %L2-L2FIB-5-SECURITY_IPSG_VIOLATION_AC :
IP source guard in AC Bundle-Ether100.100  detected violated packet - source MAC:
0000.0000.0064, destination MAC: 0000.0040.0000, source IP: 14.5.1.3, destination IP:
45.1.1.10
```

# G.8032 イーサネット リング保護の設定：例

この設定例では、完全な G.8032 設定に含まれている要素について説明します。

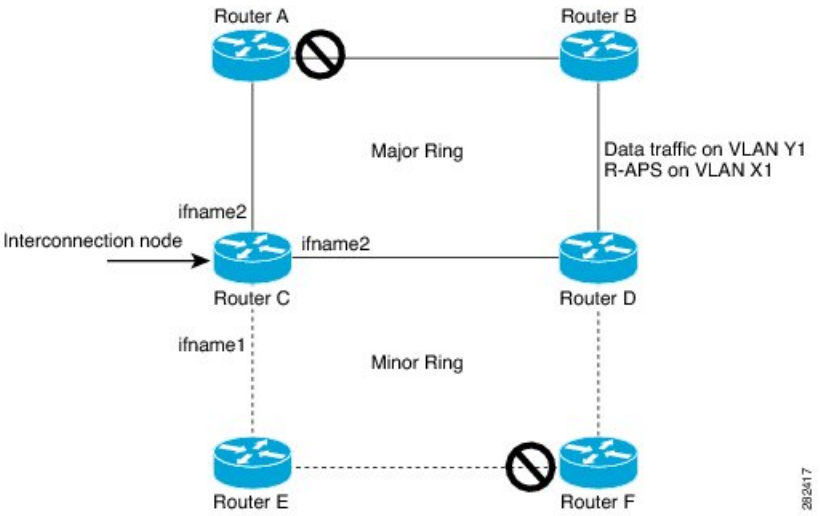
```
# Configure the ERP profile characteristics if ERP instance behaviors are non-default.
ethernet ring g8032 profile ERP-profile
  timer wtr 60
  timer guard 100
  timer hold-off 1
  non-revertive

# Configure CFM MEPs and configure to monitor the ring links.
ethernet cfm
  domain domain1
    service link1 down-meps
   continuity-check interval 100ms
      efd
   mep crosscheck
  mep-id 2
  domain domain2
    service link2 down-meps
   continuity-check interval 100ms
      efd protection-switching
   mep crosscheck
  mep id 2

Interface Gig 0/0/0/0
  ethernet cfm mep domain domain1 service link1 mep-id 1
Interface Gig 1/1/0/0
  ethernet cfm mep domain domain2 service link2 mep-id 1

# Configure the ERP instance under L2VPN
l2vpn
  ethernet ring g8032 RingA
    port0 interface g0/0/0/0
    port1 interface g0/1/0/0
    instance 1
      description BD2-ring
      profile ERP-profile
      rpl port0 owner
      vlan-ids 10-100
      aps channel
        level 3
        port0 interface g0/0/0/0.1
        port1 interface g1/1/0/0.1

# Set up the bridge domains
bridge group ABC
    bridge-domain BD2
      interface Gig 0/0/0/0.2
      interface Gig 0/1/0/0.2
      interface Gig 0/2/0/0.2

    bridge-domain BD2-APS
      interface Gig 0/0/0/0.1
      interface Gig 1/1/0/0.1

# EFPs configuration
interface Gig 0/0/0/0.1 l2transport
  encapsulation dot1q 5

interface Gig 1/1/0/0.1 l2transport
  encapsulation dot1q 5
```

```
interface g 0/0/0/0.2 l2transport
  encapsulation dot1q 10-100

interface g 0/1/0/0.2 l2transport
  encapsulation dot1q 10-100

interface g 0/2/0/0.2 l2transport
  encapsulation dot1q 10-100
```

## 相互接続ノードの設定：例

次に、相互接続ノードを設定する例を示します。次の図では、開いたリングのシナリオについて説明します。

**図 *37*：リング シナリオ：相互接続ノード**




ルータ C（開いたリング：ルータ C）で G.8032 を設定するために必要な最小設定：

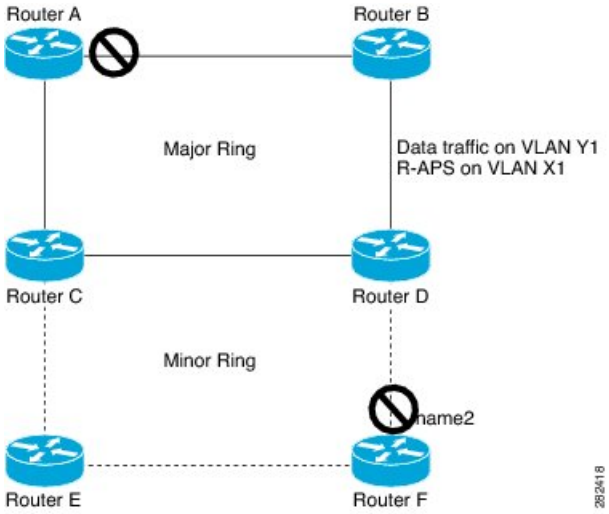
```
interface <ifname1.1> l2transport
 encapsulation dot1q X1
interface <ifname1.10> l2transport
 encapsulation dot1q Y1
interface <ifname2.10> l2transport
 encapsulation dot1q Y1
interface <ifname3.10> l2transport
 encapsulation dot1q Y1
l2vpn
ethernet ring g8032 <ring-name>
      port0 interface <main port ifname1>
      port1 interface none #? This router is connected to an interconnection node
      open-ring #? Mandatory when a router is part of an open-ring
      instance <1-2>
         inclusion-list vlan-ids  X1-Y1
         aps-channel
           Port0 interface <ifname1.1>
           Port1 none #? This router is connected to an interconnection node
 bridge group bg1
    bridge-domain bd-aps#? APS-channel has its own bridge domain
       <ifname1.1> #? There is only one APS-channel at the interconnection node
    bridge-domain bd-traffic #? Data traffic has its own bridge domain
       <ifname1.10>
```

```
        <ifname2.10>
        <ifname3.10>
```

# 開いたリングのノードの設定：例

次に、開いたリングのノード部分を設定する例を示します。次の図では、開いたリングのシナリオについて説明します。

**図 *38* : 開いたリング シナリオ**




開いたリングのノード（ルータ F で開いたリングのノード部分）で G.8032 を設定するのに必要な最小設定：

```
interface <ifname1.1> l2transport
 encapsulation dot1q X1
interface <ifname2.1> l2transport
 encapsulation dot1q X1
interface <ifname1.10> l2transport
 encapsulation dot1q Y1
interface <ifname2.10> l2transport
 encapsulation dot1q Y1
l2vpn
   ethernet ring g8032 <ring-name>
      port0 interface <main port ifname1>
      port1 interface <main port ifname2>
      open-ring #? Mandatory when a router is part of an open-ring
      instance <1-2>
         inclusion-list vlan-ids  X1-Y1
      rpl port1 owner #? This node is RPL owner and <main port ifname2> is blocked
         aps-channel
           port0 interface <ifname1.1>
           port1 interface <ifname2.1>
 bridge group bg1
    bridge-domain bd-aps#? APS-channel has its own bridge domain
       <ifname1.1>
       <ifname2.1>
    bridge-domain bd-traffic #? Data traffic has its own bridge domain
       <ifname1.10>
       <ifname2.10>
```

# Flow Aware Transport 疑似回線の設定：例

この設定例では、VPWS の FAT PW によるロード バランシングをイネーブルにする方法を示します。

```
l2vpn
pw-class class1
    encapsulation mpls
        load-balancing flow-label transmit
    !
 !
pw-class class2
    encapsulation mpls
        load-balancing flow-label both
    !

xconnect group group1
    p2p p1
        interface TenGigE 0/0/0/0.1
        neighbor 1.1.1.1 pw-id 1
            pw-class class1
        !
    !
!
```

この設定例では、VPLS の FAT PW によるロード バランシングをイネーブルにする方法を示します。

（注） VPLS の場合、ブリッジドメイン レベルでの設定は、すべての PW（アクセスおよび VFI PW）に適用されます。疑似回線クラスは、手動 PW の設定を上書きするために定義されます。

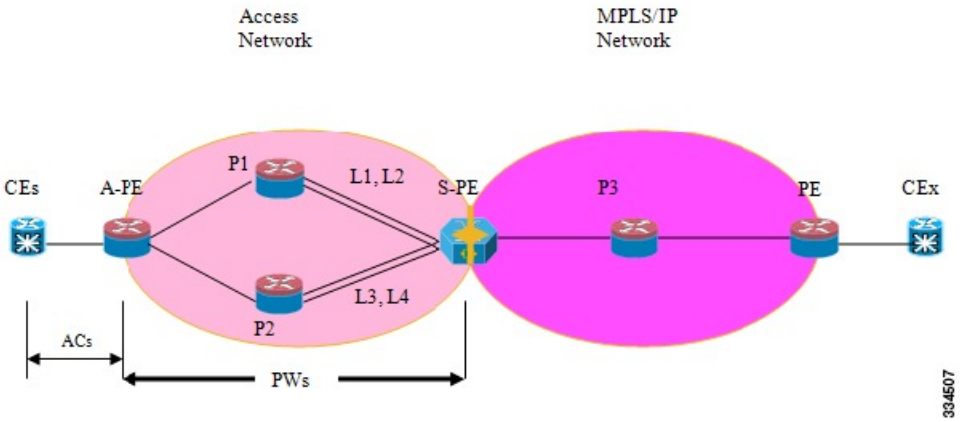
```
l2vpn
pw-class class1
 encapsulation mpls
 load-balancing flow-label both

bridge group group1
    bridge-domain domain1
        vfi vfi2-auto-bgp
                autodiscovery bgp
                   signaling-protocol bgp
                       load-balancing flow-label both static
                !
            !
        !
    !
    bridge-domain domain2
        vfi vfi2-auto-ldp
                autodiscovery bgp
                   signaling-protocol ldp
                       load-balancing flow-label both static
                !
            !
        !
    !
!
```

# 疑似回線ヘッドエンドの設定：例

次に、疑似回線ヘッドエンドを設定する例を示します。

次の図のトポロジについて考えます。

**図 *39 :* 疑似回線ヘッドエンドの例**




A-PE に接続された複数の CE があります（各 CE は 1 つのリンクによって接続されます）。アクセス ネットワークの A-PE と S-PE 間に 2 つの P ルータがあります。S-PE は、2 つのリンクで P1 に接続されています。これらは L1 および L2（P1 および S-PE の 2 つの異なるラインカード上）をリンクします。たとえば、それぞれ Gig0/1/0/0 および Gig0/2/0/0 になります。

S-PE は、P2 に 2 つのリンクで接続され、L3 および L4（P2 および S-PE の 2 つの異なるラインカード上）をリンクします。たとえば、それぞれ Gig0/1/0/1 および Gig0/2/0/1 になります。各 CE-APE リンクについて、相互接続（AC-PW）が A-PE 上に設定されます。A-PE は、ルーティングと PW シグナリングに router-id 100.100.100.100 を使用します。PW シグナリングには、S-PE 上の 2 つの router-id（111.111.111.111 および 112.112.112.112（rx pin-down 用）など）が使用されます。ルーティングには router-id 110.110.110.110 が使用されます。

### CE の設定

Ge0/3/0/0（CE1 と A-PE）および Ge0/3/0/1（CE2 と A-PE）を介して接続された 2 つの CE を考慮します。

CE1

```
interface Gig0/3/0/0
 ipv4 address 10.1.1.1/24
router static
 address-family ipv4 unicast
  110.110.110.110 Gig0/3/0/0
  A.B.C.D/N 110.110.110.110
```

CE2

```
interface Gig0/3/0/1
```

```
 ipv4 address 10.1.2.1/24
router static
 address-family ipv4 unicast
  110.110.110.110 Gig0/3/0/1
  A.B.C.D/N 110.110.110.110
```

### A-PE の設定

A-PE の場合、各 CE 接続に 1 つの相互接続があります。ここで上記の 2 つの CE 接続を設定します。接続は両方とも相互接続である L2 リンクです。各相互接続には S-PE 宛ての PW がありますが、ここでは PW をピンダウンする場所（[L1, L4] または [L2, L3]）に応じて別のネイバーアドレスを使用します。

```
interface Gig0/3/0/0
 l2transport
interface Gig0/3/0/1
 l2transport

l2vpn
 xconnect group pwhe
  p2p pwhe_spe_1
   interface Gig0/3/0/0
   neighbor 111.111.111.111 pw-id 1
  p2p pwhe_spe_2
   interface Gig0/3/0/1
   neighbor 112.112.112.112 pw-id 2
```

### P ルータの設定

S-PE の rx ピンダウン用の P ルータには、スタティック ルートが必要です。つまり、PW に、特定のリンクを介した特定のアドレスへの転送を強制します。

P1

```
router static
 address-family ipv4 unicast
  111.111.111.111 Gig0/1/0/0
  112.112.112.112 Gig0/2/0/0
```

P2

```
router static
 address-family ipv4 unicast
  111.111.111.111 Gig0/2/0/1
  112.112.112.112 Gig0/1/0/1
```

### S-PE の設定

S-PE の場合、2 つの PW-HE インターフェイス（各 PW に 1 つ）があり、tx ピンダウンにそれぞれ異なるインターフェイス リストを使用します（tx ピンダウンは rx ピンダウン用の P ルータでスタティックな設定が一致する必要があります）。各 PW-HE には A-PE に向かう PW があります（pw-id が A-PE のものと一致する必要があります）。

```
generic-interface-list il1
 interface gig0/1/0/0
 interface gig0/2/0/0
generic-interface-list il2
 interface gig0/1/0/1
 interface gig0/2/0/1

interface pw-ether1
 ipv4 address 10.1.1.2/24
 attach generic-interface-list il1
interface pw-ether2
 ipv4 address 10.1.2.2/24
 attach generic-interface-list il2

l2vpn
 xconnect group pwhe
  p2p pwhe1
   interface pw-ether1
   neighbor 100.100.100.100 pw-id 1
  p2p pwhe2
   interface pw-ether2
   neighbor 100.100.100.100 pw-id 2
```

# L2VPN over GRE の設定：例

IGP の下の PW コア インターフェイスを設定し、ループバックを到達可能にします。トンネル送信元を設定し、トンネルが現在のループバックになるように、およびピア PE ループバックの宛先になるようにします。ここでは、IGP（OSPF または ISIS）内、および **mpls ldp** の下に GRE トンネルを設定し、LDP ネイバーが PW の PE 間で確立されるようにします。これにより、トンネルで PW がアップするようになります。

PE1 の設定：

```
router ospf 1
router-id 1.1.1.1
 area 0
  interface Loopback0
  interface TenGigE0/0/0/1
router ospf 2
 router-id 200.200.200.200
 area 0
  interface Loopback1000
   interface tunnel-ip1
mpls ldp
 router-id 200.200.200.200
 interface tunnel-ip1
```

PE2 の設定：

```
router ospf 1
 router-id 3.3.3.3
 area 0
  interface Loopback0
interface TenGigE0/2/0/3
router ospf 2
 router-id 201.201.201.201
 area 0
  interface Loopback1000
   interface tunnel-ip1
!
```

```
mpls ldp
 router-id 201.201.201.201
 interface tunnel-ip1
```

### 疑似回線の優先パスとしての GRE トンネルの設定：例

次に、疑似回線の優先パスとして GRE トンネルを設定する例を示します。

```
l2vpn
 pw-class gre
  encapsulation mpls
  preferred-path interface tunnel-ip 1 fallback disable
```

## 疑似回線の優先パスとしての GRE トンネルの設定

疑似回線の優先パスとして GRE トンネルを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **pw-class** { *name* }
4. **encapsulation mpls**
5. **preferred-path** { **interface** } { **tunnel-ip** *value* | **tunnel-te** *value* | **tunnel-tp** *value* } [ **fallback disable** ]
6. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **pw-class** { *name* }

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# pw-class gre
```

疑似回線クラス名を設定します。

**ステップ 4** **encapsulation mpls**

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc)# encapsulation mpls
```

MPLS に疑似回線カプセル化を設定します。

**ステップ 5** **preferred-path** { **interface** } { **tunnel-ip** *value* | **tunnel-te** *value* | **tunnel-tp** *value* } [ **fallback disable** ]

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pwc-encap-
mpls)# preferred-path interface tunnel-ip 1 fallback disable
```

優先パス トンネルを設定します。フォールバックのディセーブル化の設定が使用されており、優先パスとして設定されている TE/TP トンネルがダウン状態になると、対応する疑似回線もダウン状態になることがあります。

（注） フォールバックがサポートされていることを確認します。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# VPLS LSM の設定：例

このセクションでは、VPLS LSM ソリューションを設定する方法の例を示します。

## VFI で RSVP-TE を使用した P2MP PW の有効化：例

次に、VFI で RSVP-TE を使用した P2MP PW を有効にする例を示します。

```
configure
l2vpn
 bridge group {bridge group name}
  bridge-domain {bridge domain name}
   vfi {vfi name}
   multicast p2mp
   transport rsvp-te
   attribute-set p2mp-te set1
commit
!
```

## VFI での P2MP PW の BGP 自動検出シグナリングの有効化：例

次に、VFI で P2MP PW の BGP 自動検出シグナリングを有効にする例を示します。

```
configure
l2vpn
 bridge group bg1
   bridge-domain bd1
     vfi vfi1
     multicast p2mp
     signaling protocol bgp
 commit
!
```

## VPN ID の設定：例

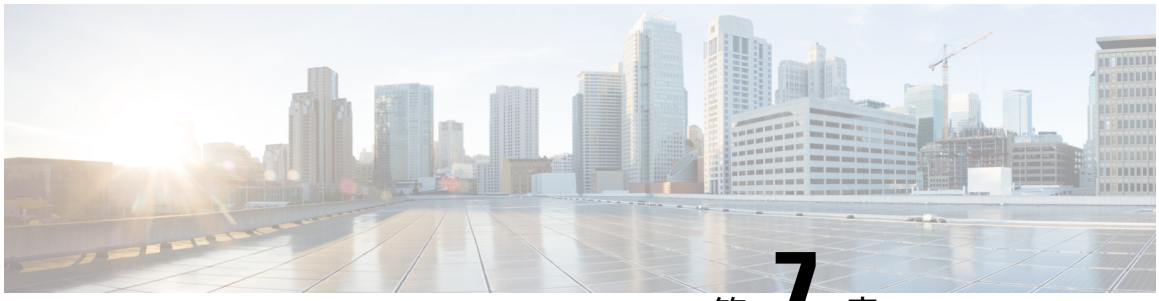
次に、VPN ID を設定する例を示します。

```
l2vpn

 bridge group bg1
  bridge-domain bd1
   interface GigabitEthernet0/1/0/0.1
   !
   interface GigabitEthernet0/1/0/0.2
   !
   vfi 1
    vpn-id 1001
    autodiscovery bgp
      rd auto
      route-target 1.1.1.1
      signaling protocol bgp
    !
   !
```

## IGMP スヌーピングの設定：例

次に、IGMP スヌーピングを設定する例を示します。

```
igmp snooping profile profile1
 [no] default-bridge-domain all enable
!
l2vpn
  bridge group bg1
   bridge domain bd1
    [no] igmp snooping disable
    !
   !
  !
 !
```



第 7 章

# IEEE 802.1ah プロバイダー バックボーン ブリッジの実装

このモジュールでは、Cisco ASR 9000 シリーズ ルータでの IEEE 802.1ah プロバイダー バックボーン ブリッジの概念および設定情報を提供します。IEEE 802.1ah 規格（Ref（4））は、大規模エンドツーエンド レイヤ 2 プロバイダー ブリッジ型ネットワークを構築するために、複数のプロバイダー ブリッジ型ネットワークを相互接続する手段を提供します。

Cisco ASR 9000 シリーズ アグリゲーション サービス ルータは現在、プロバイダー バックボーン ブリッジが VPLS ネットワークである場合のシナリオをサポートします。また、PBB エッジ ブリッジ ドメインおよびコア ブリッジ ドメインの疑似回線を設定できます。いずれのブリッジ ドメインでも、疑似回線の機能はネイティブ ブリッジ ドメインの場合と同様です。

**IEEE 802.1ah プロバイダー バックボーン ブリッジを実装するための機能の履歴**

| リリース | 変更内容 |
|---|---|
| リリース 3.9.1 | この機能が、Cisco ASR 9000 シリーズ ルータで導入されました。 |
| リリース 4.3.0 | 次の機能のサポートが追加されました。<br>• プロバイダーバックボーンブリッジVPLS<br>• Multiple I-SID Registration Protocol Lite（MIRP Lite） |
| リリース 4.3.2 | PBB-EVPN機能のサポートが追加されました。 |
| リリース 5.1.2 | PBB VPLS フラッディング最適化機能の MMRP のサポートが追加されました。 |

**サポート対象ハードウェア**

| 機能名 | ASR 9000 イーサネットライン カード | ASR 9000 拡張イーサネットライン カード |
|---|---|---|
| 基本的な PBB | 対応 | 対応 |

| 機能名 | ASR 9000 イーサネットライン カード | ASR 9000 拡張イーサネットライン カード |
|---|---|---|
| Multiple I-SID Registration Protocol Lite | なし | 対応 |
| PBB VPLS | なし | 対応 |
| PBB EVPN | なし | 対応 |
| PBB VPLS フラッディング最適化の MMRP | なし | 対応 |



# 802.1ah プロバイダー バックボーン ブリッジを実装するための前提条件

この前提条件は、802.1ah プロバイダー バックボーン ブリッジの実装に適用されます。

- 適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。

  ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

- マルチポイントブリッジングの概念に関する知識が必要です。「マルチポイント レイヤ 2 サービスの実装」モジュールを参照してください。

# 802.1ah サービス プロバイダー バックボーン ブリッジの実装に関する情報

802.1ah を実装するには、次の概念を理解している必要があります。

## IEEE 802.1ah 規格の利点

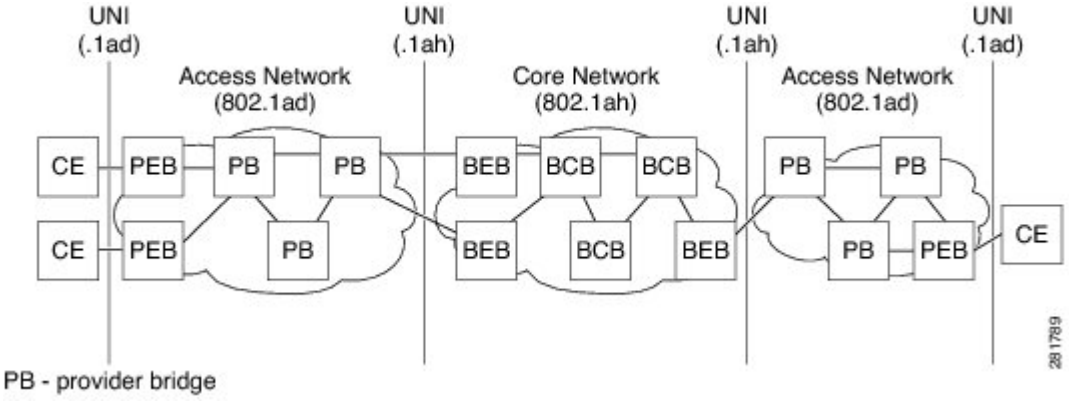
IEEE 802.1ah プロバイダー バックボーン ブリッジの利点を以下に示します。

- サービスインスタンスの拡張性の向上：サービスプロバイダーのプロバイダーブリッジ型ネットワーク（PBN）でのサービス（サービス VLAN またはサービスインスタンス）の数を拡張できます。
- MAC アドレスの拡張性：MAC アドレスなどのカスタマーパケットを、新しい MAC アドレス（バックボーンブリッジ MAC アドレス）を持つ新しいイーサネットフレームにカプセル化します。これは、バックボーンコアブリッジが顧客ごとにすべての MAC アドレスを学習する必要性をなくし、バックボーンエッジブリッジの負荷を軽減します。
- VPLS 疑似回線の低減およびメッシュ拡張性：IP/MPLS コアの疑似回線の数を大幅に削減できます。これは、単一の VPLS サービスが複数のカスタマー サービス インスタンスを転送できるようになり、多数のカスタマーサービスを転送するために必要な IP/MPLS コア内の疑似回線の数が少なくなるためです。
- レイヤ 2 バックボーン トラフィック エンジニアリング：サービス識別機能を分離して I タグに移動することで、レイヤ 2 トラフィック エンジニアリングを明示的に制御できます。これにより、バックボーン VLAN をレイヤ 2 トラフィック エンジニアリング機能で使用できます。
- ポイントツーポイント サービスの拡張性および最適化：サービス多重化の複数のオプションとエンドポイント検出を含むポイントツーポイント サービスの実装をイネーブルにします。
- バックボーンのフラッディングトラフィックの削減：ネットワークのコアの MAC アドレス数が少ないことにより、トポロジ変更で MAC テーブルがフラッシュされると、再学習される MAC アドレスの数が少ないためコアネットワークのフラッディングトラフィックの量が削減されます。

# IEEE 802.1ah 規格プロバイダー バックボーン ブリッジ概要

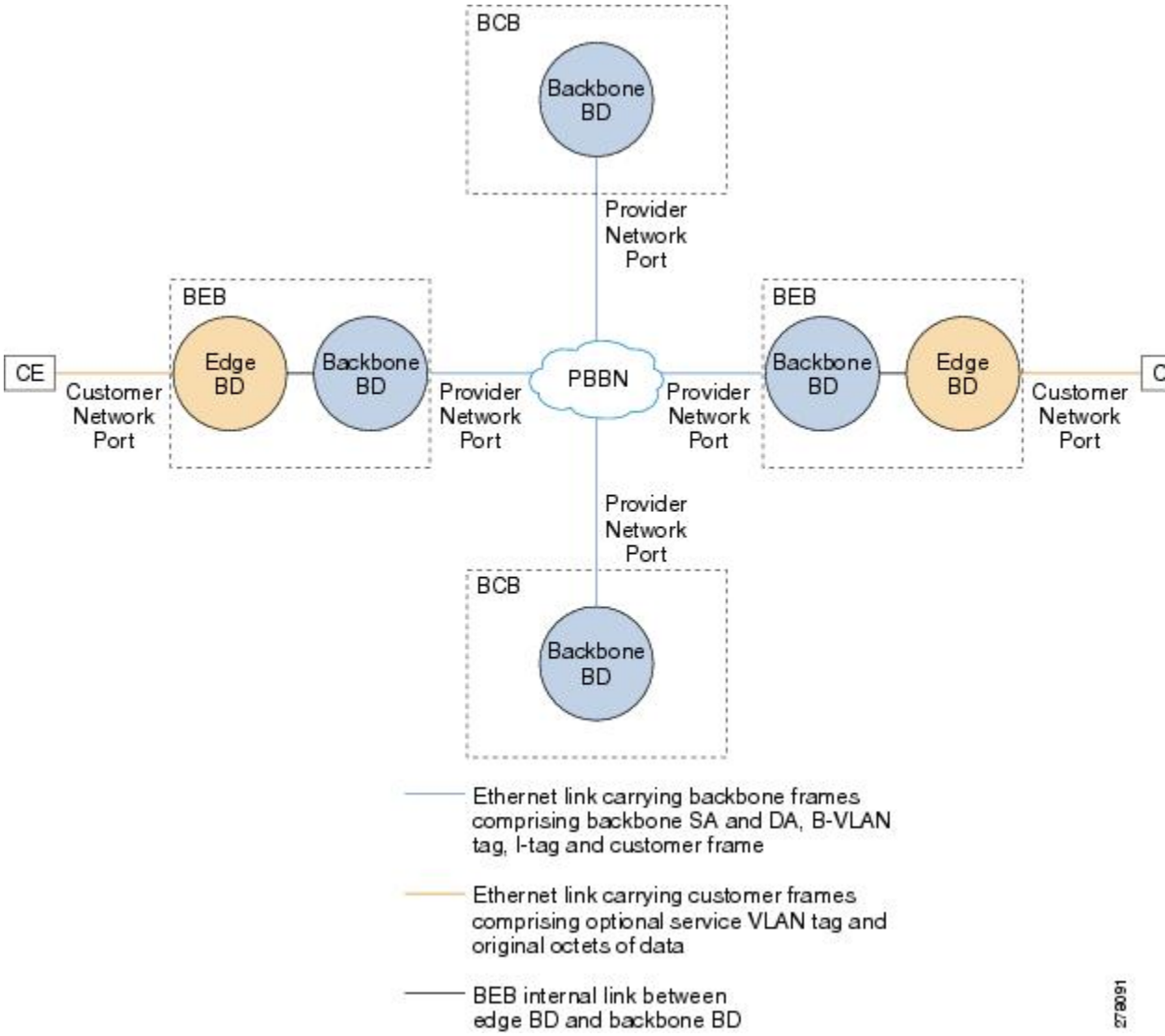
IEEE 802.1ah プロバイダー バックボーン ブリッジ機能は、プロバイダー バックボーン ブリッジ型ネットワーク（PBBN）のエッジで、バックボーンエッジブリッジ（BEB）のエンドユーザ トラフィックをカプセル化またはカプセル化解除します。バックボーンコアブリッジ（BCB）ベースのネットワークは、PBBN 内での IEEE 802.1ah カプセル化フレームの内部転送を提供します。次の図は、一般的な 802.1ah PBB のネットワークを表しています。

**図 *40 : IEEE 802.1ah* プロバイダー バックボーン ブリッジ**



次の図は、一般的なプロバイダーのバックボーン ネットワーク トポロジを表しています。

**図 *41* : プロバイダー バックボーン ネットワークのトポロジ**




# バックボーン エッジ ブリッジ

バックボーンエッジブリッジ（BEB）には、I-Component または B-Component を含めることができます。I-Component は、サービス VLAN ID（S-VID）をサービスインスタンス ID（I-SID）にマッピングし、バックボーン VLAN タグ（B-Tag）なしのプロバイダーバックボーンブリッジ（PBB）ヘッダーを追加します。B-Component は、I-SID をバックボーン VID（B-VID）にマッピングし、B-Tag を持つ PBB ヘッダーを追加します。

IEEE 802.1ah 規格では、次の 3 つのタイプの BEB が指定されています。

- B-BEB には、MAC-in-MAC ブリッジの B-Component が含まれます。これは、I-SID を検証し、フレームをバックボーン VLAN（B-VLAN）にマッピングします。また、コアブリッジ内の B-VLANS に基づいてトラフィックを切り替えます。
- I-BEB には、MAC-in-MAC ブリッジの I-Component が含まれます。これは、B-MAC カプセル化を実行し、プロバイダー VLAN タグ（S-tag）、カスタマー VLAN タグ（C-Tag）、または S-tag/C-tag のペアに基づいて I-SID を挿入します。
- IB-BEB には、LAN セグメントによって相互接続された 1 つ以上の I-Component と 1 つの B-Component が含まれます。

（注） Cisco ASR 9000 シリーズ ルータでは、IB-BEB のみがサポートされています。Cisco IOS XR は、エッジノードで IB-BEB ブリッジタイプをサポートします。

# IB-BEB

IB-BEB には、I-Component と B-Component の両方が含まれます。このブリッジは、B-MAC を選択し、プロバイダー VLAN タグ（S-tag）、カスタマー VLAN タグ（C-Tag）、または S-tag と C-Tag の両方に基づいて I-SID を挿入します。これは、I-SID を検証し、B-VLAN 上でフレームを送受信します。

IEEE 802.1ah プロバイダー バックボーン ブリッジ機能は、IEEE 802.1ah 規格で要求されるすべてのサービスをサポートし、さらにサービスを拡張して次の追加機能を提供します。

- S-Tagged サービス：
  - 多重化環境では、各 S-tag が I-SID にマッピングされ、各 S-tag は保持または削除できます。
  - バンドル環境では、複数の S-tag が同じ I-SID にマッピングされ、S-tag は保持する必要があります。
- C-Tagged サービス
  - 多重化環境では、各 C-tag が I-SID にマッピングされ、各 C-tag は保持または削除できます。
  - バンドル環境では、複数の C-tag が同じ I-SID にマッピングされ、C-tag は保持する必要があります。
- S/C-Tagged サービス：
  - 多重化環境では、各 S-tag/C-tag ペアが I-SID にマッピングされます。S-tag または S-tag/C-tag ペアは、保持または削除できます。
  - バンドル環境では、複数の S-tag/C-tag ペアが同じ I-SID にマッピングされ、S-tag/C-tag ペアは保持する必要があります。
- ポートベースのサービス
  - ポートベースのサービス インターフェイスは、カスタマー ネットワーク ポート（CNP）で提供されます。ポートベースのサービス インターフェイスは、C-VLAN ブ

リッジ、802.1dブリッジ、ルータ、またはエンドステーションに接続できます。このインターフェイスが提供するサービスは、単一のバックボーン サービス インスタンスのバックボーン上で、S-Tag なしですべてのフレームを転送します。ポートベースインターフェイスは、ヌル以外の VLAN ID を持つ S タグを含むすべてのフレームをドロップします。

次に、ポートベースのサービスを設定する例を示します。

```
 interface GigabitEthernet0/0/0/10.100 l2transport
 encapsulation untagged
```

--> タグなしフレームの EFP を作成します。

```
interface GigabitEthernet0/0/0/10.101 l2transport
encapsulation dot1ad priority-tagged
```

--> ヌルの S-tag 付きフレームの EFP を作成します。

```
interface GigabitEthernet0/0/0/10.102 l2transport
encapsulation dot1q priority-tagged
```

--> ヌルの C-tag 付きフレームの EFP を作成します。

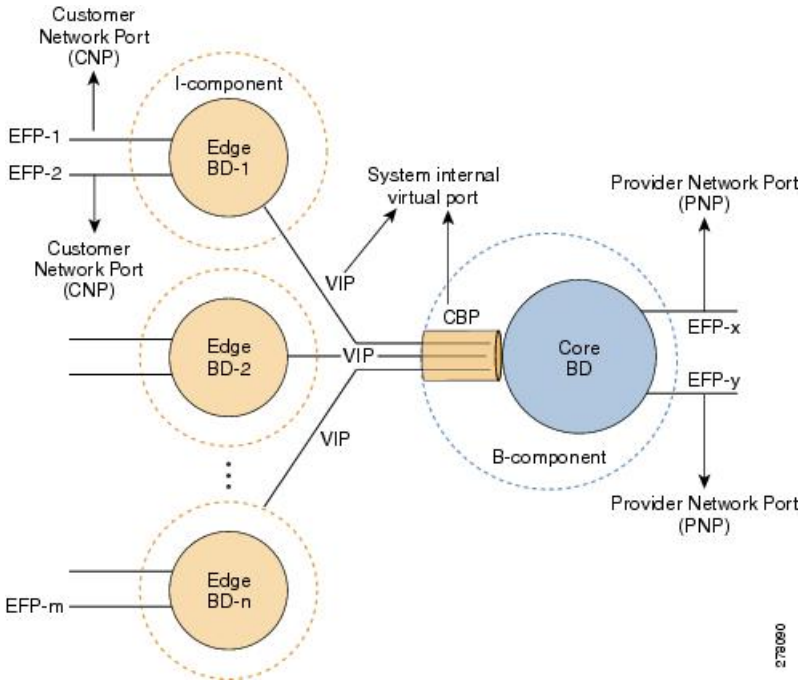
```
interface GigabitEthernet0/0/0/10.103 l2transport
encapsulation dot1q any
```

--> C-tag 付きフレームの EFP を作成します。

（注） ポートベースのサービスを設定するには、上記のすべての EFP を、同じエッジ ブリッジ ドメインに追加する必要があります。

次の図は、Cisco ASR 9000 シリーズ ルータの PBB ブリッジ コンポーネント トポロジを示しています。

図 *42 : Cisco ASR 9000* シリーズ ルータの *PBB* ブリッジ コンポーネント トポロジ




# Multiple I-SID Registration Protocol Lite

802.1Qbe マルチ I-SID 登録プロトコル（MIRP）規格は、I-SID ごとに I-Component のフィルタリング データベースに保持される学習された MAC アドレスの登録エントリをフラッシュする機能を提供します。バックボーン サービス インスタンス ID（I-SID）は、フレームのバックボーン サービス インスタンスを一意に識別するバックボーン サービス インスタンス タグのフィールドです。MIRP は I-SID フラッシュのメカニズムを定義し、プロバイダー バックボーンブリッジ型ネットワークに接続されたネットワークで発生するトポロジ変更を処理するために必要な機能を備えています。バックボーン エッジ ブリッジ（BEB）は、影響を受ける可能性のある（カスタマー MAC アドレスとバックボーン MAC アドレスについて、学習した特定の関連付けを変更する必要のある）他の BEB に信号を送信します。MIRP がない場合、プロバイダー バックボーン ネットワーク上のカスタマー接続では、アクセス ネットワークでのトポロジの変更後の接続の復元に数分かかることがあります。

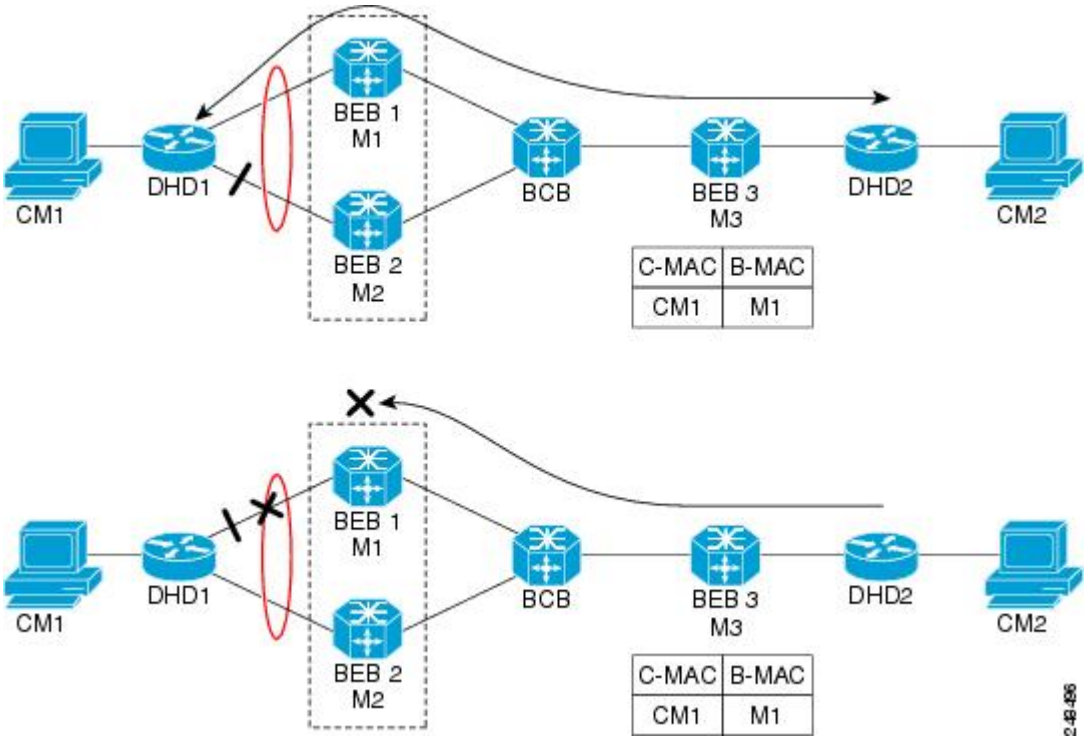
以前のリリースでは、PBB エッジ ブリッジ ドメインでポートが使用不可能になるかスパニングツリー トポロジが変更されることによりブリッジ フォワーディング トポロジの変更が発生すると、PBB トラフィックが MAC エージング サイクルにドロップされました。このため、PBB ブリッジの使用は厳しく制限されていました。

Cisco ASR 9000 シリーズ アグリゲーション サービス ルータは、Multiple I-SID Registration Protocol Lite（MIRP Lite）と呼ばれる MIRP プロトコルの簡略化された実装をサポートしています。MIRP Lite 機能は、サイトでのトポロジ変更の検出をイネーブルにします。サイトがトポロジの変更を検出すると、特別に定義されたパケットは、PBB ネットワークのすべてのリモート エッジ サイトにフラッディングされます。送信者のサイトでは、MAC フラッシュを必要とする I-SID を指定するために、I-Component の I-SID がフレーム ヘッダーの I-TAG に配置

されます。受信者のサイトでは、各 PBB エッジ スイッチが I-SID のチェックを実行します。I-SID が I-Component の 1 つと一致すると、I-Component の MAC がフラッシュされます。

802.1ah ネットワーク内での MIRP の使用を次の図に示します。

図 *43 : 802.1ah* ネットワーク内での *MIRP*




デバイス DHD1 は、2 つの 802.1ah バックボーン エッジ ブリッジ（BEB1 と BEB2）にデュアルホーム接続しています。当初のプライマリ パスはBEB1経由であると想定しています。この構成では、BEB3 は、DHD1 の背後にあるホスト（MAC アドレスは CM1）は、宛先 B-MAC M1 を介して到達できることを学習しています。DHD1 と BEB1 間のリンクに障害が発生し、DHD1 の背後にあるホストが非アクティブのままになっていると、BEB3 の MAC キャッシュ テーブルは、新規のパスビューがB-MACアドレスM2のBEB2経由であっても、BEB1のMACアドレスを引き続き参照します。DHD2の背後にあるホストからDHD1の背後にあるホストに転送されたブリッジトラフィックは、誤ってB-MAC M1 でカプセル化され、MAC トンネルを経由して BEB1 に送信されて、トラフィックがドロップされています。

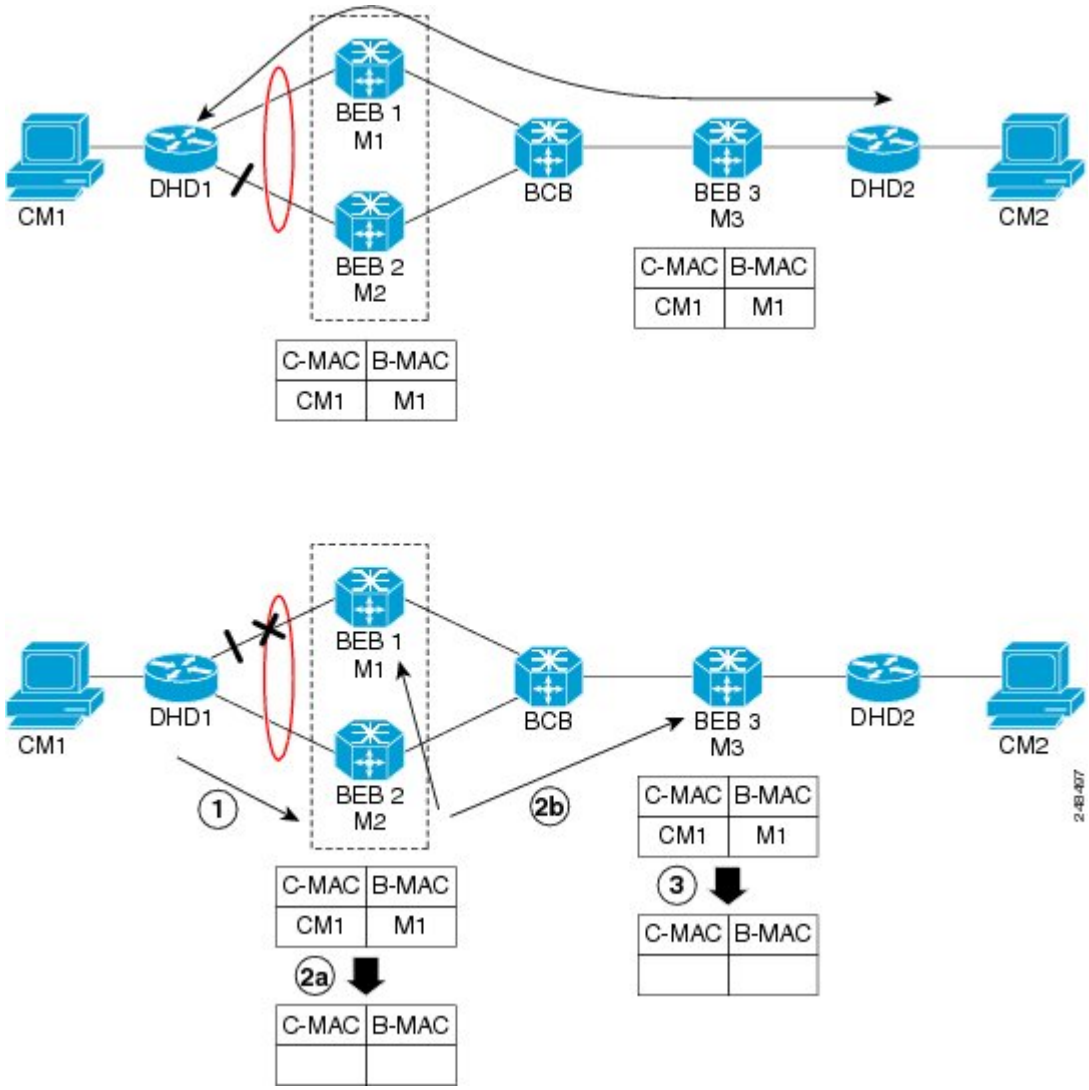
DHD1 と BEB1 間のリンクに障害が発生した場合にトラフィックがドロップされないように、BEB2 は次の 2 つのタスクを実行します。

- サービスに対して固有の MAC アドレステーブルをフラッシュします。
- MIRP パケットを受信するリモート PE に、固有の MAC テーブルのクリアを要求します。MIRP メッセージは、バックボーンコアブリッジ（BCB）に対して透過的です。MIRP メッセージは BEB 上で処理されます。BCB だけが B-MAC アドレスに基づいた取得と転送を行っており、C-MAC アドレスでは認識されないためです。

（注） MIRP は、ネイティブ 802.1ah と VPLS 経由の PBB の両方に C-MAC アドレスフラッシュをトリガーします。

次の図に MIRP の動作を示します。

**図 *44 : MIRP* 動作**




# プロバイダー バックボーン ブリッジング イーサネット VPN

プロバイダー バックボーン ブリッジング イーサネット VPN（PBB-EVPN）は、復元力と転送ポリシーの要件に対応する次世代 L2VPN ソリューションです。この機能により、マルチホーミングの詳細オプション、マルチパスのサポート、およびユーザ定義の BGP ポリシー機能もイーサネット L2VPN で使用できるようになりました。PBB-EVPN は、BGP を使用してパケッ

ト交換ネットワーク（PSN）で MAC アドレスの配布と習得を行います。PBB-EVPN は、これらキャリアイーサネットおよびデータセンターの相互接続の要件に対応する、PBB とイーサネット VPN の機能を組み合わせたものです。

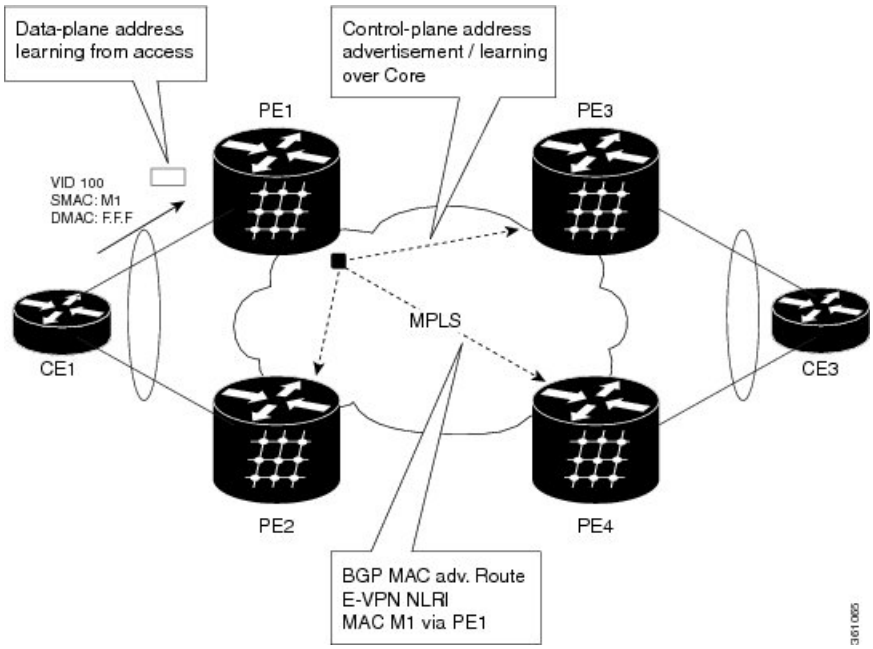
- 全アクティブ冗長性とロードバランシング
- シンプルなプロビジョニングと運用
- 最適な転送
- 短時間でのコンバージェンス
- MAC アドレスの拡張性

## イーサネット VPN

イーサネット仮想プライベートネットワーク（EVPN）は、組織内の複数のサイトの安全でプライベートな接続のためのソリューションです。EVPN サービスは、イーサネットテクノロジーの利点をワイドエリアネットワーク（WAN）にまで拡大します。このサービスは、MPLS ネットワークを介して提供されます。

EVPN を使用すると、仮想プライベートネットワーク上のルーティングを管理して、完全な制御とセキュリティを実現できます。EVPN は、MPLS/IP ネットワークを介して顧客またはクライアントの MAC アドレスの到達可能性情報を配信するために BGP を使用する、高度なマルチホーミング機能を備えたマルチポイント L2VPN サービスのソリューションを導入します。EVPN は顧客の各 MAC アドレスを BGP ルートとしてアドバタイズするため、MAC アドレスに対する BGP ポリシー制御が可能になります。

**図 *45 : BGP* での *MAC* 配信（*EVPN*）**



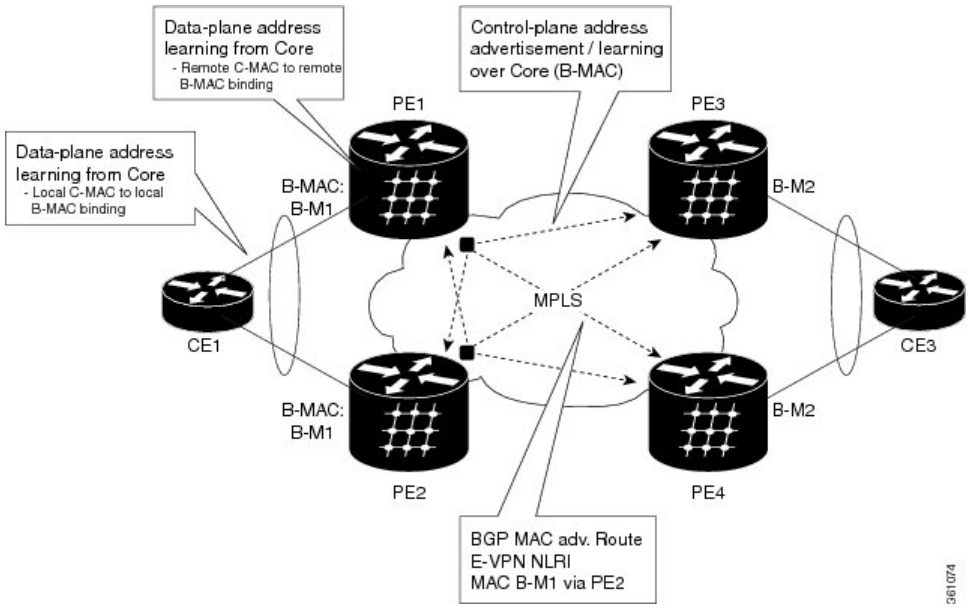

上の図では、プロバイダーエッジ（PE）ルータがマルチプロトコル BGP を実行し、MPLS を介して MAC アドレスをアドバタイズおよび学習しています。顧客の MAC アドレスは、接続回線（顧客のデバイスを PE に接続するリンク）を介してデータプレーンで学習されます。次

に、MAC アドレスが、EVPN インスタンスを識別する MPLS ラベルにより、BGP を使用して MPLS を介して配信されます。

## PBB-EVPN の概要

PBB-EVPN ソリューションは、イーサネットプロバイダーバックボーンブリッジ（PBB - IEEE 802.1ah）とイーサネット VPN を組み合わせたもので、PE が PBB バックボーンエッジブリッジ（BEB）として機能します。PE は、接続回線から 802.1Q イーサネットフレームを受信します。これらのフレームは、PBB ヘッダーにカプセル化され、IP/MPLS コアを介して転送されます。出力側（EVPN PE）では、MPLS ディスポジション後に PBB ヘッダーが削除され、元の 802.1Q イーサネットフレームが顧客機器に配信されます。

**図 *46 : PBB-EVPN* ネットワーク**




PE ルータは、次の機能を実行します。

- 通常のブリッジ動作ごとに、データプレーンの接続回線を介して顧客またはクライアントの MAC アドレス（C-MAC）を学習します。
- コアからのトラフィック入力から、データプレーン内のリモート C-MAC からバックボーン MAC（B-MAC）へのバインディングを学習します。
- BGP 内のローカル B-MAC アドレス到達可能性情報を、同じサービスインスタンスのセット内の他のすべての PE ノードにアドバタイズします。各 PE には、デバイスを一意に識別する一連のローカル B-MAC アドレスがあることに注意してください。
- 受信したリモート BGP アドバタイズメントから転送テーブルを作成し、リモート B-MAC アドレスとリモート PE IP アドレスを関連付けます。

アクセスで顧客の MAC アドレスを制限することにより、PBB-EVPN には何百万もの顧客の MAC アドレスがある大規模ネットワークのための高い拡張性があります。コアでは B-MAC アドレスだけがアドバタイズされ、交換される BGP ルートが管理が容易な数になります。

PBB EVPNの場合、B-MACフラッシュはイーサネットVPNインスタンス（EVI）ごとのB-MACごとです。

### EVPN インスタンス

E-VPN インスタンス（EVI）は、MPLS/IP ネットワーク内の VPN を識別します。コアブリッジごとに 1 つの EVI のみが存在できます。

### イーサネットセグメント

イーサネットセグメントは、1 つ以上の PE に接続されたサイトです。イーサネットセグメントは、次のような単一のデバイス（カスタマーエッジ（CE））またはネットワーク全体になります。

- シングルホームデバイス（SHD）
- イーサネット マルチシャーシ リンク アグリゲーション グループを使用したマルチホームデバイス（MHD）
- シングルホームネットワーク（SHN）
- マルチホームネットワーク（MHN）

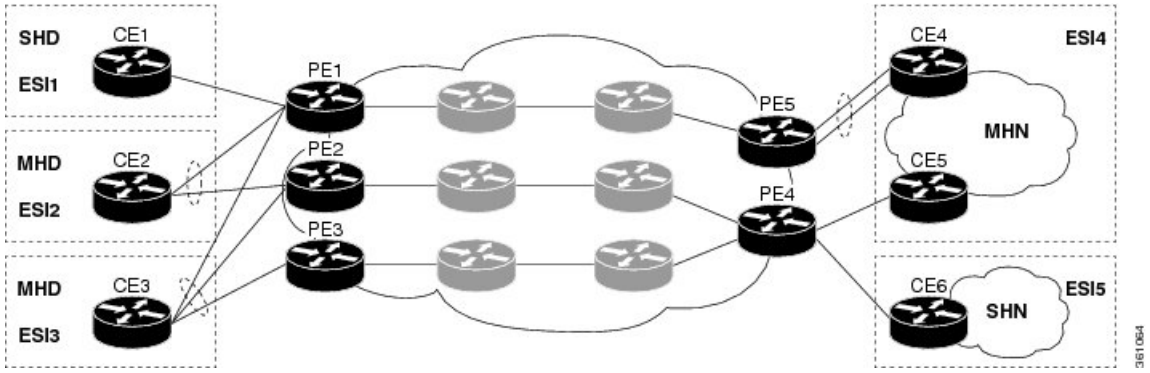
イーサネットセグメントは、10バイトのグローバルイーサネットセグメント識別子（ESI）によって一意に識別されます。

ESI の形式は RFC 7432 に準拠しています。ESI 値は ESI タイプによって異なります。現在は、ESI タイプ 0 および 1 のみがサポートされています。次の表に、ESI タイプに基づく ESI の形式を示します。

| ESI のタイプ | 説明 | ESI の形式 |
|---|---|---|
| タイプ 0 | 設定に基づく任意の ESI 値 | 1 オクテット ESI タイプ 0x00<br>9 オクテット ESI 値 |
| タイプ 1 | LACP に基づいて自動生成された ESI 値 | 1 オクテット ESI タイプ 0x01<br>6 オクテット CE LACP MAC アドレス<br>2 オクテット CE LACP ポートキー<br>0x00 の 1 オクテット値 |

次の図は、イーサネットセグメントと ESI の例を示しています。

図 *47*: イーサネットセグメント




### PBB-EVPN BGP ルート

PBB-EVPNは、さまざまなルートのタイプを新しい属性とともにアドバタイズするために使用される、単一の新しい BGP ネットワーク層到達可能性情報（NLRI）を定義します。

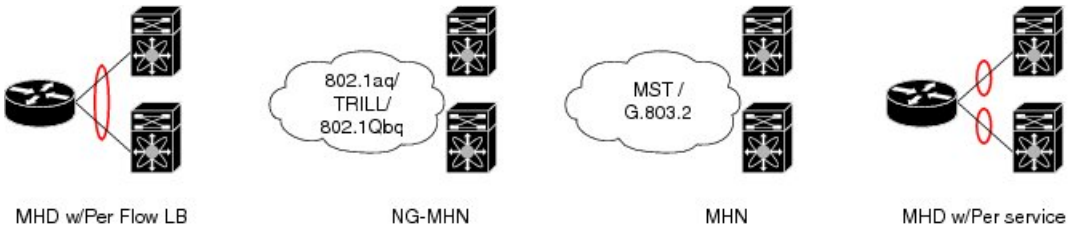
### 指定フォワーダ選択

デュアルホームまたはマルチホームのデバイスまたはネットワークで指定されたフォワーダを決定するために、指定フォワーダ（DF）選択メカニズムが使用されます。この選択は、サービスごとに実行されます。MHN の DF フィルタリング機能は、MHD の機能とは次の点が異なります。

- 方向性：MHN の DF フィルタリングは、アクセス側のイーサネット インターフェイスでの入力と出力両方のトラフィックに適用されます。一方、MHD の DF フィルタリングは、アクセス側のインターフェイスから出力されるトラフィックにのみ適用されます。
- トラフィックタイプ：MHN の DF フィルタリングは、ユニキャストトラフィックとフラッディングされたマルチ宛先トラフィックの両方に影響します。一方、MHD の DF フィルタリングは、フラッディングされたマルチ宛先トラフィックにのみ適用されます。

次の図は、MHN および MHD のさまざまな DF フィルタリングルールを示しています。

図 *48 : MHN/MHD* の *DF* フィルタリングの比較




| Scenario | MHD w/Per Flow LB | MHN (always treated as SHN) | MHD w/Per service LB |
|---|---|---|---|
| Filtering Direction (on AC) | Egress | Egress<br>Ingress | Egress<br>Ingress |
| Filtered Traffic | Multicast | Multicast<br>Unicast | Multicast<br>Unicast |
| Granularity | EFP | EFP | EFP |

361063

### アクセス自動検知

マルチホームデバイスまたはデュアルホームデバイスに接続されている PE は、フローベースのロードバランシングとも呼ばれる、アクティブ-アクティブフロー単位をサポートしている可能性があります。PEは、物理ポートまたはバンドルポートを介してCEにサービスを提供します。イーサネットセグメント識別子はポートごとに割り当てられます。この値は、CE システムプライオリティ、CE システム ID、CE ポートキーなどの情報を使用して、接続された CE から計算されます。PE は、ロードバランシングのタイプを判別するためにアクセストポロジを自動検出する必要があります。ロードバランシングは、アクティブ-アクティブフロー単位ロードバランシング、サービス単位ロードバランシング、または単にロードバランシングなしにすることができます。

# PBB VPLS フラッディング最適化の MMRP

PBBネットワークでは、デバイスがトラフィックの宛先であるサービスインスタンスをホストしていない場合でも、トラフィック（不明なユニキャスト、マルチキャスト、またはブロードキャスト）がネットワーク内のすべての PE デバイスにフラッディングされます。

PBB VPLS フラッディング最適化機能のマルチMAC登録プロトコル（MMRP）は、特定のサービスインスタンスに関連するPEデバイスにのみトラフィックを送信することで、PEデバイスでのフラッディングトラフィックの影響を最適化します。

VPLS 経由の PBB ネットワークでは、PE デバイス間のトラフィックは、フル メッシュ ネットワーク トポロジですべての PE デバイスを接続する MPLS 疑似回線を介して転送されます。

プロバイダー バックボーン ネットワーク トポロジの図は、一般的な 802.1ah PBB ネットワークを示しています。

すべての I-SID（サービスインスタンス VLAN ID）にグループ B-MAC アドレスと呼ばれる対応するマルチキャスト MAC アドレスがあり、これは I-SID に基づいて導出されます。グループ B-MAC アドレスは、プロバイダーバックボーン全体にフラッディングトラフィックを伝播するときに、外部 MAC ヘッダーで宛先アドレスとして使用されます。

PE デバイスによって MMRP が使用され、ホストするサービスインスタンスの I-SID に対応するグループ B MAC アドレスのセットについて相互に通知されます。これにより、各デバイスは、どの疑似回線のセットのフラッディングトラフィックを転送するかを決定できます。これは、I-SID に対応するグループ B-MAC アドレスの MMRP 登録が受信された疑似回線です。

（注） PBB-VPLS フラッディング最適化機能は、PBB-VPLS ネットワークでのみ有効で、イーサネット経由の PBB ネットワークでは有効になりません。

# PBB-VPLS フラッディング最適化の設定

PBB-VPLS フラッディング最適化機能を設定するには、次の手順を実行します。

## PBB コアブリッジの PBB-VPLS フラッディング最適化の有効化

PBB コアブリッジで PBB-VPLS フラッディング最適化を有効にするには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *domain-name*
5. **pbb core**
6. **mmrp-flood-optimization**
7. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group pbb
```

名前付きブリッジ グループのコンフィギュレーション モードを開始します。このコマンドは、新しいブリッジ グループを作成するか、既存のブリッジ グループを変更します（ブリッジ グループが存在する場合）。ブリッジ グループは、ブリッジ ドメインを整理します。

**ステップ 4** **bridge-domain** *domain-name*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain pbb-core
```

名前付きブリッジ ドメインのコンフィギュレーション モードを開始します。このコマンドは、新しいブリッジ ドメインを作成するか、既存のブリッジ ドメインを変更します（ブリッジ ドメインが存在する場合）。

**ステップ 5** **pbb core**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# pbb core
```

ブリッジ ドメインを PBB コアとして設定し、PBB コア コンフィギュレーション サブモードを開始します。

このコマンドは、カスタマー ブリッジ ポート（CBP）と呼ばれる内部ポートを作成します。

このブリッジ ドメインのすべてのインターフェイス（ブリッジ ポート）は、プロバイダー ネットワーク ポート（PNP）として扱われます。

**ステップ 6** **mmrp-flood-optimization**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-pbb-core)# mmrp-flood-optimization
```

コアブリッジの PBB over VPLS 機能でフラッディング最適化を有効にします。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## 汎用 MRP プロトコルパラメータの設定

PBB-VPLS フラッディング最適化用に汎用 MRP プロトコルパラメータを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **mmrp-flood-optimization**
3. （オプション）**periodic transmit interval** *seconds*
4. （オプション）**join-time** *milliseconds*
5. （オプション）**leaveall-time** *seconds*
6. （オプション）**leave-time** *seconds*
7. （オプション）**flood-time** *seconds*
8. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **mmrp-flood-optimization**

例：

```
RP/0/RSP0/cpu 0: router(config)# mmrp-flood-optimization
```

コアブリッジの PBB over VPLS 機能でフラッディング最適化を有効にします。

**ステップ 3** （オプション）**periodic transmit interval** *seconds*

例：

```
RP/0/RSP0/cpu 0: router(config-mmrp-flood-opt)# periodic transmit interval 3
```

定期的なマルチ MAC 登録プロトコルデータユニット（MMRPDU）を有効にします。

**ステップ 4** （オプション）**join-time** *milliseconds*

例：

```
RP/0/RSP0/cpu 0: router(config-mmrp-flood-opt)# join time interval 300
```

すべてのアクティブ ポートの Join 時間を設定します。

**ステップ 5** （オプション）**leaveall-time** *seconds*

例：

```
RP/0/RSP0/cpu 0: router(config-mmrp-flood-opt)# leaveall-time 10
```

権限をすべてのアクティブ ポートの Leave all 時間を設定します。

**ステップ 6** （オプション）**leave-time** *seconds*

例：

```
RP/0/RSP0/cpu 0: router(config-mmrp-flood-opt)# leave-time 40
```

すべてのアクティブ ポート Leave 時間を設定します。

**ステップ 7** （オプション）**flood-time** *seconds*

例：

```
RP/0/RSP0/cpu 0: router(config-mmrp-flood-opt)# flood-time 1000
```

コアブリッジで PBB-VPLS フラッディング最適化機能が有効になっている場合に、コアブリッジ全体へのトラフィックのフラッディングが有効になります。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# 802.1ah プロバイダー バックボーン ブリッジを実装する方法

この項では、次の手順について説明します。

# 802.1ah プロバイダー バックボーン ブリッジの実装に関する制約事項

次の機能はサポートされていません。

- MAC-in-MAC 上での相互接続ベースのポイントツーポイント サービス
- 1 つのエッジ ブリッジと複数のコア ブリッジのマッピング
- I タイプのバックボーン エッジ ブリッジ（I-BEB）と B タイプのバックボーン エッジ ブリッジ（B-BEB）
- IEEE 802.1ah over VPLS
- シャーシごとの複数の送信元 B-MAC アドレス
- ネイティブの MPLS LSP カプセル化を通じた 802.1ah フォーマット パケットのダイレクト カプセル化

プロバイダー バックボーン ブリッジ イーサネット VPN（PBB-EVPN）を実装する場合は、次の追加の制限事項が適用されます。

- プロバイダーエッジルータおよびルートリフレクタルータは、同じIETFドラフトバージョンの L2VPN イーサネット VPN（EVPN）をサポートするソフトウェアを実行している必要があります。BGP ネットワーク層到達可能性情報（NLRI）エンコーディングの違いにより、以降のドラフトバージョンには、以前のバージョンとの後方互換性がありません。次の表に、さまざまなCisco IOS XR ソフトウェアリリースでサポートされているドラフトを示します。

| Cisco IOS XR ソフトウェア リリース | サポートされている L2VPN EVPN ドラフトバージョン | |
|---|---|---|
| | draft-ietf-l2vpn-evpn-04 | draft-ietf-l2vpn-evpn-06 |
| 5.1.1 以前のリリース | ✓ | — |
| 5.2.0 | ✓ | — |
| 5.1.2 以降のリリース（5.2.0 を除く） | — | ✓ |

# CNP および PNP ポートでのイーサネット フロー ポイントの設定

カスタマー ネットワーク ポート（CNP）またはプロバイダー ネットワーク ポート（PNP）にイーサネット フロー ポイント（EFP）を設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **interface** *type interface-path-id.subinterface* **l2transport**
3. **encapsulation dot1q** *vlan-id* または **encapsulation dot1ad** *vlan-id* または **encapsulation dot1ad** *vlan-id* **dot1q** *vlan-id*
4. **commit** コマンドまたは **end** コマンドを使用します。

手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface** *type interface-path-id.subinterface* **l2transport**

例:

```
RP/0/RSP0/cpu 0: router(config)# interface
GigabitEthernet0/0/0/10.100 l2transport
```

L2 スイッチングのインターフェイスを設定します。

**ステップ 3** **encapsulation dot1q** *vlan-id* または **encapsulation dot1ad** *vlan-id* または **encapsulation dot1ad** *vlan-id* **dot1q** *vlan-id*

例:

```
RP/0/RSP0/cpu 0: router(config-subif)# encapsulation dot1q 100
or
encapsulation dot1ad 100
or
encapsulation dot1ad 100 dot1q 101
```

一致する VLAN ID および EtherType をインターフェイスに割り当てます。

**ステップ 4** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# PBB エッジ ブリッジ ドメインおよびサービス インスタンス ID の設定

PBB エッジ ドメインおよびサービス ID を設定するには、次の作業を行います。

（注） PBB 機能を設定するには、admin ユーザ権限でログインし、**hw-module profile feature l2** コマンドを実行して、PBB 機能をサポートする ASR 9000 イーサネットラインカードの ucode バージョンを選択します。この設定を行わない限り、PBB 機能は、ASR 9000 イーサネット ラインカードでサポートされません。機能プロファイル設定の詳細については、『Cisco ASR 9000 Series Aggregation Services Router System Management Configuration Guide』を参照してください。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *domain-name*
5. **interface** *type interface-path-id.subinterface*
6. **pbb edge i-sid** *service-id* **core-bridge** *core-bridge-name*
7. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例:

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group pbb
```

名前付きブリッジ グループのコンフィギュレーション モードを開始します。このコマンドは、新しいブリッジ グループを作成するか、既存のブリッジ グループを変更します（ブリッジ グループが存在する場合）。ブリッジ グループは、ブリッジ ドメインを整理します。

**ステップ 4** **bridge-domain** *domain-name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain pbb-edge
```

名前付きブリッジ ドメインのコンフィギュレーション モードを開始します。このコマンドは、新しいブリッジ ドメインを作成するか、既存のブリッジ ドメインを変更します（ブリッジ ドメインが存在する場合）。

**ステップ 5** **interface** *type interface-path-id.subinterface*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# interface GigabitEthernet0/5/0/0.20
```

一致する VLAN ID および EtherType をインターフェイスに割り当てます。この EFP はエッジ ブリッジの CNP と見なされます。

**ステップ 6** **pbb edge i-sid** *service-id* **core-bridge** *core-bridge-name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# pbb edge i-sid 1000 core-bridge pbb-core
```

サービス ID および割り当てられたコア ブリッジ ドメインを指定して、ブリッジ ドメインを PBB エッジとして設定し、PBB エッジ コンフィギュレーション サブモードを開始します。

このコマンドは、指定したコア ブリッジ ドメインに PBB エッジ ブリッジ ドメインを関連付ける仮想インスタンス ポート（VIP）も作成します。

このブリッジ ドメインのすべてのインターフェイス（ブリッジ ポート）は、カスタマー ネットワーク ポート（CNP）として扱われます。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# PBB コア ブリッジ ドメインの設定

PBB コア ブリッジ ドメインを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *group-name*
4. **bridge-domain** *domain-name*
5. **interface** *type interface-path-id.subinterface*
6. **pbb core**
7. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

**例**：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モード を開始します。

**ステップ 2** **l2vpn**

例:

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *group-name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group pbb
```

名前付きブリッジ グループのコンフィギュレーション モードを開始します。このコマンドは、新しいブリッジ グループを作成するか、既存のブリッジ グループを変更します（ブリッジ グループが存在する場合）。ブリッジ グループは、ブリッジ ドメインを整理します。

**ステップ 4** **bridge-domain** *domain-name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain pbb-core
```

名前付きブリッジ ドメインのコンフィギュレーション モードを開始します。このコマンドは、新しいブリッジ ドメインを作成するか、既存のブリッジ ドメインを変更します（ブリッジ ドメインが存在する場合）。

**ステップ 5** **interface** *type interface-path-id.subinterface*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# interface GigabitEthernet0/5/0/0.20
```

一致する VLAN ID および EtherType をインターフェイスに割り当てます。

**ステップ 6** **pbb core**

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# pbb core
```

ブリッジ ドメインを PBB コアとして設定し、PBB コア コンフィギュレーション サブモードを開始します。

このコマンドは、カスタマー ブリッジ ポート（CBP）と呼ばれる内部ポートを作成します。

このブリッジ ドメインのすべてのインターフェイス（ブリッジ ポート）は、プロバイダー ネットワーク ポート（PNP）として扱われます。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# PBB コア ブリッジ ドメイン下でのバックボーン VLAN タグの設定

PBB コア ブリッジ ドメイン下でバックボーン VLAN タグを設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *domain-name*
5. **interface** *type interface-path-id.subinterface*
6. **interface** *type interface-path-id. subinterface*
7. **pbb core**
8. **rewrite ingress tag push dot1ad** *vlan-id* **symmetric**
9. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

**例**：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

**例**：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group pbb
```

名前付きブリッジ グループのコンフィギュレーション モードを開始します。このコマンドは、新しいブリッジ グループを作成するか、既存のブリッジ グループを変更します（ブリッジ グループが存在する場合）。ブリッジ グループは、ブリッジ ドメインを整理します。

**ステップ 4** **bridge-domain** *domain-name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain pbb-core
```

名前付きブリッジ ドメインのコンフィギュレーション モードを開始します。このコマンドは、新しいブリッジ ドメインを作成するか、既存のブリッジ ドメインを変更します（ブリッジ ドメインが存在する場合）。

**ステップ 5** **interface** *type interface-path-id.subinterface*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# interface GigabitEthernet0/5/0/0.20
```

一致する VLAN ID および EtherType をインターフェイスに割り当てます。

**ステップ 6** **interface** *type interface-path-id. subinterface*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)# interface GigabitEthernet0/5/0/1.15
```

ブリッジ ドメインにインターフェイスを追加し、パケットの転送と、同じブリッジ ドメイン内の他のインターフェイスからのパケットの受信を可能にします。これで、このインターフェイスは、このブリッジ ドメイン上の接続回線になります。

**ステップ 7** **pbb core**

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# pbb core
```

ブリッジ ドメインを PBB コアとして設定し、PBB コア コンフィギュレーション サブモードを開始します。

このコマンドは、カスタマー ブリッジ ポート（CBP）と呼ばれる内部ポートを作成します。

このブリッジ ドメインのすべてのインターフェイス（ブリッジ ポート）は、プロバイダー ネットワーク ポート（PNP）として扱われます。

**ステップ 8** **rewrite ingress tag push dot1ad** *vlan-id* **symmetric**

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-pbb-core)# end
```

Mac-in-MAC フレームのバックボーン VLAN タグを設定し、また、タグの書き換えポリシーを設定します。

（注） コア ブリッジ ドメインのすべての PNP で同じバックボーン VLAN を使用します。

**ステップ 9** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# バックボーン送信元 MAC アドレスの設定

バックボーン送信元 MAC アドレス（B-SA）は、バックボーン ネットワークの一意のアドレスです。各 Cisco ASR 9000 シリーズ ルータは 1 つのバックボーン送信元 MAC アドレスを持ちます。B-SA が設定されていない場合、EEPROM の最も大きい MAC が PBB B-SA として使用されます。

（注） バックボーン送信元 MAC アドレスの設定は任意です。バックボーン送信元 MAC アドレスを設定しない場合、Cisco ASR 9000 シリーズ ルータは、シャーシバックプレーン MAC プールからデフォルトのバックボーン送信元 MAC アドレスを割り当てます。

バックボーン送信元 MAC アドレスを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **pbb**
4. **backbone-source-address** *mac-address*
5. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **pbb**

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# pbb
```

PBB コンフィギュレーション モードを開始します。

**ステップ 4** **backbone-source-address** *mac-address*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-pbb)# backbone-source-address 0045.1200.04
```

バックボーン送信元 MAC アドレスを設定します。

**ステップ 5** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# PBB エッジ ブリッジ ドメイン下での不明ユニキャスト バックボーン MAC の設定

PBB エッジ ブリッジ ドメイン下で不明ユニキャスト バックボーン MAC を設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *domain-name*
5. **interface** *type interface-path-id.subinterface*
6. **pbb edge i-sid** *service-id* **core-bridge** *core-bridge-name*
7. **unknown-unicast-bmac** *mac-address*
8. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例:

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例:

```
RRP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group pbb
```

名前付きブリッジ グループのコンフィギュレーション モードを開始します。このコマンドは、新しいブリッジ グループを作成するか、既存のブリッジ グループを変更します（ブリッジ グループが存在する場合）。ブリッジ グループは、ブリッジ ドメインを整理します。

**ステップ 4** **bridge-domain** *domain-name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain pbb-edge
```

名前付きブリッジ ドメインのコンフィギュレーション モードを開始します。このコマンドは、新しいブリッジ ドメインを作成するか、既存のブリッジ ドメインを変更します（ブリッジ ドメインが存在する場合）。

**ステップ 5** **interface** *type interface-path-id.subinterface*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# interface GigabitEthernet0/5/0/0.20
```

一致する VLAN ID および EtherType をインターフェイスに割り当てます。

**ステップ 6** **pbb edge i-sid** *service-id* **core-bridge** *core-bridge-name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# pbb edge i-sid 1000 core-bridge pbb-core
```

サービス ID および割り当てられたコア ブリッジ ドメインを指定して、ブリッジ ドメインを PBB エッジとして設定し、PBB エッジ コンフィギュレーション サブモードを開始します。

このコマンドは、指定したコア ブリッジ ドメインに PBB エッジ ブリッジ ドメインを関連付ける仮想インスタンス ポート（VIP）も作成します。

このブリッジドメインのすべてのインターフェイス（ブリッジポート）は、カスタマーネットワーク ポート（CNP）として扱われます。

**ステップ 7** **unknown-unicast-bmac** *mac-address*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-pbb-edge)# unknown-unicast-bmac 1.1.1
```

不明ユニキャスト バックボーンの MAC アドレスを設定します。

（注） Tridentラインカードで、不明ユニキャストBMACを設定すると、マルチキャスト、ブロードキャスト、および不明ユニキャスト宛先 MAC アドレスを持つカスタマー トラフィックを転送するために、BMAC が使用されます。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# PBB エッジ ブリッジ ドメイン下でのスタティック MAC アドレスの設定

PBB エッジ ブリッジ ドメイン下でスタティック MAC アドレスを設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *domain-name*
5. **interface** *type interface-path-id.subinterface*
6. **interface** *type interface-path-id.subinterface*
7. **pbb edge i-sid** *service-id* **core-bridge** *core-bridge-name*
8. **static-mac-address** *cda-mac-address* **bmac** *bda-mac-address*
9. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

**例**：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例:

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn)#bridge group pbb
```

名前付きブリッジ グループのコンフィギュレーション モードを開始します。このコマンドは、新しいブリッジ グループを作成するか、既存のブリッジ グループを変更します（ブリッジ グループが存在する場合）。ブリッジ グループは、ブリッジ ドメインを整理します。

**ステップ 4** **bridge-domain** *domain-name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#bridge-domain pbb-edge
```

名前付きブリッジ ドメインのコンフィギュレーション モードを開始します。このコマンドは、新しいブリッジ ドメインを作成するか、既存のブリッジ ドメインを変更します（ブリッジ ドメインが存在する場合）。

**ステップ 5** **interface** *type interface-path-id.subinterface*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#interface GigabitEthernet0/5/0/0.20
```

一致する VLAN ID および EtherType をインターフェイスに割り当てます。

**ステップ 6** **interface** *type interface-path-id.subinterface*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)#interface GigabitEthernet0/5/0/1.15
```

ブリッジドメインにインターフェイスを追加し、パケットの転送と、同じブリッジドメイン内の他のインターフェイスからのパケットの受信を可能にします。これで、このインターフェイスは、このブリッジドメイン上の接続回線になります。

**ステップ 7** **pbb edge i-sid** *service-id* **core-bridge** *core-bridge-name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)#pbb edge i-sid 1000 core-bridge pbb-core
```

サービス ID および割り当てられたコア ブリッジ ドメインを指定して、ブリッジ ドメインを PBB エッジとして設定し、PBB エッジ コンフィギュレーション サブモードを開始します。

このコマンドは、指定したコア ブリッジ ドメインに PBB エッジ ブリッジ ドメインを関連付ける仮想インスタンス ポート（VIP）も作成します。

このブリッジ ドメインのすべてのインターフェイス（ブリッジ ポート）は、カスタマー ネットワーク ポート（CNP）として扱われます。

**ステップ 8** **static-mac-address** *cda-mac-address* **bmac** *bda-mac-address*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-pbb-edge)#static-mac-address 0033.3333.3333 bmac
0044.4444.4444
```

PBB エッジ サブモードで CMAC と BMAC のスタティック マッピングを設定します。

**ステップ 9** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# PBB VPLS の設定

PBB VPLS を設定するには、次の作業を実行します。

## I-Component のアクセス疑似回線の設定

PBB エッジ ブリッジ ドメイン下でスタティック MAC アドレスを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *domain-name*
5. **mac withdraw state-down**
6. **exit**
7. **interface** *type interface-path-id.subinterface*
8. **interface** *type interface-path-id.subinterface*
9. **neighbor** { *A.B.C.D* } **pw-id** *value*

10. **exit**
11. **pbb edge i-sid** *service-id* **core-bridge** *core-bridge-name*
12. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例:

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group pbb
```

ブリッジ グループ コンフィギュレーション モードを開始します。このコマンドは、新しいブリッジ グループを作成するか、既存のブリッジ グループを変更します（ブリッジ グループが存在する場合）。ブリッジ グループは、ブリッジ ドメインを整理します。

**ステップ 4** **bridge-domain** *domain-name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain pbb-edge
```

ブリッジ ドメイン コンフィギュレーション モードを開始します。このコマンドは、新しいブリッジ ドメインを作成するか、既存のブリッジ ドメインを変更します（ブリッジ ドメインが存在する場合）。

**ステップ 5** **mac withdraw state-down**

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# mac withdraw state-down
```

（任意）MAC 取り消しをイネーブルにします。

**ステップ 6** **exit**

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-mac)# exit
```

現在のコンフィギュレーション モードを終了します。

**ステップ 7** **interface** *type interface-path-id.subinterface*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# interface GigabitEthernet0/5/0/0.20
```

一致する VLAN ID および EtherType をインターフェイスに割り当てます。

**ステップ 8** **interface** *type interface-path-id.subinterface*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-ac)# interface GigabitEthernet0/5/0/1.15
```

ブリッジ ドメインにインターフェイスを追加し、パケットの転送と、同じブリッジ ドメイン内の他のインターフェイスからのパケットの受信を可能にします。これで、このインターフェイスは、このブリッジ ドメイン上の接続回線になります。

**ステップ 9** **neighbor** { *A.B.C.D* } **pw-id** *value*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# neighbor 10.1.1.2 pw-id 1000
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-pw)#
```

アクセス疑似回線ポートをブリッジ ドメインに追加します。

- 相互接続ピアの IP アドレスを指定するには、*A.B.C.D* 引数を使用します。

  （注） *A.B.C.D* は再帰的または非再帰的プレフィックスです。

- 疑似回線 ID および ID 値を設定するには、**pw-id** キーワードを使用します。指定できる範囲は 1 ～ 4294967295 です。

**ステップ 10** **exit**

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-pw)# exit
```

現在のコンフィギュレーション モードを終了します。

**ステップ 11** **pbb edge i-sid** *service-id* **core-bridge** *core-bridge-name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# pbb edge i-sid 1000 core-bridge pbb-core
```

サービス ID および割り当てられたコア ブリッジ ドメインを指定して、ブリッジ ドメインを PBB エッジとして設定し、PBB エッジ コンフィギュレーション サブモードを開始します。

このブリッジ ドメインのすべてのインターフェイス（ブリッジ ポート）は、カスタマー ネットワーク ポート（CNP）として扱われます。

**ステップ 12** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## B-Component のコア疑似回線の設定

PBB エッジ ブリッジ ドメイン下でスタティック MAC アドレスを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *domain-name*
5. **vfi** { *vfi-name* }
6. **neighbor** { *A.B.C.D* } { **pw-id** *value* }
7. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例:

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *bridge-group-name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn)#bridge group pbb
```

名前付きブリッジ グループのコンフィギュレーション モードを開始します。このコマンドは、新しいブリッジ グループを作成するか、既存のブリッジ グループを変更します（ブリッジ グループが存在する場合）。ブリッジ グループは、ブリッジ ドメインを整理します。

**ステップ 4** **bridge-domain** *domain-name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#bridge-domain pbb-core
```

名前付きブリッジ ドメインのコンフィギュレーション モードを開始します。このコマンドは、新しいブリッジ ドメインを作成するか、既存のブリッジ ドメインを変更します（ブリッジ ドメインが存在する場合）。

**ステップ 5** **vfi** { *vfi-name* }

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# vfi PBB-core-vfi
```

仮想転送インターフェイス（VFI）パラメータを設定し、L2VPN ブリッジ グループ ブリッジ ドメイン VFI コンフィギュレーション モードを開始します。

- 指定した仮想転送インターフェイス名を設定するには、*vfi-name* 引数を使用します。

**ステップ 6** **neighbor** { *A.B.C.D* } { **pw-id** *value* }

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# neighbor 10.1.1.2 pw-id 1000
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-pw)#
```

アクセス疑似回線ポートをブリッジ ドメインに追加するか、または疑似回線を仮想転送インターフェイス（VFI）に追加します。

- 相互接続ピアの IP アドレスを指定するには、*A.B.C.D* 引数を使用します。

  （注） *A.B.C.D* は再帰的または非再帰的プレフィックスです。

- 疑似回線 ID および ID 値を設定するには、**pw-id** キーワードを使用します。指定できる範囲は 1 ～ 4294967295 です。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# PBB-EVPN の設定

この項では、次の内容について説明します。

## PBB コアブリッジドメインの設定

PBB コアブリッジドメインを作成し、それに対応する EVPN EVI ID を割り当てるには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *group_name*
4. **bridge-domain** *bridge_domain_name*
5. **pbb core**
6. **evpn evi** *evi_id*
7. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例:

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

L2VPN コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group** *group_name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group group1
```

ブリッジグループ コンフィギュレーション モードを開始します。このコマンドは、新しいブリッジグループを作成します。ブリッジ グループは、ブリッジ ドメインを整理します。

**ステップ 4** **bridge-domain** *bridge_domain_name*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)#bridge-domain sample-pbb-core#
```

ブリッジグループドメイン設定モードを開始します。このコマンドは、新しいブリッジドメインを作成します。

**ステップ 5** **pbb core**

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# pbb core
```

ブリッジ ドメインを PBB コアとして設定し、PBB コア コンフィギュレーション サブモードを開始します。

このコマンドは、カスタマーブリッジポート（CBP）と呼ばれる内部ポートを作成します。このブリッジドメインのすべてのインターフェイス（ブリッジポート）は、プロバイダーネットワーク ポート（PNP）として扱われます。

**ステップ 6** **evpn evi** *evi_id*

例:

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-pbb-core)# evpn evi 100
```

EVPN 設定モードを開始し、イーサネット VPN ID を設定します。EVI ID の範囲は 1 ～ 65534 です。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## PBB エッジブリッジドメインの設定

前提条件として、PBB-EVPNプロバイダーエッジ（PE）は、片側がアクセスインターフェイスからのトラフィックに一致するイーサネットフローポイントに関連付けられ、もう片側がコアを介したトラフィック転送用のPBBコアブリッジドメインにリンクされた、PBBエッジブリッジドメインを使用して設定する必要があります。

エッジブリッジドメインの設定の詳細については、「PBBエッジブリッジドメインおよびサービス インスタンス ID の設定」を参照してください。

## EVPN イーサネットセグメントの設定

ESIおよびサービスカービングの動作（手動または動的）などのイーサネットセグメントパラメータの明示的な設定は、アクティブ/アクティブ サービス単位ロードバランシングを使用するデュアルホームシナリオの場合のみ必要です。

（注） デフォルトでは、アクティブ/アクティブ フロー単位ロードバランシングを使用するデュアルホームシナリオは、CE の LACP 情報から ESI 値を自動認識します。

（注） PBB-EVPN 設定では、24 個 の ICCP グループのみを作成できます。

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。

### 手順の概要

1. **configure**
2. **evpn**
3. **interface type** *interface-path-id*
4. **ethernet-segment**
5. **backbone-source-mac** *mac_address*
6. **force single-homed**
7. **identifier type** *esi-type esi-identifier*
8. **bgp route-target** *ipv4/v6-address*
9. **load-balancing-mode per-service**
10. **service-carving manual primary** {*isid*} **secondary** {*isid*}
11. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **evpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# evpn
```

EVPN 設定モードを開始します。

**ステップ 3** **interface type** *interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-evpn)# interface gigabitEthernet 0/1/0/4
```

物理ポートインターフェイスまたはバンドルインターフェイスの設定モードを開始します。

**ステップ 4** **ethernet-segment**

例：

```
RP/0/RSP0/cpu 0: router(config-evpn-ac)# ethernet-segment
```

EVPN イーサネットセグメント設定モードを開始します。

**ステップ 5** **backbone-source-mac** *mac_address*

例：

```
RP/0/RSP0/cpu 0: router(config-evpn-ac-es)# backbone-source-mac 0045.1200.04
```

バックボーン送信元 MAC アドレスを設定します。

**ステップ 6** **force single-homed**

例：

```
RP/0/RSP0/cpu 0: router(config-evpn-ac-es)# force single-homed
```

このイーサネットセグメントの強制属性を指定します。

**ステップ 7** **identifier type** *esi-type esi-identifier*

例：

```
RP/0/RSP0/cpu 0: router(config-evpn-ac-es)# identifier type 0 ce.01.ce.01.ce.01.ce.01.01
```

インターフェイスのイーサネットセグメント識別子（ESI）を設定します。

**ステップ 8** **bgp route-target** *ipv4/v6-address*

**例**：

```
RP/0/RSP0/cpu 0: router(config-evpn-ac-es)# bgp route-target ce01.ce01.ce01
```

イーサネットセグメントの BGP インポートルートターゲットを設定します。

（注） このコマンドの設定は、ESI タイプ 0 に必須です。

**ステップ 9** **load-balancing-mode per-service**

**例**：

```
RP/0/RSP0/cpu 0: router(config-evpn-ac-es)# load-balancing-mode per-service
```

ロードバランシングモードを指定します。

**ステップ 10** **service-carving manual primary** {*isid*} **secondary** {*isid*}

**例**：

```
RP/0/RSP0/cpu 0: router(config-evpn-ac-es)# service-carving manual primary 100 secondary 200
```

サービス識別子（isid）のリストをアクティブおよびスタンバイサービスとして指定します。**isid** の範囲は 256 ～ 16777216 です。

**ステップ 11** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## BGP ルートターゲットの設定

デフォルトでは、次のパラメータが PE の設定から自動生成されます。

- グローバル イーサネット セグメント テーブルのルート識別（RD）

デフォルト：ループバック IP アドレスに基づく自動生成 RD

- EVI の BGP ルート識別子（RD）

デフォルト：ループバック IP アドレスに基づく自動生成 RD

- EVI の BGP ルート ターゲット。デフォルト：EVI ID に基づく自動生成 RT

自動生成された BGP RD/RT 値を上書きするには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **evpn**
3. **bgp**
4. **rd** { *2-byte as_number* | *4-byte as_number* | *IP_address* | **none** } : { *nn* }
5. **exit**
6. **evpn**
7. **evi** *evi_id*
8. **bgp**
9. **route-target** [ **import** | **export** ] { *2-byte as_number* | *4-byte as_number* | *IP_address* | **none** } : { *nn* }
10. **rd** { *2-byte as_number* | *4-byte as_number* | *IP_address* | **none** } : { *nn* }
11. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **evpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# evpn
```

EVPN 設定モードを開始します。

**ステップ 3** **bgp**

例：

```
RP/0/RSP0/cpu 0: router(config-evpn)# bgp
```

EVPN BGP 設定モードを開始し、イーサネットセグメント ES:GLOBAL EVI（ES ルートの処理に使用）のスタティック BGP 設定を行います。

**ステップ 4** **rd** { *2-byte as_number* | *4-byte as_number* | *IP_address* | **none** } : { *nn* }

例：

```
RP/0/RSP0/cpu 0: router(config-evpn-bgp)# rd 200:50
```

ルート識別子を設定します。

**ステップ 5** **exit**

**例**：

```
RP/0/RSP0/cpu 0: router(config-evpn-bgp)# exit
```

グローバル コンフィギュレーション モードに戻ります。

**ステップ 6** **evpn**

**例**：

```
RP/0/RSP0/cpu 0: router(config)# evpn
```

EVPN 設定モードを開始します。

**ステップ 7** **evi** *evi_id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-evpn)# evi 100
```

イーサネット VPN ID を設定します。

EVI ID の範囲は 1 ～ 65534 です。

**ステップ 8** **bgp**

**例**：

```
RP/0/RSP0/cpu 0: router(config-evpn-evi)# bgp
```

特定の EVI の BGP 設定モードを開始します。

**ステップ 9** **route-target** [ **import** | **export** ] { *2-byte as_number* | *4-byte as_number* | *IP_address* | **none** } : { *nn* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-evpn-evi-bgp)# route-target 10:20
```

ルートターゲット拡張コミュニティを作成します。

- import キーワードを使用すると、ターゲット VPN 拡張コミュニティからルーティング情報がインポートされます。
- export キーワードを使用すると、ルーティング情報がターゲット VPN 拡張コミュニティにエクスポートされます。

**ステップ 10** **rd** { *2-byte as_number* | *4-byte as_number* | *IP_address* | **none** } : { *nn* }

**例**：

```
RP/0/RSP0/cpu 0: router(config-evpn-evi-bgp)# rd 25:30
```

ルート識別子を設定します。

**ステップ 11** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## グローバル EVPN タイマーの設定

グローバル EVPN タイマーを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**
2. **evpn**
3. **timers [flushagain | peering | programming | recovery]**
4. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **evpn**

例：

```
RP/0/RSP0/cpu 0: router evpn
```

EVPN 設定モードを開始します。

**ステップ 3** **timers [flushagain | peering | programming | recovery]**

例：

```
RP/0/RSP0/cpu 0: router(config-evpn)# timers flushagain 40
```

グローバル EVPN タイマーを設定します。

- 再フラッシュタイマー（AApS の場合のみ）：MAC フラッシュが送信された場合（通常はプログラミングタイマーの期限切れの終了時）、再フラッシュタイマーは、再フラッシュタイマー値で開始されます。期限が切れると、再度 MAC フラッシュメッセージ（MVRP または STP-TCN）が CE に送信されます。このタイマーは、セグメントインターフェイスごとに設定できます。

  範囲：0 ～ 120 秒（0 は無効化を意味します）

  デフォルト：60 秒

- ピアリングタイマー：BGPにアドバタイズする条件をすべて満たすと、PEはピアリングタイマー値の期間待機してから、その RT、ESI、およびローカル MAC をアドバタイズします（シングルホームの場合）。

  範囲：0 ～ 300 秒（0 は無効化を意味します）

  デフォルト：45 秒

- プログラミングタイマー：HW でカービング結果を適用するために必要な時間を示します。プログラミングタイマーが期限切れすると、次のイーサネット セグメント ルート オブジェクトが処理されます。

  範囲：0 ～ 100000 マイクロ秒

  デフォルト：1500 マイクロ秒

- リカバリタイマー（AApS の場合のみ）：インターフェイスが起動すると、PE は、STP プロトコルを実行している CE が収束するために、リカバリタイマー値の期間待機します。このタイマーは、セグメントインターフェイスごとに設定できます。

  範囲：20 ～ 3600 秒

  デフォルト： 20 秒

（注） タイマーの変更は、拡張性設定でのみ有効です。

**ステップ 4** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## イーサネットセグメントごとの EVPN タイマーと CE フラッシュメカニズムの設定

イーサネットセグメントごとのタイマーを設定するには、次の作業を実行します。

### 手順の概要

1. **configure**

2. **evpn**
3. **interface type** *interface-path-id*
4. **ethernet-segment**
5. **mac-flush mvrp**
6. **timers [flushagain | recovery]**
7. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **evpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# evpn
```

EVPN 設定モードを開始します。

**ステップ 3** **interface type** *interface-path-id*

例：

```
RP/0/RSP0/cpu 0: router(config-evpn)# interface gigabitEthernet 0/1/0/4
```

物理ポートインターフェイスまたはバンドルインターフェイスの設定モードを開始します。

**ステップ 4** **ethernet-segment**

例：

```
RP/0/RSP0/cpu 0: router(config-evpn-ac)# ethernet-segment
```

EVPN イーサネットセグメント設定モードを開始します。

**ステップ 5** **mac-flush mvrp**

例：

```
RP/0/RSP0/cpu 0: router(config-evpn-ac)#
```

このイーサネットセグメントの MAC フラッシュモードを指定します。

**ステップ 6** **timers [flushagain | recovery]**

例：

```
RP/0/RSP0/cpu 0: router(config-evpn-ac)# timers flushagain 40
```

イーサネットセグメントごとのタイマーを設定します。

- 再フラッシュタイマー（AApS の場合のみ）：MAC フラッシュが送信された場合（通常はプログラミングタイマーの期限切れの終了時）、再フラッシュタイマーは、再フラッシュタイマー値で開始されます。期限が切れると、再度 MAC フラッシュメッセージ（MVRP または STP-TCN）が CE に送信されます。このタイマーは、セグメントインターフェイスごとに設定できます。

  範囲：0 ～ 120 秒（0 は無効化を意味します）

  デフォルト：60 秒

- リカバリタイマー（AApS の場合のみ）：インターフェイスが起動すると、PE は、STP プロトコルを実行している CE が収束するために、リカバリタイマー値の期間待機します。このタイマーは、セグメントインターフェイスごとに設定できます。

  範囲：20 ～ 3600 秒

  デフォルト： 20 秒

（注） タイマーの変更は、拡張性設定でのみ有効です。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## マルチシャーシ リンク集約の設定

マルチシャーシ リンク集約（MCLAG）は、マルチホームデバイスに関連するシナリオで使用されます。**mlacp system mac, mlacp system priority, mlacp node id** などの関連する MLACP パラメータとバックボーン インターフェイスを指定するために、ICCP 冗長グループを作成する必要があります。

（注） 冗長グループは **redundancy-iccp-group** サブモードで作成されますが、このソリューションは同じサイトに接続されている PE 間の実際の ICCP セッションに依存しません。mode singleton コマンドは、ICCP モジュールのアラートを出すために導入されました。

MCLAG の設定の詳細については、『Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Configuration Guide』の「Configuring Link Bundling on the Cisco ASR 9000 Series Router」モジュールを参照してください。

# BGP ルーティングプロセスの設定

PBB-EVPN の前提条件では、BGP ルーティングプロセスおよび BGP ネイバーサブモードで新しい EVPN アドレスファミリを有効にする必要があります。BGP の詳細については、『Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide』の「Implementing BGP」モジュールを参照してください。

BGP ルーティングプロセスと BGP ネイバーサブモードで 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。

## 手順の概要

1. **configure**
2. **router bgp** *asn_id*
3. **address-family l2vpn evp**
4. **exit**
5. **neighbor** *peer_ip_add*
6. **address-family l2vpn evpn**
7. **address-family l2vpn evpn**
8. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **router bgp** *asn_id*

例：

```
RP/0/RSP0/cpu 0: router(config)# router bgp 100
```

BGP AS 番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。

**ステップ 3** **address-family l2vpn evp**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# address-family l2vpn evpn
```

BGP ルーティングプロセスで EVPN アドレスファミリを有効にし、EVPN アドレスファミリ設定サブモードを開始します。

**ステップ 4** **exit**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# exit
```

現在のコンフィギュレーション モードを終了します。

**ステップ 5** **neighbor** *peer_ip_add*

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 10.1.1.1
```

BGP ルーティングのためにルータをネイバー コンフィギュレーション モードにして、ネイバーの IP アドレスを BGP ピアとして設定します。

**ステップ 6** **address-family l2vpn evpn**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# exit
```

BGP ルーティングプロセスで EVPN アドレスファミリを有効にし、EVPN アドレスファミリ設定サブモードを開始します。

**ステップ 7** **address-family l2vpn evpn**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-family l2vpn evpn
```

BGP ルーティングプロセスで EVPN アドレスファミリを有効にし、EVPN アドレスファミリ設定サブモードを開始します。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

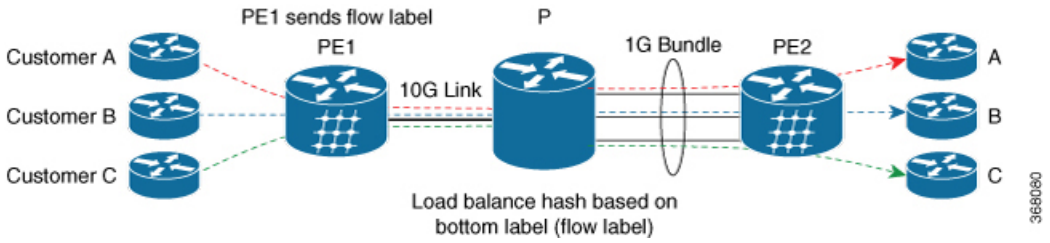
- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# PBB EVPN フローラベル

PBB EVPN のフローラベル機能により、プロバイダー（P）ルータはフローベースのロードバランシングを使用して、プロバイダーエッジ（PE）デバイス間でトラフィックを転送できます。インポジション PE に伝送される個々のパケットフローに基づいてフローラベルが作成され、最低ラベルとしてパケットに挿入されます。P ルータは、フローラベルをロードバランシングに使用し、コア内の ECMP パス全体にわたって、またはリンクがバンドルされたパス全体にわたって、より適切にトラフィックを分配します。フローは、トラフィックの送信元/宛先 IP アドレスとトラフィックのレイヤ 4 送信元/宛先ポートによって識別されるか、またはトラフィックの送信元/宛先 MAC アドレスによって識別されます。

PBB EVPN インポジションルータ（PE1）が EVPN トラフィックにフローラベルを追加する次のトポロジについて考えてみましょう。PBB EVPN ディスポジションルータ（PE2）では、フローラベルを持つトラフィックとフローラベルを持たないトラフィックの混合タイプが許可されます。P ルータはフローラベルを使用して、PE 間でトラフィックのロードバランシングを行います。PE2は、トラフィックのフローラベルを無視し、すべてのユニキャストトラフィックで 1 つの EVPN ラベルを使用します。

**図 *49 : PBB EVPN* フローラベル**




# PBB EVPN フローラベルの設定

PBB EVPN フローラベル機能を設定するには、次の作業を実行します。

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)#bridge group PBB
RP/0/RSP0/CPU0:router(config-l2vpn-bg)#bridge-domain EDGE
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#interface Bundle-Ether1.10
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)#pbb edge i-sid 1010 core-bridge CORE
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac-pbb-edge)#exit
!
RP/0/RSP0/CPU0:router(config-l2vpn-bg)#bridge-domain CORE
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#pbb-core
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pbb-core)#evpn evi 10
!
RP/0/RSP0/CPU0:router(config)#evpn
RP/0/RSP0/CPU0:router(config-evpn)#evi 10
RP/0/RSP0/CPU0:router(config-evpn-instance)#load-balancing
RP/0/RSP0/CPU0:router(config-evpn-instance-lb)#flow-label static
!
```

```
RP/0/RSP0/CPU0:router(config)#router bgp 20
RP/0/RSP0/CPU0:router(config-bgp)#address-family l2vpn evpn
RP/0/RSP0/CPU0:router(config-bgp-af)#commit
```

### 実行コンフィギュレーション

このセクションでは、PBB EVPN フローラベル実行コンフィギュレーションを示します。

```
l2vpn
 bridge group PBB
  bridge-domain EDGE
   interface Bundle-Ether1.10
   pbb edge i-sid 1010 core-bridge CORE

  bridge-domain CORE
   pbb-core
     evpn evi 10

evpn
 evi 10
  load-balancing
   flow-label static

router bgp 20
address-family l2vpn evpn
```

### 確認

PBB EVPN フローラベル設定を確認します。

```
RP/0/RSP0/CPU0:router#show evpn evi vpn-id 10 detail
EVI        Bridge Domain                 Type
---------- ----------------------------- -------
10         EVPN                          PBB
   Unicast Label  : 24001
   Multicast Label: 24002
   Flow Label: Y
   RD Config: none
   RD Auto  : (auto) 1.1.1.1:10
   RT Auto  : 1:10
   Route Targets in Use           Type
   ------------------------------ -------
   1:10                           Import
   1:10                           Export
```

# 802.1ah プロバイダー バックボーン ブリッジを実装するための設定例

ここでは、次の設定例を示します。

## イーサネット フロー ポイントの設定：例

次に、イーサネット フロー ポイントを設定する例を示します。

```
config
interface GigabitEthernet0/0/0/10.100 l2transport
 encapsulation dot1q 100
or
 encapsulation dot1ad 100
or
 encapsulation dot1ad 100 dot1q 101
```

# PBB エッジ ブリッジ ドメインおよびサービス インスタンス ID の設定：例

次に、PBB エッジ ブリッジ ドメインを設定する例を示します。

```
config
l2vpn
 bridge group PBB
  bridge-domain PBB-EDGE
  interface GigabitEthernet0/0/0/38.100
  !
  interface GigabitEthernet0/2/0/30.150
  !
  pbb edge i-sid 1000 core-bridge PBB-CORE
   !
 !
!
```

# PBB コア ブリッジ ドメインの設定：例

次に、PBB コア ブリッジ ドメインを設定する例を示します。

```
config
l2vpn
 bridge group PBB
  bridge-domain PBB-CORE
  interface G0/5/0/10.100
  !
  interface G0/2/0/20.200
  !
   pbb core
   !
 !
!
```

# バックボーン VLAN タグの設定：例

次に、バックボーン VLAN タグを設定する例を示します。

```
config
l2vpn
 bridge group PBB
  bridge-domain PBB-CORE
  interface G0/5/0/10.100
  !
  interface G0/2/0/20.200
  !
   pbb core
    rewrite ingress tag push dot1ad 100 symmetric
```

```
   !
 !
!
```

# バックボーン送信元 MAC アドレスの設定：例

次に、バックボーン送信元 MAC アドレスを設定する例を示します。

```
config
l2vpn
 pbb
  backbone-source-mac 0045.1200.04
 !
!
```

# PBB エッジ ブリッジ ドメイン下でのスタティック マッピングおよび不明ユニキャスト MAC アドレスの設定

次に、PBB エッジ ブリッジ ドメイン下でスタティック マッピングおよび不明ユニキャスト MAC アドレスを設定する例を示します。

```
config
l2vpn
 bridge group PBB
  bridge-domain PBB-EDGE
  interface GigabitEthernet0/0/0/38.100
  !
  interface GigabitEthernet0/2/0/30.150
  !
  pbb edge i-sid 1000 core-bridge PBB-CORE
   static-mac-address 0033.3333.3333 bmac 0044.4444.4444
   unknown-unicast-bmac 0123.8888.8888
   !
 !
!
```

# PBB-VPLS の設定：例

次に、PBB VPLS を設定する例を示します。

### I-Component のアクセス疑似回線の設定

```
l2vpn
  bridge group PBB
    bridge-domain PBB-EDGE
      mac withdraw state-down ------ can be used with MIRP, optional
      interface GigabitEthernet0/0/0/38.100
      interface GigabitEthernet0/2/0/30.150
      neighbor 10.10.10.1 pw-id 1010 ------- configures access PW
      !
      pbb edge i-sid 1200 core-bridge PBB-CORE
   !
 !
!
```

### B-Component のコア疑似回線の設定

```
l2vpn
  bridge group PBB
    bridge-domain PBB-CORE
      interface G0/5/0/10.100
      !
    vfi PBB-CORE-vfi
     neighbor 1.1.1.1 pw-id 1004 ------- configures core PW
     !
!
 !
```

# MIRP Lite の設定：例

MIRP 機能はデフォルトでイネーブルです。ただし、MIRP パケットは、接続回線が機能しない場合、および次のように **mac withdraw state-down** を設定した場合に送信されます。

```
l2vpn
  bridge group PBB
    bridge-domain PBB-EDGE
```

mac withdraw state-down

ただし、mac withdraw state-down を設定しないと、MIRP パケットは接続回線が機能しているときに送信されます。

# PBB-EVPN の設定：例

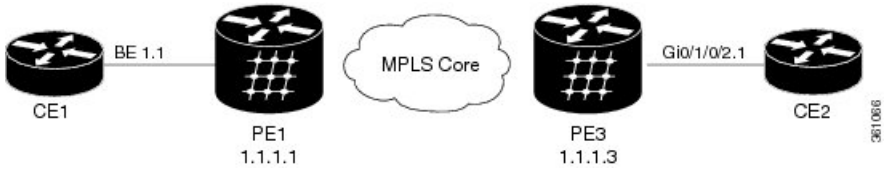
ここでは、次の例を示します。

## シングルホームデバイス/シングルホームネットワークの PBB-EVPN

この例では、次のように設定されます。

- シングルホーム CE を設定した同じ AS 内の 2 つの PE 間 PBB-EVPN サービス
- PE1 に接続されているバンドルインターフェイスを使用したデュアル接続 CE
- PE2 に接続されているシングル接続 CE
- 単一 I-SID のトラフィックを伝送する EVI
- 両方の PE でトラッキングを容易にする設定を使用してカスタマイズした PBB 送信元 MAC
- BGP ASN および EVI ID からの BGP RD/RT 自動生成

図 *50 :* シングルホームデバイス/シングルホームネットワークの *PBB-EVPN*




**Configuration on PE1:**

```
interface Bundle-Ether1.1 l2transport
   encapsulation dot1q 1 200

l2vpn
  pbb
    backbone-source-mac 00aa.00bb.00cc
  bridge group gr1
    bridge-domain bd1
      interface Bundle-Ether1.1
      pbb edge i-sid 300 core-bridge core_bd1

  bridge group gr2
      bridge-domain core_bd1
        pbb core
          evpn evi 1000
!
router bgp 100
 bgp router-id 1.1.1.1
 address-family l2vpn evpn
 !
 neighbor 1.1.1.3
  remote-as 100
  address-family l2vpn evpn
```

**Configuration on PE3:**

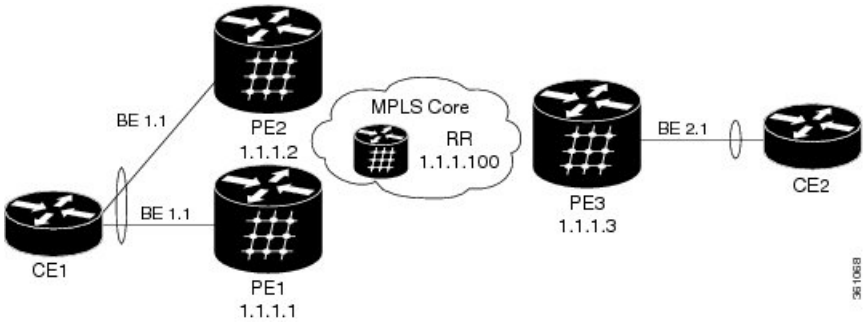
```
interface GigabitEthernet 0/1/0/2.1 l2transport
   encapsulation dot1q 200

l2vpn
  pbb
    backbone-source-mac 00bb.00cc.00dd
  bridge group gr1
    bridge-domain bd1
      interface GigabitEthernet0/1/0/2.1
      pbb edge i-sid 300 core-bridge core_b1

  bridge group gr2
    bridge-domain core_bd1
      pbb core
        evpn evi 1000
!
router bgp 100
 bgp router-id 1.1.1.3
 address-family l2vpn evpn
 !
 neighbor 1.1.1.1
  remote-as 100
  address-family l2vpn evpn
```

# アクティブ/アクティブフロー単位ロードバランシングを設定したデュアルホームデバイス/マルチホームデバイスの PBB EVPN

この例では、次のように設定されます。

- デュアルホーム CE（PE1 と PE2 の背後）とシングルホーム CE（PE3 の背後）を設定した同じ AS 内にある 3 つの PE 間の PBB-EVPN サービス
- 同じ I-SID の入力トラフィックが両方の PE で処理できる、アクティブ/アクティブ フロー単位ロードバランシングを実行するように設定した PE1 と PE2
- 単一 I-SID のトラフィックを伝送する EVI の例を示します
- PBB I-SID 値は、共通のデュアルホームサイトに接続されている PE 間で一致している必要があります
- ICCP は、新しいモード（モードシングルトン）を使用して PE1 と PE2 で設定する必要があります。ICCP ネイバーは設定しません。システム MAC/プライオリティなどの MLACP パラメータは同一である必要がありますが、MLACP ノード ID は PE1/PE2 で一意である必要があることに注意してください
- ESI は、共通のデュアルホームサイトに接続されている PE で同一である必要があります。ESI 値が CE の LACP 情報から自動生成されるデフォルトの動作の例を示します
- PBB 送信元 MAC は、アクティブ/アクティブ フロー単位ロードバランシングで動作しているデュアルホームサイトに接続されている PE で同じである必要があります。PBB 送信元 MAC 値が CE の LACP 情報から自動生成されるデフォルトの動作の例を示します
- CE は、両方の PE に接続するすべてのメンバーインターフェイスを含む 1 つのバンドルインターフェイスを使用して設定する必要があります
- BGP ASN および EVI ID からの BGP RD/RT 自動生成

**図 *51*:アクティブ/アクティブ フロー単位ロードバランシングを設定したデュアルホームデバイス/マルチホームデバイスの *PBB EVPN***




**Configuration on PE1:**

```
redundancy
 iccp
```

```
  group 1
   mlacp node 1
   mlacp system mac 0aaa.0bbb.0ccc
   mlacp system priority 1
   backbone interface GigabitEthernet0/1/0/2
   mode singleton

interface bundle-Ether1
mlacp iccp-group 1

interface bundle-Ether1.1 l2transport
  encapsulation dot1q  10
l2vpn
  bridge group gr1
    bridge-domain bd1
      interface bundle-ether1.1
      pbb edge i-sid 600 core-bridge core_bd1

    bridge group gr2
      bridge-domain core_bd1
        pbb core
          evpn evi 1000
!
router bgp 100
 bgp router-id 1.1.1.1
 address-family l2vpn evpn
 !
 neighbor 1.1.1.100
  remote-as 100
  address-family l2vpn evpn
```

**Configuration on PE2:**

```
redundancy
 iccp
  group 1
   mlacp node 2
   mlacp system mac 0aaa.0bbb.0ccc
   mlacp system priority 1
   backbone interface GigabitEthernet0/1/0/2
   mode singleton

interface bundle-Ether1
mlacp iccp-group 1

interface bundle-Ether1.1 l2transport
  encapsulation dot1q  10
l2vpn
  bridge group gr1
    bridge-domain bd1
      interface bundle-Ether1.1
      pbb edge i-sid 600 core-bridge core_b1

    bridge group gr2
      bridge-domain core_bd1
        pbb core
            evpn evi 1000
!
router bgp 100
 bgp router-id 1.1.1.2
 address-family l2vpn evpn
 !
 neighbor 1.1.1.100
```

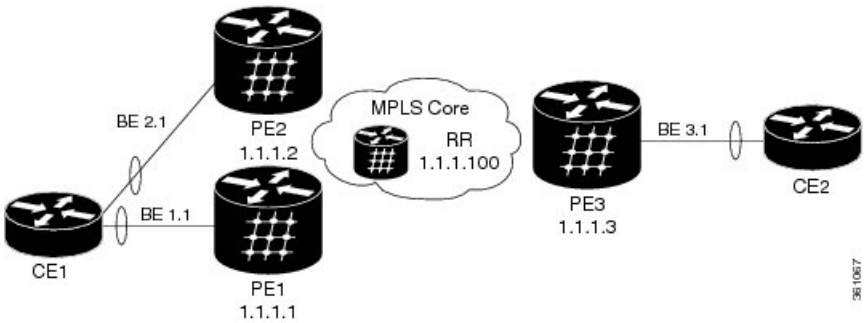
```
remote-as 100
address-family l2vpn evpn
```

# アクティブ/アクティブサービス単位ロードバランシングとダイナミックサービスカービングを設定したデュアルホームデバイス/マルチホームデバイスの PBB EVPN

この例では、次のように設定されます。

- デュアルホーム CE（PE1 と PE2 の背後）とシングルホーム CE（PE3 の背後）を設定した同じ AS 内にある 3 つの PE 間の PBB-EVPN サービス
- 一部の I-SID のトラフィックが PE1 で処理され、その他が PE2 で処理される、ダイナミック サービス カービング/DF 選択を使用してアクティブ/アクティブ サービス単位（I-SID 単位）ロードバランシングを実行するように設定した PE1 と PE2
- 2 つの I-SID のトラフィックを伝送する EVI
- PBB I-SID 値は、共通のデュアルホームサイトに接続されている PE 間で一致している必要があります
- ICCP は、新しいモード（モードシングルトン）を使用して PE1 と PE2 で設定する必要があります。ICCP ネイバーは設定しません。コア分離の失敗を処理するために ICCP 設定が必要です。この例では、PE1/PE2 で同じ MLACP システムの MAC/プライオリティと固有の MLACP ノード値を使用します
- ESI は、デュアルホームサイトの PE 間で同一である必要があります。これを保証するには、ユーザ設定を入力する必要があります
- PBB の送信元 MAC は、デュアルホームサイトに接続された各 PE で異なる必要があります。デフォルトでは、PE はシステム全体の PBB 送信元 MAC を使用します
- CE は 2 つのバンドルインターフェイスを使用して設定する必要があります。個々の PE につながるメンバーインターフェイスのセットごとに 1 つ設定します
- BGP ASN および EVI ID からの BGP RD/RT 自動生成

**図 *52:* アクティブ/アクティブ サービス単位ロードバランシングとダイナミック サービス カービングを設定したデュアルホームデバイス/マルチホームデバイスの *PBB EVPN***



**Configuration on PE1:**

```
redundancy
 iccp
  group 66
   mlacp node 1
   mlacp system mac 0aaa.0bbb.0ccc
   mlacp system priority 1
   backbone interface GigabitEthernet0/1/0/2
   mode singleton

interface Bundle-Ether1
 mlacp iccp-group 66
interface bundle-Ether1.1 l2transport
   encapsulation dot1q 10

interface bundle-Ether1.20 l2transport
   encapsulation dot1q 20

evpn
  interface bundle-Ether1
    ethernet-segment
       identifier type 0 01.11.00.00.00.00.00.00.01
      load-balancing-mode per-service

l2vpn
  bridge group gr1
    bridge-domain bd1
      interface bundle-ether1.1
      pbb edge i-sid 300 core-bridge core_bd1

    bridge-domain bd20
      interface bundle-ether1.20
      pbb edge i-sid 320 core-bridge core_bd1

   bridge group gr2
     bridge-domain core_bd1
            pbb core
                evpn evi 1000
!
router bgp 100
 bgp router-id 1.1.1.1
 address-family l2vpn evpn
 !
 neighbor 1.1.1.100
  remote-as 100
  address-family l2vpn evpn
```

**Configuration on PE2:**

```
redundancy
 iccp
  group 66
   mlacp node 2
   mlacp system mac 0aaa.0bbb.0ccc
   mlacp system priority 1
   backbone interface GigabitEthernet0/1/0/2
   mode singleton

interface Bundle-Ether2
 mlacp iccp-group 66

interface bundle-Ether2.1 l2transport
```

```
   encapsulation dot1q 10


interface bundle-Ether2.20 l2transport
   encapsulation dot1q 20


evpn
  interface bundle-Ether2
    ethernet-segment
      identifier type 0 01.11.00.00.00.00.00.00.01
      load-balancing-mode per-service
l2vpn
  bridge group gr1
    bridge-domain bd1
      interface bundle-Ether2.1
      pbb edge i-sid 300 core-bridge core_bd1

    bridge-domain bd20
      interface bundle-Ether2.20
      pbb edge i-sid 320 core-bridge core_bd1

  bridge group gr2
         bridge-domain core_bd1
              pbb core
                   evpn evi 1000
!
router bgp 100
 bgp router-id 1.1.1.2
 address-family l2vpn evpn
 !
 neighbor 1.1.1.100
  remote-as 100
  address-family l2vpn evpn
```

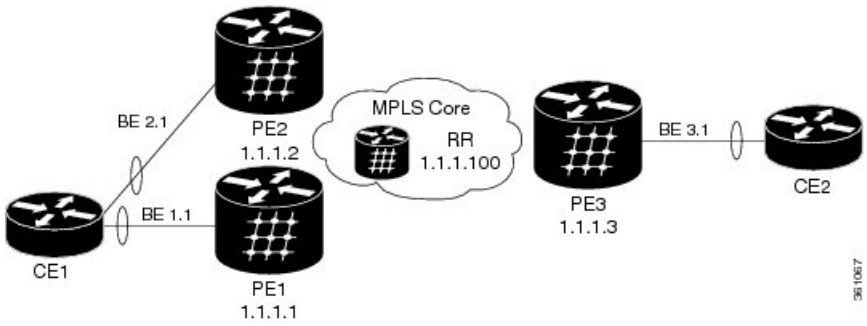
# アクティブ/アクティブ サービス単位ロードバランシングと手動サービスカービングを設定したデュアルホームデバイス/マルチホームデバイスの PBB EVPN

この例では、次のように設定されます。

- デュアルホーム CE（PE1 と PE2 の背後）とシングルホーム CE（PE3 の背後）を設定した同じ AS 内にある 3 つの PE 間の PBB-EVPN サービス
- 手動サービスカービング/DF 選択を使用してアクティブ/アクティブ サービス単位（I-SID 単位）ロードバランシングを実行するように設定した PE1 と PE2
- PE1 は、I-SID 範囲 256 ～ 276 と I-SID のバックアップ 277 ～ 286 からトラフィックを転送するように設定します。PE2 は、PE1 とは逆の動作をするように設定します
- 2 つの I-SID のトラフィックを伝送する EVI
- PBB I-SID 値は、共通のデュアルホームサイトに接続されている PE 間で一致している必要があります

- ICCP は、新しいモード（モードシングルトン）を使用して PE1 と PE2 で設定する必要があります。ICCP ネイバーは設定しません。コア分離の失敗を処理するために ICCP 設定が必要です。この例では、PE1/PE2 で同じ MLACP システムの MAC/プライオリティと固有の MLACP ノード値を使用します
- ESI は、デュアルホームサイトの PE 間で同一である必要があります。これを保証するには、ユーザ設定を入力する必要があります
- PBB の送信元 MAC は、デュアルホームサイトに接続された各 PE で異なる必要があります。この例では、トラッキングを容易にする設定を使用して PBB 送信元 MAC 値をカスタマイズします
- CE は 2 つのバンドルインターフェイスを使用して設定する必要があります。個々の PE につながるメンバーインターフェイスのセットごとに 1 つ設定します
- BGP ASN および EVI ID からの BGP RD/RT 自動生成

**図 *53*：アクティブ/アクティブ サービス単位ロードバランシングと手動サービスカービングを設定したデュアルホームデバイス/マルチホームデバイスの *PBB EVPN***




**Configuration on PE1:**

```
redundancy
 iccp
  group 66
   mlacp node 1
   mlacp system mac 0aaa.0bbb.0ccc
   mlacp system priority 1
   backbone interface GigabitEthernet0/1/0/2
   mode singleton

interface Bundle-Ether1
 mlacp iccp-group 66

interface bundle-Ether1.1 l2transport
  encapsulation dot1q  10

interface bundle-Ether1.20 l2transport
  encapsulation dot1q 20

evpn
  interface bundle-Ether1
    ethernet-segment
      identifier type 0 01.11.00.00.00.00.00.00.01
      load-balancing-mode per-service
```

```
      service-carving manual primary isid 256-276 secondary isid 277-286
l2vpn
  pbb
    backbone-source-mac 00aa.00bb.00cc
  bridge group gr1
    bridge-domain bd_256
      interface bundle-ether1.1
      pbb edge i-sid 260 core-bridge core_bd1

    bridge-domain bd_286
      interface bundle-ether1.20
      pbb edge i-sid 280 core-bridge core_bd1
  bridge group gr2
    bridge-domain core_bd1
      pbb core
        evpn evi 1000
!
router bgp 100
 bgp router-id 1.1.1.1
 address-family l2vpn evpn
 !
 neighbor 1.1.1.100
  remote-as 100
  address-family l2vpn evpn
```

**Configuration on PE2:**

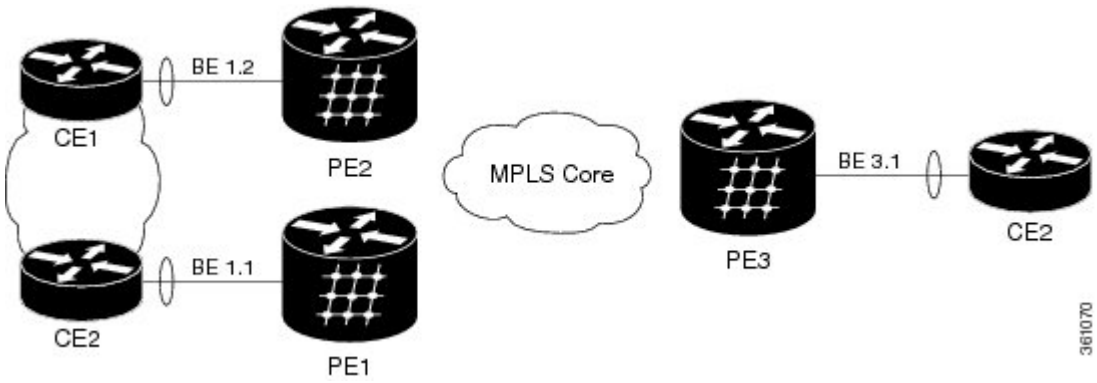
```
redundancy
 iccp
  group 66
   mlacp node 2
   mlacp system mac 0aaa.0bbb.0ccc
   mlacp system priority 1
   backbone interface GigabitEthernet0/1/0/2
   mode singleton

interface Bundle-Ether2
 mlacp iccp-group 66

interface bundle-Ether2.1 l2transport
  encapsulation dot1q  10

interface bundle-Ether2.20 l2transport
  encapsulation dot1q 20


evpn
  interface bundle-Ether2
    ethernet-segment
      identifier type 0 01.11.00.00.00.00.00.00.01
        load-balancing-mode per-service
        service-carving manual primary 277-286 secondary 256-276

l2vpn
  pbb
    backbone-source-mac 00cc.00dd.00ee
  bridge group gr1
    bridge-domain bd1
      interface bundle-Ether2.1
      pbb edge i-sid 260 core-bridge core_b1

    bridge-domain bd30

      Interface bundle-Ether2.20
```

```
      pbb edge i-sid 280 core-bridge core_b1
  bridge group gr2
    bridge-domain core_bd1
      pbb core
        evpn evi 1000
!
router bgp 100
 bgp router-id 1.1.1.2
 address-family l2vpn evpn
 !
 neighbor 1.1.1.100
  remote-as 100
  address-family l2vpn evpn
```

# PBB-EVPN マルチホームネットワーク

次に、アクティブ-アクティブ サービス単位ロードバランシングを設定したマルチホームネットワークに PBB-EVPN を設定する例を示します。

**図 *54 : PBB-EVPN* マルチホームネットワーク**

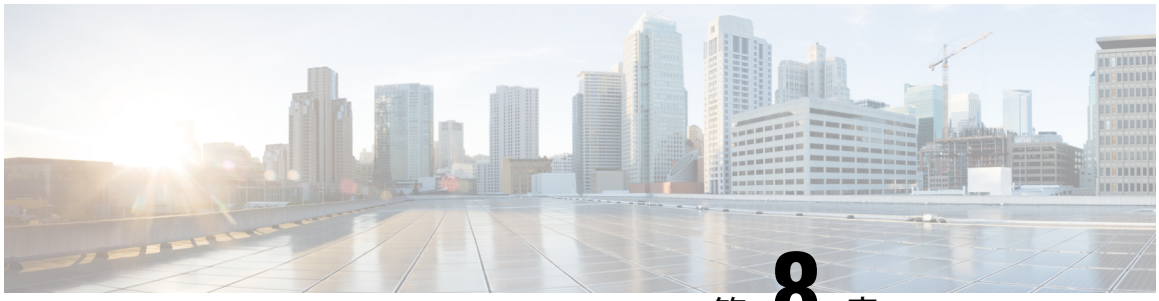



**Configuration on PE1:**

```
interface bundle-Ether1.1 l2transport
  encapsulation dot1q  1

evpn
  interface bundle-Ether1
    ethernet-segment
        load-balancing-mode per-service
l2vpn
  pbb
    backbone-source-mac 00aa.00bb.00cc
  bridge group gr1
    bridge-domain bd1
      interface bundle-ether1.1
      pbb edge i-sid 400 core-bridge core_bd1

    bridge group gr2
      bridge-domain core_bd1
        pbb core
          evpn evi 1000
```

**Configuration on PE2:**

```
interface bundle-Ether1.1 l2transport
  encapsulation dot1q  1
evpn
  interface bundle-Ether1
    ethernet-segment
      load-balancing-mode per-service

l2vpn
  pbb
    backbone-source-mac 00cc.00dd.00ee
  bridge group gr1
    bridge-domain bd1
      interface bundle-Ether1.1
      pbb edge i-sid 400 core-bridge core_bd1

  bridge group gr2
    bridge-domain core_bd1
      pbb core
        evpn evi 1000
```



第 8 章

# マルチ スパニングツリー プロトコルの実装

このモジュールでは、Cisco ASR 9000 シリーズ ルータでのマルチ スパニングツリー プロトコルの概念および設定情報について説明します。マルチスパニングツリープロトコル（MSTP）は、ブリッジ設定のループを防ぐために使用されるスパニングツリープロトコルです。他のタイプの STP とは異なり、MSTP は VLAN ごとにポートを選択的にブロックできます。

| リリース | 変更内容 |
|---|---|
| リリース 3.7.3 | この機能が、Cisco ASR 9000 シリーズ ルータで導入されました。 |
| リリース 3.9.1 | バンドル機能での MSTP のサポートが追加されました。 |
| リリース 4.0.1 | PVST+ および PVSTAG 機能のサポートが追加されました。 |
| リリース 4.1.0 | MSTAG エッジ モード機能のサポートが追加されました。 |
| リリース 4.3.0 | バンドル インターフェイスに PVSTAG のサポートが追加されました。 |
| リリース 5.1.0 | PVRST のサポートが追加されました。 |

# マルチスパニングツリープロトコルを実装するための前提条件

この前提条件は、MSTP の実装に適用されます。

適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。

ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

# マルチスパニングツリープロトコルの実装に関する情報

イーサネット サービス アクセス リストを実装するには、次の概念を理解している必要があります。

## スパニングツリー プロトコルの概要

イーサネットは、ネットワークの手段とホストの相互接続に使用される、単なるリンク層テクノロジーではありません。シンプルなプラグ アンド プレイ プロビジョニングの考え方と統合されている低コストで幅広い帯域幅機能によって、特にサービス プロバイダー ネットワークのアクセスおよび集約の領域で、イーサネットはネットワークを構築するための正規の技法になっています。

レイヤ 2（L2）ヘッダーの TTL フィールドがなく、マルチキャスト トラフィック ネットワーク全体が推奨されるか必要とされるイーサネット ネットワークは、ループが発生する場合にブロードキャスト ストームの影響を受けやすくなります。ただし、ループは、冗長パスを提供するため、望ましい特性です。スパニングツリー プロトコル（STP）は、イーサネット ネットワーク内のループ フリー トポロジを提供するために使用され、リンク障害に対処するようにネットワーク内の冗長性を確保できます。

STP には、多くのバリアントがあります。ただし、同じ基本原則で動作します。ループを含む可能性があるネットワーク内では、ループ フリーのスパニングツリーを確保できるように（つまり、ネットワーク内の任意の 2 台のデバイス間に 1 つだけパスが存在するように）、十分な数のインターフェイスが STP によってディセーブルになります。アクティブ リンクの 1 つに影響を与えるネットワークに障害がある場合、プロトコルは、すべてのデバイスが引き続き到達可能になるように、スパニングツリーを再計算します。STP は、単一の LAN セグメントに接続されているか、複数のセグメントが含まれてループがないことを確認するために STP を使用するスイッチド LAN に接続されているかを検出できないエンド ステーションに対してトランスペアレントです。

## STP プロトコルの動作

STP のすべてのバリアントは同じ方法で動作します。STP フレーム（ブリッジプロトコルデータユニット（BPDU）とも呼ばれます）は、STP に参加しているネットワーク デバイス間でレイヤ 2 LAN セグメントを介して定期的に交換されます。このようなネットワーク デバイスはこれらのフレームを転送しませんが、ループ フリー スパニングツリーを構築するために情報を使用します。

スパニングツリーは、最初にスパニングツリーのルート（ルートブリッジと呼ばれます）であるデバイスを選択してから、そのルートブリッジからネットワーク内のその他すべてのデバイスへのループフリーパスを判別することで構成されます。冗長パスは、適切なポートをブロック状態に設定することで無効にされます。ブロック状態では、STP フレームを引き続き交換できますが、データ トラフィックは転送されません。ネットワーク セグメントで障害が発生し、冗長パスが存在する場合、STP プロトコルがスパニングツリー トポロジを再計算し、適切なポートのブロックを解除することによって、冗長パスをアクティブにします。

STP ネットワーク内のルートブリッジは、各デバイスの設定されたブリッジプライオリティおよび組み込み MAC アドレスの組み合わせであるブリッジ ID の最も小さなものが選択されます。プライオリティが最低であるか、または等しく最低のプライオリティであるが最小 MAC アドレスを持つデバイスが、ルートブリッジとして選択されます。

一連の冗長パス内でのアクティブ パスの選択は、主にポート パス コストによって決定されます。ポート パス コストは、そのポートとルート ブリッジ間の転送コストを表します。ポートがルートブリッジから遠いほど、コストは高くなります。コストは、（デフォルトで）メディア速度に依存する量だけ、パスのリンクごとに増加します。指定された LAN セグメントからの 2 つのパスのコストが等しい場合、接続先デバイスのブリッジ ID が最小のものが選択されます。また、2 つの接続が同じデバイスに対するものである場合は、接続されたネイバーポートの設定されたポートプライオリティとポート ID で決まります。

アクティブなパスを選択すると、アクティブ トポロジの一部にならないポートはすべてブロッキング状態に移行します。

## トポロジの変更

スイッチド LAN のネットワーク デバイスは、MAC 学習を実行します。つまり、受信したデータ トラフィックを使用して、その MAC アドレス宛のフレームの送信先となるインターフェイスとユニキャスト MAC アドレスを関連付けます。STP を使用すると、スパニングツリーの再計算（たとえば、ネットワーク障害後）によって、この学習した情報を無効にできます。したがってプロトコルには、古い情報を削除（フラッシュ）して、新しいトポロジに基づいた新しい情報を学習できるように、ネットワーク全体でのトポロジ変更を通知するメカニズムが含まれます。

トポロジ変更通知は、STP がポートをブロッキング状態から転送状態に移行するたびに送信されます。これを受信すると、受信デバイスは、通知を受け取ったポート以外のブロックされないすべてのポートで MAC 学習エントリをフラッシュして、さらにこれらのポートから独自のトポロジ変更通知を送信します。このように、古い情報がネットワーク内のすべてのデバイスから削除されるようにします。

## STP のバリアント

スパニングツリー プロトコルには、多くのバリアントがあります。

- レガシー STP（STP）：元の STP プロトコルは、IEEE 802.1D-1998 で定義されていました。これはすべての VLAN で使用する単一のスパニングツリーを作成し、コンバージェンスのほとんどはタイマーベースです。
- 高速 STP（RSTP）：これは、イベントベースであるためにより高速なコンバージェンスを提供するために IEEE 802.1D-2004 で定義された機能拡張です。ただし、引き続きすべての VLAN で単一のスパニングツリーを作成します。
- マルチ STP（MSTP）：さらなる拡張機能が IEEE 802.1Q-2005 で定義されました。これにより、複数のスパニング ツリー インスタンスが同じ物理トポロジで作成できます。個々の VLAN を異なるスパニング ツリー インスタンスに割り当てることによって、データトラフィックは各物理リンクに負荷分散できます。作成できるスパニング ツリー インスタンスの数は、使用可能な VLAN の数よりもさらに小さい値に制限されますが、複数の VLAN を同じスパニング ツリー インスタンスに割り当てることができます。MSTP 情報の交換に使用される BPDU は常にタグなしで送信されます。VLAN およびスパニングツリー インスタンス データは BPDU 内で符号化されます。
- Per-VLAN STP（PVST）：これは、マルチ スパニングツリーを作成するための代替メカニズムです。MSTP の標準化の前にシスコが開発しました。PVST を使用して、別個のスパニングツリーが VLAN ごとに作成されます。PVST+（レガシー STP に基づく）および PVRST（RSTP に基づく）の 2 つのバリアントがあります。パケット レベルのスパニングツリーの分離は、適切な VLAN タグでタグ付けされた標準の STP または RSTP BPDU を送信して行われます。
- Per-VLAN Rapid Spanning Tree（PVRST）は、VLAN ごとに実装されている IEEE 802.1w（RSTP）規格で、Rapid PVST や PVST+ と呼ぶこともあります。（手作業で STP をディセーブルにしていない場合、）STP の 1 つのインスタンスは、設定されている各 VLAN で実行されます。VLAN 上の各 Rapid PVST+ インスタンスには、1 つのルート スイッチがあります。Rapid PVST+ の実行中には、VLAN ベースで STP をイネーブルまたはディセーブルにできます。
- PVST では、ポイントツーポイントの配線を使用して、スパニングツリーの高速収束が行われます。PVRST によりスパニングツリーの再設定を 1 秒未満に発生させることができます（802.1D STP のデフォルト設定では 50 秒）。
- REP（シスコ独自のリング冗長プロトコル）：これは、リングで復元力を提供するためのシスコ独自のプロトコルです。これは、MSTP ピアとの相互運用を行うために使用する MSTP 互換モードが提供されるため、完全を期すために組み込まれています。

# マルチ スパニングツリー プロトコルの概要

マルチ スパニングツリープロトコル（MSTP）は、複数および独立したスパニングツリーを同じ物理ネットワークに作成できるようにする STP バリアントです。各スパニングツリーのパラメータは、ループフリー トポロジを形成するために、ルート ブリッジとして別のネットワーク デバイスを選択するか、別のパスを選択するように、別個に設定できます。その結果、特定の物理インターフェイスを一部のスパニングツリーではブロックして、その他のツリーではブロック解除できます。

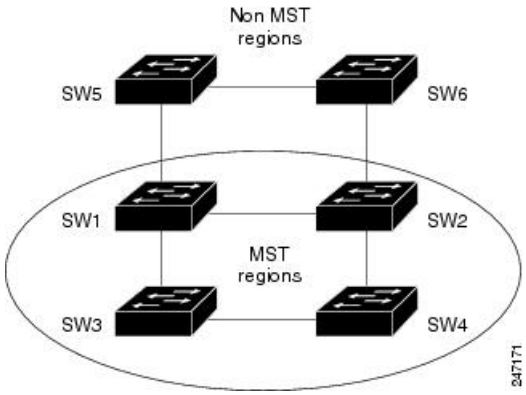
マルチ スパニングツリー インスタンスを設定すると、使用中の VLAN セットをツリー間で分割できます。たとえば、VLAN 1 ～ 100 をスパニングツリー インスタンス 1 に、VLAN 101 ～ 200 をスパニングツリー インスタンス 2 に、VLAN 201 ～ 300 をスパニングツリー インスタンス 3 に割り当てるなどができます。各スパニングツリーには、異なるアクティブリンクとの別のアクティブトポロジがあるため、VLAN に基づいて、利用可能な冗長リンク間でデータトラフィックを分割できます（ロードバランシングの実行）。

## MSTP リージョン

マルチ スパニングツリーのサポートとともに、MSTP では、リージョンの概念が採用されています。リージョンは、同じ管理制御下にあるデバイス グループであり、類似した設定があります。特に、リージョン名の設定、リビジョン、スパニングツリー インスタンスへの VLAN のマッピングは、リージョン内のすべてのネットワーク デバイスで同じでなければなりません。同じリージョン内にあるかどうかを他のデバイスが確認できるように、この情報のダイジェストが、各デバイスによって送信される BPDU に含まれています。

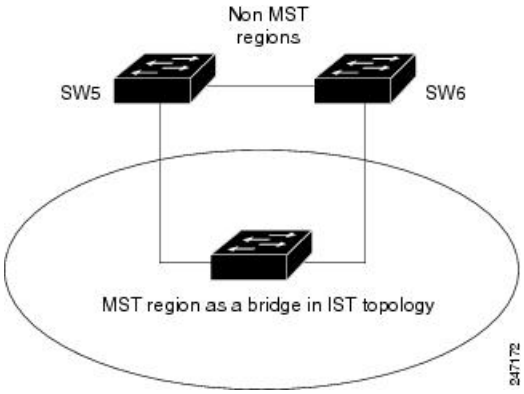
次の図に、MSTP を実行するブリッジがレガシー STP または RSTP を実行するブリッジに接続されている場合の MST リージョンの動作を示します。この例では、スイッチ SW1、SW2、SW3、SW4 では MSTP がサポートされるのに対して、スイッチ SW5 および SW6 ではサポートされません。

**図 *55 :* 非 *MST* リージョンとの *MST* の対話**




この状況に対処するために、Internal Spanning Tree（IST）が使用されます。これは、常にスパニングツリー インスタンス 0（ゼロ）です。MSTP 非認識デバイスと通信する場合、全体の MSTP リージョンは単一のスイッチとして表されます。次の図に、この場合の論理 IST トポロジを示します。

図 *56 :* 非 *MST* ブリッジと対話する *MST* リージョンの論理トポロジ




同じメカニズムが、別のリージョンにあるMSTPデバイスとの通信時に使用されます。たとえば、上の図の SW5 は、すべてが SW1、SW2、SW3、SW4 とは別のリージョンにある多数の MSTP デバイスを表している可能性があります。

## MSTP Port Fast

MSTPには、スイッチドイーサネットネットワークのエッジでポートを処理するためのPortFast機能が組み込まれています。スイッチド ネットワーク（通常はホスト デバイス）へのリンクが 1 つだけあるデバイスでは、使用可能なパスが 1 しかないため、MSTP を実行する必要はありません。さらに、代替パスがないため、単一のリンクで障害が発生するか復元された場合に、トポロジの変更（およびその結果の MAC フラッシュ）が起動されることは望ましくありません。

デフォルトでは、MSTP は、BPDU を受け取らないポートを監視して、タイムアウト後に、MSTP に参加しないようにするエッジモードにします。ただし、エッジ ポートを PortFast として明示的に設定することで、このプロセスを高速化（およびそれによってネットワーク全体のコンバージェンスを改善）できます。

（注）

- Port Fast 設定を有効にするには、ポートを無効にしてから再度有効にする必要があります。ポートを無効にしてから再度有効にするには、（インターフェイス コンフィギュレーション モードで）**shutdown** コマンドと **no shutdown** コマンドを使用します。
- レガシー STP のシスコ実装では、PortFast はシスコ独自の拡張として実装されます。ただし、エッジ ポートと呼ばれる RSTP と MSTP 用の標準に含まれています。

## MSTP ルート ガード

共有管理制御のネットワークでは、ネットワーク管理者が、ネットワーク トポロジの側面および特にルートブリッジの場所を強化することを推奨します。デフォルトでは、より低いプライオリティまたはブリッジ ID がある場合、すべてのデバイスがスパニングツリーのルート ブリッジになることができます。ただし、ネットワークの中心の特定の場所にルートブリッジを配置することで、より最適な転送トポロジを実現できます。

（注） 管理者は、ルート ブリッジの場所を保護するために、ルート ブリッジのプライオリティを 0 に設定できます。ただし、これによって、プライオリティが 0 で、低いブリッジ ID を持つ別のブリッジは保証されません。

ルート ガード機能は、管理者はルート ブリッジを強制的に配置できるメカニズムを提供します。ルートガードがインターフェイスで設定されている場合、そのインターフェイスがルートポート（つまり、ルートに到達できるポート）になるのを防ぎます。通常はルートポートになるインターフェイス上で BPDU を介して優位情報を受信すると、代わりにバックアップ ポートまたは代替ポートになります。この場合、ブロッキング状態になり、データ トラフィックは転送されません。

ルート ブリッジ自体にはルート ポートがありません。このため、管理者は、デバイスのすべてのインターフェイス上でルート ガードを設定することでデバイスを強制的にルートにします。競合する情報を受信するインターフェイスはブロックされます。

（注） ルート ガードはレガシー STP および RSTP のシスコ実装でシスコ独自の拡張として実装されます。ただし、制限付きロールと呼ばれる MSTP 用の標準に含まれています。

## MSTP のトポロジ変更の監視

特定の状況では、特定のポートで発信されたか受信したトポロジ変更を、ネットワークのその他の部分に伝播することが望ましい場合があります。これは、たとえば、ネットワークが単一の管理制御下になく、ネットワーク コアの外部にあるデバイスによるコアでの MAC アドレスのフラッシュを防ぐことが望ましいような場合です。この動作は、ポートのトポロジ変更を設定することでイネーブルにできます。

（注） トポロジ変更ガードは、MSTP 標準の制限 TCN と呼ばれます。

# MSTP サポート機能

Cisco ASR 9000 シリーズ ルータでは、MSTP は、IEEE 802.1Q-2005 で定義されているように物理イーサネット インターフェイスおよびイーサネット バンドル インターフェイスでサポートされます。これには、レガシー STP、RSTP、および PVST の Cisco 実装にある PortFast、BackboneFast、UplinkFast、およびルート ガード機能が含まれることに注意してください。これらの機能は、標準の MSTP プロトコルに含まれるためです。Cisco ASR 9000 シリーズ ルータは、標準 802.1Q モードまたはプロバイダーエッジ（802.1ad）モードのいずれかで動作できます。プロバイダー エッジ モードでは、BPDU には別の MAC アドレスが使用され、802.1Q MAC アドレスで受信されたすべての BPDU がトランスペアレントに転送されます。

また、次の追加のシスコの機能がサポートされます。

- BPDU ガード：このシスコの機能は、エッジ ポートの設定ミスから保護します。
- Flush Containment：このシスコの機能は、トポロジを変更すると発生する不要な MAC フラッシュを防止するために役立ちます。
- 起動遅延：このシスコの機能は、トラフィックを転送する準備が完了する前に、インターフェイスがアクティブ トポロジに追加されないようにします。

（注） 802.1Q 規格で規定されているように、RSTP との相互運用がサポートされます。ただし、レガシー STP との相互運用性はサポートされません。

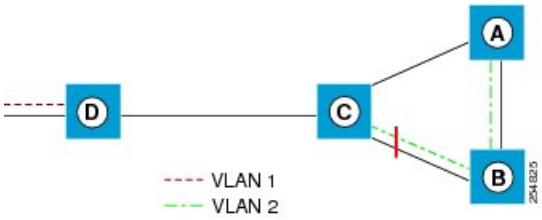
## BPDU ガード

BPDU ガードは、エッジ ポートの設定ミスから保護するシスコの機能です。これは、MSTP の PortFast 機能の拡張です。PortFast がインターフェイスで設定されている場合、MSTP は、スパニングツリーの計算時に、そのインターフェイスをエッジポートであると見なし、考慮の対象から外します。BPDU ガードが設定されている場合、MSTP はさらに、MSTP BPDU を受信すると、errdisable を使用してインターフェイスをシャットダウンします。

## Flush Containment

Flush Containment は、ネットワーク内の他の領域での非関連トポロジの変更が原因で発生する不要な MAC フラッシュを防止するために役立つシスコの機能です。これは、例で詳しく説明します。次の図は、4 つのデバイスが含まれているネットワークを示しています。2 つの VLAN が使用されていて、VLAN 1 はデバイス D でのみ使用され、VLAN 2 はデバイス A、B、C にまたがっています。2 つの VLAN は同じスパニング ツリー インスタンスにありますが、リンクは共有していません。

**図 *57 : Flush Containment***




リンク AB がダウンすると、通常の動作では、C がブロックされたポートを起動し、D を含むその他すべてのインターフェイスでトポロジ変更通知を送信します。これにより、行われたトポロジ変更は VLAN 2 だけに影響を与えるにもかかわらず、VLAN 1 で MAC フラッシュが行われます。

Flush containment は、対象の MSTI で VLAN が設定されていないインターフェイスでトポロジ変更通知が送信されないようにすることで、この問題に対処します。ネットワーク例では、これは、トポロジ変更通知が C から D に送信されないこと、および行われる MAC フラッシュがネットワークの右側に制限されることを意味します。

（注） Flush containment はデフォルトでイネーブルにされますが、設定でディセーブルにできるため、IEEE 802.1Q 規格で規定されている動作が復元されます。

## 起動遅延

起動遅延は、インターフェイスがまだトラフィックを転送する準備が完了していない場合に、スパニングツリーの計算時にMSTPがインターフェイスを考慮しないようにするシスコの機能です。これは、データプレーンがトラフィックを転送する準備が十分に完了する前に、そのカードのインターフェイスがアップしていることをシステムが宣言するため、ラインカードの最初の起動時に役立ちます。標準に従って、MSTPは、アップしていることを宣言するとインターフェイスを考慮します。これによって、新しいインターフェイスが代わりに選択される場合に、他のインターフェイスがブロッキング状態に移行されることがあります。

起動遅延は、MSTPで設定されたインターフェイスが最初に現れたときに発生する設定可能な遅延期間を追加することで、この問題を解決します。この遅延時間が終了するまで、インターフェイスはブロッキング状態のままになり、スパニングツリーの計算時に考慮されません。

起動遅延は、MSTPですでに設定されているインターフェイスの作成時（たとえば、カードのリロード時）だけ発生します。すでに存在しているインターフェイスがMSTPで設定されている場合は、遅延は発生しません。

# MSTP の設定に関する制約事項

次の制限が、MSTP の使用時に適用されます。

- MSTPは、インターフェイス自体（L2モードになっている場合）またはすべてのサブインターフェイスに単純なカプセル化が設定されているインターフェイスだけでイネーブルにする必要があります。これらのカプセル化の一致基準は単純であると見なされます。
  - 一重タグ付き 802.1Q フレーム
  - 二重タグ付き Q-in-Q フレーム（最も外側のタグだけが検査されます）
  - 802.1ad フレーム（MSTP がプロバイダー ブリッジ モードで動作している場合）
  - タグの範囲またはリスト（上記のいずれか）
- L2 インターフェイスまたはサブインターフェイスが、複数の VLAN と一致するカプセル化を使用して設定されている場合、それらの VLAN はすべて同じスパニングツリー インスタンスにマップする必要があります。そのため、各 L2 インターフェイスまたはサブインターフェイスに関連付けられたスパニングツリーインスタンスは1つだけ存在します。
- 特定のブリッジドメインのすべてのインターフェイスまたはサブインターフェイスは、同じスパニングツリー インスタンスに関連付ける必要があります。
- 同じインターフェイス上の複数のサブインターフェイスは、これらのサブインターフェイスが同じスプリット ホライズン グループ内にある場合を除き、同じスパニングツリー インスタンスに関連付けることはできません。つまり、ヘアピニングはできません。
- ネットワーク全体で、L2 インターフェイスまたはサブインターフェイスを、各スパニングツリー インスタンスにマップされたすべての VLAN のすべての冗長パスで設定する必

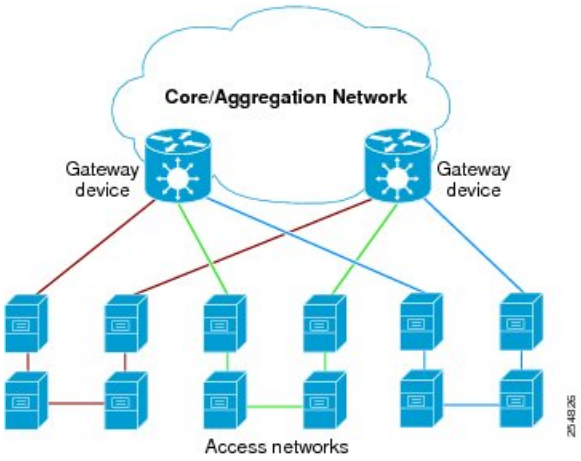
要があります。これは、ポートのSTPブロッキングが原因で接続が誤って切断されることを避けるためです。

**注意** デフォルトまたはタグなしカプセル化を使用するサブインターフェイスは、MSTPステートマシンの障害の原因となります。

# アクセス ゲートウェイ

Cisco ASR 9000シリーズルータに共通する1つの導入シナリオには、uPEアクセスデバイスのネットワークとコアまたは集約ネットワークとの間に配置されたnPEゲートウェイデバイスがあります。各ゲートウェイデバイスは、次の図に示すように、多数のアクセスネットワークの接続を提供できます。アクセスネットワーク（一般的にリング）には、コアまたは集約ネットワークへの冗長リンクがあるため、ネットワークがループフリーのままになるようにするには、STPのいくつかのバリアントまたは類似したプロトコルを使用する必要があります。

**図 *58 :* コアまたは集約ネットワーク**




ゲートウェイ デバイスはSTPプロトコルにも参加できます。ただし、各ゲートウェイ デバイスは多くのアクセスネットワークに接続されているため、これによって、2つのソリューションのうちの1つになります。

- アクセスネットワークをすべてカバーする単一のトポロジが維持されます。これは、1つのアクセス ネットワークのトポロジ変更が、他のすべてのアクセス ネットワークに影響を与えることを意味するため、望ましくありません。
- ゲートウェイ デバイスは、STPプロトコルの複数のインスタンスを、アクセスネットワークごとに1つずつ実行します。これは、アクセス ネットワークごとに別個のプロトコル データベースと別個のプロトコル ステート マシンが維持されることを意味します。これは、ゲートウェイ デバイスで必要なメモリとCPUリソースが原因で望ましくありません。

これらの両方のオプションには重要な欠点があることがわかります。

別の方法として、各アクセスネットワークのレグ間でプロトコル BPDU をトンネリングするが、プロトコル自体には参加しないゲートウェイデバイスがあります。これによって正確なループフリー トポロジになりますが、重要な欠点もあります。

- アクセスリングのレグ間に直接接続されていないため、レグリンクの 1 つの障害が、他のレグに接続されたアクセスデバイスですぐに検出されません。したがって、6 秒以上のトラフィック損失が発生する障害からの回復はプロトコルタイムアウトを待つ必要があります。
- ゲートウェイ デバイスはプロトコルに参加しないため、アクセス ネットワークの任意のトポロジ変更を認識できません。そのため集約ネットワークは、トポロジ変更に従って、誤ったレグによるアクセスネットワーク宛のトラフィックを送信する場合があります。これにより、MAC 学習タイムアウト（デフォルトでは 5 分）の順序でトラフィック損失が発生する可能性があります。

アクセス ゲートウェイは、上記のソリューションの欠点を招くことなく、この導入シナリオに対処することを意図したシスコの機能です。

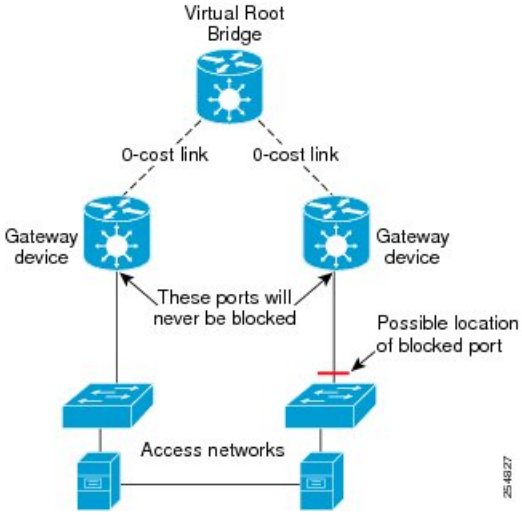
## アクセス ゲートウェイの概要

アクセス ゲートウェイは次の 2 つの前提に基づいています。

- 両方のゲートウェイ デバイスが、常にコアまたは集約ネットワークへの接続を提供します。通常、これにあてはまることを確認するには、コアまたは集約ネットワーク内で使用される復元力メカニズムで十分です。ほとんどの導入では、この復元力を提供するために、コアまたは集約ネットワークで VPLS が使用されます。
- 各アクセス ネットワークのすべてのスパニングツリーで必要なルートは、ゲートウェイ デバイスの 1 つです。これは、（一般に）トラフィックの大部分がアクセスデバイスとコアまたは集約ネットワーク間に存在し、アクセスデバイス間にトラフィックがほとんど存在しない場合にあてはまります。

これらの前提では、STP トポロジには、すべてのスパニングツリーでゲートウェイ デバイスの背後に（つまり、コア側に）仮想ルート ブリッジがあり、両方のゲートウェイ デバイスに仮想ルート ブリッジへのゼロのコスト パスがあると考えることができます。この場合、ゲートウェイ デバイスをアクセス ネットワークに接続するポートは、スパニングツリー プロトコルによってブロックされませんが、常に転送状態にあります。これを次の図に示します。

図 *59 :* アクセス ネットワーク



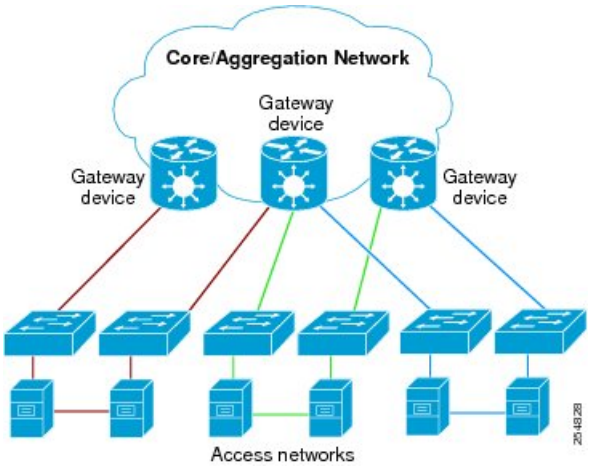

このトポロジでは、ゲートウェイ デバイスによって送信された BPDU が一定であることを確認することができます。これは、（集約またはコア ネットワークは常に接続を提供することを想定しているため）ルートブリッジが変更されることはなく、ポートは常に転送しているという理由から、BPDU で送信される情報は変更されません。

アクセス ゲートウェイは、ゲートウェイ デバイスで完全な STP プロトコルおよび関連するステートマシンを実行する必要性をなくすことでこれを活用し、代わりに、スタティックに設定された BPDU を単にアクセス ネットワークに送信します。BPDU は、完全なプロトコルが実行されている場合に送信される同じ情報を含むように、上記の動作をシミュレート用に設定します。アクセス デバイスには、ゲートウェイ デバイスがプロトコルに完全に参加しているように表示されます。ただし、実際はゲートウェイ デバイスは、スタティック BPDU を送信しているだけであるため、ゲートウェイ デバイスではほとんどメモリまたは CPU リソースは必要なく、多くのネットワーク アクセスを同時にサポートできます。

たいていゲートウェイ デバイスは、アクセス ネットワークから受信した BPDU を無視できます。ただし、1 つの例外は、アクセス ネットワークがトポロジ変更を信号通知する場合です。ゲートウェイ デバイスは、たとえばコアまたは集約ネットワークが VPLS を使用した場合に LDP MAC 取り消しをトリガーすることで、これを適切に実行できます。

多くの場合、ゲートウェイ デバイス間の直接接続は必要ありません。ゲートウェイ デバイスは、アクセス リンク上で設定された BPDU をスタティックに送信するため、（それぞれの設定が一致している限り）それぞれ個別に設定できます。またこれは、次の図に示すように、さまざまなアクセスネットワークがゲートウェイデバイスの異なるペアを使用できることを示します。

図 *60* : アクセスネットワーク




（注） 上の図はアクセスリングを示していますが、一般にアクセス ネットワーク トポロジ、またはゲートウェイデバイスへのリンクの数または場所に制限はありません。

アクセス ゲートウェイによって、次の障害の場合にループ フリー接続が確保されます

- アクセス ネットワークでのリンクの障害。
- アクセス ネットワークとゲートウェイ デバイス間のリンクの障害。
- アクセス デバイスの障害。
- ゲートウェイ デバイスの障害。

## トポロジ変更の伝播

アクセス ネットワーク トポロジの変更を処理するために、2 台のゲートウェイ デバイスが互いに BPDU を交換する必要があるような場合があります。アクセス ネットワークの障害の結果、前にブロックされたポートが転送に移行されるトポロジ変更が発生する場合、アクセスデバイスは、残りのネットワークに変更について通知して、必要な MAC 学習フラッシュをトリガーするように、そのポートにトポロジ変更通知を送信します。通常、トポロジ変更通知は、アクセス ゲートウェイの場合はルートブリッジ方向に送信されます。これは、いずれかのゲートウェイ デバイスに送信されることを意味します。

上記のように、これによって、ゲートウェイ デバイス自体が必要な処理を実行します。ただし、障害によりアクセス ネットワークが分割された場合は、残りのアクセス ネットワーク（つまり、他のゲートウェイデバイスに接続されている部分）にトポロジ変更通知を伝播する必要が生じる場合もあります。これを行うには、ゲートウェイ デバイス間の接続を確認して、各ゲートウェイ デバイスが、受信するトポロジ変更通知をアクセス ネットワークから他のデバイスに伝播できるようにします。ゲートウェイ デバイスはトポロジ変更を示す BPDU を他のゲートウェイ デバイスから受信すると、スタティック BPDU でこれを信号通知します（つまり、アクセス ネットワークに向けて送信します）。

トポロジ変更の伝播は、次の 2 つの条件が満たされた場合だけ必要です。

- アクセスネットワークに3台以上のアクセスデバイスが含まれる場合。デバイスが3台未満の場合、すべてのデバイスが、発生する可能性があるすべての障害を検出する必要があります。
- アクセスデバイスが、コアまたは集約ネットワーク間だけなく、相互にトラフィックを送信する場合。すべてのトラフィックがコアまたは集約ネットワーク間のトラフィックである場合、すべてのアクセスデバイスが、すでに正しい方向でトラフィックを送信しているか、トラフィックの発信元アクセスデバイスからのトポロジ変更を学習する必要があります。

## プリエンプション遅延

アクセス ゲートウェイを支える前提の1つは、ゲートウェイ デバイスはコアまたは集約ネットワークへの接続を提供するために常時使用可能なことです。ただし、この前提が成り立たない可能性のある状況が1つあり、これは起動時に発生します。起動時に、トラフィックをコアまたは集約ネットワークに正常に転送できることを意味する、必要なすべてのシグナリングとコンバージェンスが完了する前に、アクセス側インターフェイスが使用可能になるような場合です。インターフェイスが起動するとすぐにアクセス ゲートウェイが BPDU の送信を開始するため、これによって、ゲートウェイ デバイスで受信する準備が完了する前に、アクセス デバイスがゲートウェイデバイスにトラフィックを送信する可能性があります。この問題を回避するには、プリエンプション遅延機能が使用されます。

プリエンプション遅延機能によって、インターフェイスが起動した後、通常の値に戻るまでの期間にアクセス ゲートウェイは下位 BPDU を送信します。他のゲートウェイ デバイスもダウンしている場合を除き、アクセス ネットワークがすべてのトラフィックを他のゲートウェイ デバイスに送信するようにこれらの下位 BPDU を設定できます。他のゲートウェイ デバイスが使用できない場合、部分的にだけ使用可能でも、トラフィックを完全にドロップするのではなく、このデバイスに送信することを推奨します。したがって、BPDUをまったく送信しないのではなく、下位 BPDU はプリエンプション遅延時間中に送信されます。

## サポートされるアクセス ゲートウェイ プロトコル

アクセスゲートウェイは、次のプロトコルがアクセスネットワークで使用されている場合に、Cisco ASR 9000 シリーズ ルータでサポートされます。

**表 *3:* プロトコル**

| ネットワーク プロトコルへのアクセス | アクセス ゲートウェイ バリアント |
|---|---|
| MSTP | MST アクセス ゲートウェイ（MSTAG） |
| REP | REP アクセスゲートウェイ（REPAG）[1] |
| PVST+ | PVST+ アクセスゲートウェイ（PVSTAG）[2] |
| PVRST | PVRST アクセスゲートウェイ（PVRSTAG）[3] |

1. REP アクセス ゲートウェイは、ゲートウェイ デバイスに接続されているアクセス デバイス インターフェイスが REP MSTP 互換モードで設定されている場合にサポートされます。

2. トポロジ変更の伝播は PVSTAG ではサポートされません。
3. トポロジ変更の伝播は PVRSTAG ではサポートされません。

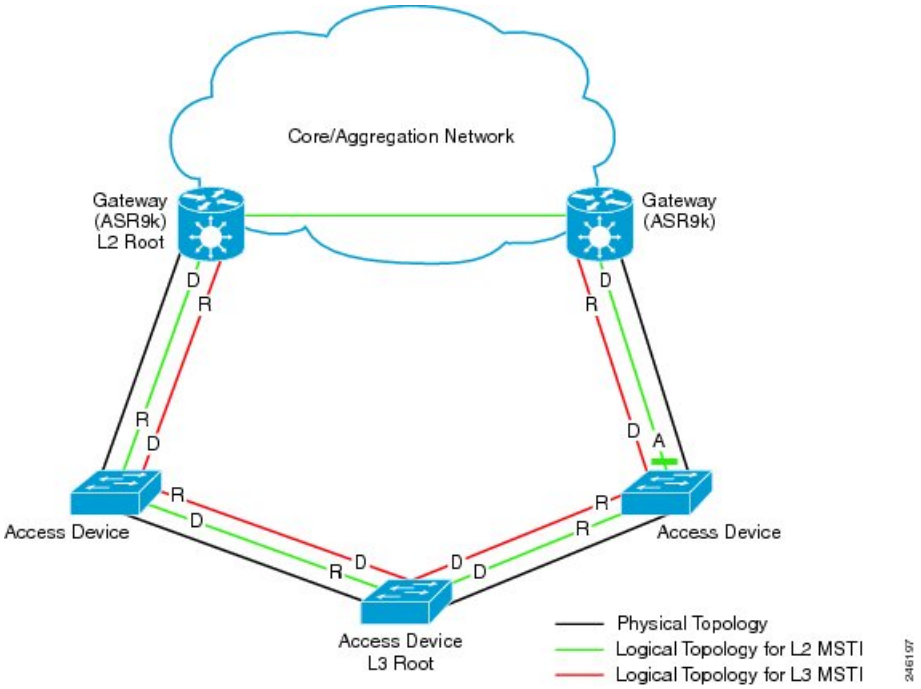
## MSTAG エッジ モード

マルチスパニングツリーインスタンス（MSTI）ごとに、各アクセスデバイスにコアまたは集約ネットワークへのパスが 1 つあることを確認するために、レイヤ 2（L2）環境ではアクセスゲートウェイが使用されます。コアまたは集約ネットワークは、2 台のゲートウェイ デバイス間の L2（イーサネット）接続を提供します。そのため、障害がない場合、各 MSTI のアクセスネットワークにブロックされたポートが少なくとも 1 つ必要です。アクセス リングの場合は、アクセス リングにブロック ポートが 1 つ必要です。各 MSTI では、これは通常、ゲートウェイ デバイスの 1 つに接続されているアップリンク ポートの 1 つです。これは、ゲートウェイ デバイスが最善のマルチスパニングツリープロトコル（MSTP）ルート ノードへの最適なパスを持つように MSTAG を設定することによって行われます。したがって、アクセス デバイスは、ルートに到達するために常にゲートウェイ デバイスを使用し、ゲートウェイ デバイスのポートは常に指定された転送状態になります。

混合レイヤ 2 レイヤ 3 環境では、特定の VLAN のレイヤ 2 サービスおよび他の VLAN のレイヤ 3（L3）サービスを提供するために、L2 アクセス ネットワークが使用されます。アクセスネットワークでは、L2 サービスと L3 サービスに異なる MSTI が使用されます。L2 VLAN の場合、コアまたは集約ネットワークはゲートウェイ デバイス間の L2 接続を提供します。ただし、L3 サービスでは、ゲートウェイ デバイスは L2 ネットワークを終了し、L3 ルーティングを実行します。通常、エンドホストが適切なゲートウェイにルーティングできるように、HSRP や VRRP などの L3 冗長性メカニズムが使用されます。

このシナリオでは、単独で MSTAG を使用しても、L3 MSTI の望ましい動作は達成されません。これは、実際にはループがなくても、アクセス ネットワークのいずれかのポートがブロックされるためです。（これは、L3 VLAN のゲートウェイデバイス間に L2 接続がないためです。）実際は、ゲートウェイデバイスが L3 VLAN の L2 ネットワークを終了するため、望ましい動作とは、アクセスネットワークに MSTP ルートが存在し、ゲートウェイデバイスが単一接続を持つリーフノードとして表示されることです。これを行うには、MSTAG 設定を逆にします。つまり、最低品質のルートに最低品質のパスをアドバタイズするようにゲートウェイ デバイスを設定します。これは、アクセス デバイスはルートとしていずれかのアクセス デバイスを強制的に選択させるため、ポートはブロックされません。この場合、ゲートウェイデバイスのポートは常にルート転送状態になります。MSTAG エッジモード機能は、ゲートウェイ デバイスによってアドバタイズされるロールを指定からルートに変更することで、このシナリオをイネーブルにします。次の図では、このシナリオについて説明します。

図 *61 : MSTAG* エッジ モードのシナリオ




- D：指定ポート（転送）
- R：ルート ポート（転送）
- A：代替ポート（ブロック）

正常な MSTAG と L2 MSTI では、トポロジ変更通知が 1 台のゲートウェイ デバイスから他のゲスト デバイスに伝播され、アクセス ネットワークに再アドバタイズされます。ただし、L3 MSTI の場合、これは望ましくありません。アクセス ネットワークに L3 MSTI のブロックがないため、トポロジ変更通知が永続的にループする可能性があります。この状況を回避するためには、MSTAG エッジ モードで、ゲートウェイ デバイスのトポロジ変更通知の処理を完全にディセーブルにします。

## バンドル インターフェイスの PVSTAG

Per-VLAN スパニングツリー アクセス ゲートウェイ（PVSTAG）のサポートは、バンドル インターフェイスとともに物理インターフェイスにも拡張されています。そのため、PVST アクセス ネットワークをサポートするカスタマーの数の増加に対応できるようになりました。

物理インターフェイスでは、ブリッジプロトコルデータユニット（BDPU）は、インターフェイスをホストするラインカードから送信されます。ただし、バンドルインターフェイス BPDU はルート プロセッサ（RP）から送信されます。RP フェールオーバーが発生しても、バンドル インターフェイスでオーバーフローしたデータ トラフィックは影響を受けません。そのため、BPDU は、フェールオーバーが完了して新しいアクティブ RP に引き継がれるまで送信されません。遅延がある場合、ピア デバイスは BPDU 情報をタイムアウトします。これにより、イーサネット ネットワークの中断の原因になる転送ループが生じる可能性があります。そのため、

RP フェールオーバーが発生した場合、ピア デバイスが BPDU 情報をタイムアウトしないようにすることが重要です。

## Per-VLAN Rapid Spanning Tree

Per-VLAN Rapid Spanning Tree（PVRST）、Rapid PVST、または PVST+ は、VLAN ごとに実装されている IEEE 802.1w（RSTP）規格です。（手作業で STP をディセーブルにしていない場合、）STP の 1 つのインスタンスは、設定されている各 VLAN で実行されます。VLAN 上の各 Rapid PVST+ インスタンスには、1 つのルート スイッチがあります。Rapid PVST+ の実行中には、VLAN ベースで STP をイネーブルまたはディセーブルにできます。

PVST では、ポイントツーポイントの配線を使用して、スパニングツリーの高速収束が行われます。PVRSTによりスパニングツリーの再設定を1秒未満に発生させることができます（802.1D STP のデフォルト設定では 50 秒）。

（注） PVRST では、VLAN ごとに 1 つの STP インスタンスがサポートされます。

PVRST を使用すると、STP コンバージェンスが迅速に行われます。STP にある各指定ポートまたは各ルート ポートにより、デフォルトで、2秒ごとにBPDUが送信されます。トポロジの指定ポートまたはルート ポートで、hello メッセージが 3 回連続失われた場合、または、最大経過時間の期限が切れた場合、ポートでは、すべてのプロトコル情報がテーブルにただちにフラッシュされます。ポートでは、3 つの BPDU が失われるか、最大経過時間の期限が切れた場合、直接のネイバールートまたは指定ポートへの接続が失われたと見なされます。プロトコル情報の急速な経過により、障害検出を迅速に行うことができます。

PVRST では、エッジポートとポイントツーポイントリンクでのみ、フォワーディングステートへの迅速な移行を実現できます。リンク タイプは設定が可能ですが、システムでは、ポートのデュプレックス設定からリンク タイプ情報が自動的に引き継がれます。全二重ポートはポイントツーポイント ポートであると見なされ、半二重ポートは共有ポートであると見なされます。

### 欠点

- パケットレートの増加により負荷が増加しています。
- LAN ごとに 1 つの STP インスタンスがあるため、CPU とメモリの使用率が高くなります。

### IOS-XR での PVRST の実装

IOS-XR での PVRST の実装には、次の特性があります。

- 転送遅延タイマーと最大経過時間タイマーの設定は、VLAN ごとではなく、グローバルでのみサポートされます。
- Hello タイマーの設定は、VLAN ごとではなく、ポートごとでサポートされます。ポートに設定された Hello タイマーは、その特定のポート上のすべての VLAN に適用されます。
- スパニング ツリー バンドル ポートのコストは常に 10000 です。これは、次のいずれの影響も受けません。
    - バンドルメンバーの数または速度

- バンドルメンバーポートの論理状態または管理動作ステータス
- バンドルメンバーの追加または削除

- BPDU ガードが設定されたインターフェイスで BPDU を受信すると、物理インターフェイスと、物理インターフェイスに設定されているレイヤ 2 またはレイヤ 3 サブインターフェイスがエラーディセーブルになります。
- タグなしまたは単一の VLAN タグを持つイーサネットフローポイント（EFP）だけが PVRST によって保護されます。
- ブリッジドメイン内のいずれかの EFP が PVRST によって保護される場合は、そのブリッジドメイン内のすべての EFP が同じ VLAN に属している必要があります。
- ポート上のいずれかの EFP が PVRST によって保護される場合は、そのポート上のすべての EFP が PVRST によって保護される必要があります。
- PVRST は、ラインカードごとに最大 8000 のポートツリーインスタンス（PTI）、システムごとに最大 16000 の PTI をサポートしています。PTI とは、ポートと VLAN の数の積を指します。PVRST は 1000 の VLAN をサポートしています。

# マルチ VLAN 登録プロトコル

マルチ VLAN 登録プロトコルは IEEE 802.1ak で定義され、マルチキャストおよびブロードキャスト フレームの伝播を最適化するために MSTP ベースのネットワークで使用されます。

デフォルトでは、マルチキャストおよびブロードキャストフレームは、スパニングツリーおよびネットワークに接続されている各エッジ（ホスト）デバイスに従って、ネットワーク内の各ポイントに伝播されます。ただし、特定の VLAN では、特定のホストだけがその VLAN のトラフィックの受信に関与する場合があります。さらに、特定のネットワーク デバイスまたは場合によってはネットワークのセグメント全体に、その VLAN のトラフィックの受信に関与する接続済みのホストがないようなことがあります。この場合、その VLAN のトラフィックを、関係のないデバイスに伝播することで、最適化が可能です。MVRP は、各ホストおよびデバイスが、接続されたピアに関与する VLAN を示すことができる、必要なプロトコル シグナリングを提供します。

MVRP がイネーブルにされたデバイスは、次の 2 つのモードで動作します。

- スタティック モード：このモードでは、デバイスは、スタティックに設定された一連の VLAN への関与を宣言する MVRP メッセージを開始します。プロトコルが、MSTP トポロジに対してまだダイナミックであることに注意してください。これは、スタティックな VLAN のセットです。

- ダイナミックモード：このモードでは、デバイスは、異なるポートで受信する MVRP メッセージを処理し、関与する VLAN のセットを決定するためにダイナミックに集約します。これは、このセットへの関与を宣言する MVRP メッセージを送信します。ダイナミックモードでは、またデバイスは受信 MVRP メッセージを使用して、接続デバイスが関与を示した VLAN だけでトラフィックが送信されるように、各ポートから送信されるトラフィックをプルーニングします。

Cisco ASR 9000 シリーズ ルータは、スタティック モードでの動作をサポートしています。これは、MVRP-lite と呼ばれます。

# マルチ スパニングツリー プロトコルの実装方法

この項では、次の手順について説明します。

## MSTP の設定

ここでは、MSTP を設定する手順を説明します。

（注） ここでは、データのスイッチングを設定する方法については説明しません。詳細については、「マルチポイント レイヤ 2 サービスの実装」を参照してください。

### MSTP のイネーブル化

デフォルトでは、STP はすべてのインターフェイス上でディセーブルです。MSTP は、各物理またはイーサネット バンドル インターフェイスの設定によって明示的にイネーブルにする必要があります。MSTP がインターフェイス上で設定されると、そのインターフェイスのサブインターフェイスはすべて自動的に MSTP イネーブルになります。

### MSTP パラメータの設定

MSTP 標準は、多数の設定可能なパラメータを定義します。次にグローバル パラメータを示します。

- リージョン名およびリビジョン
- 起動遅延
- 転送遅延
- 最大経過時間またはホップ
- 転送保留カウント
- プロバイダー ブリッジ モード
- Flush Containment
- VLAN ID（スパニングツリー インスタンスごと）
- ブリッジ プライオリティ（スパニングツリー インスタンスごと）

次に、インターフェイスごとのパラメータを示します。

- 外部ポート パス コスト

- Hello タイム
- リンクタイプ
- PortFast および BPDU ガード
- ルート ガードおよびトポロジ変更ガード
- ポート プライオリティ（スパニングツリー インスタンスごと）
- 内部ポート パス コスト（スパニングツリー インスタンスごと）

インターフェイス単位の設定は、MST コンフィギュレーション サブモード内のインターフェイス サブモードで行われます。

（注） 次の項にリストされている設定手順では、設定可能なパラメータがすべて表示されます。ただし、通常、そのほとんどではデフォルト値を保持できます。

## 手順の概要

1. **configure**
2. **spanning-tree mst** *protocol instance identifier*
3. **bringup delay for** *interval* { **minutes** | **seconds** }
4. **flush containment disable**
5. **name** *name*
6. **revision** *revision* -number
7. **forward-delay** *seconds*
8. **maximum** { **age** *seconds* | **hops** *hops* }
9. **transmit hold-count** *count*
10. **provider-bridge**
11. **instance** *id*
12. **priority** *priority*
13. **vlan-id** *vlan-range* [,*vlan-range* ][,*vlan-range* ] [,*vlan-range* ]
14. **interface** { **Bundle-Ether** | **GigabitEthernet** | **TenGigE** | **FastEthernet** } *instance*
15. **instance** *id* **port-priority** *priority*
16. **instance** *id* **cost** *cost*
17. **external-cost** *cost*
18. **link-type** { **point-to-point** | **multipoint** }
19. **hello-time** *seconds*
20. **portfast** [ **bpdu-guard** ]
21. **guard root**
22. **guard topology-change**
23. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **spanning-tree mst** *protocol instance identifier*

例：

```
RP/0/RSP0/cpu 0: router(config)# spanning-tree mst a
RP/0/RSP0/cpu 0: router(config-mstp)#
```

MSTP コンフィギュレーション サブモードを開始します。

**ステップ 3** **bringup delay for** *interval* { **minutes** | **seconds** }

例：

```
RP/0/RSP0/cpu 0: router(config-mstp)#bringup delay for 10 minutes
```

起動を遅らせる時間間隔を設定します。

**ステップ 4** **flush containment disable**

例：

```
RP/0/RSP0/cpu 0: router(config-mstp)#flush containment disable
```

Flush Containment をディセーブルにします。

このコマンドは、状態に関係なく、すべてのインスタンスの MAC フラッシュを実行します。

**ステップ 5** **name** *name*

例：

```
RP/0/RSP0/cpu 0: router(config-mstp)# name m1
```

MSTP 領域の名前を設定します。

デフォルト値は、IEEE Std 802 で指定する 16 進数表記を使用してテキスト文字列としてフォーマットされたスイッチの MAC アドレスです。

**ステップ 6** **revision** *revision* -number

例：

```
RP/0/RSP0/cpu 0: router(config-mstp)# revision 10
```

MSTP 領域のリビジョン レベルを設定します。

指定できる値は 0 ～ 65535 です。

**ステップ 7** **forward-delay** *seconds*

例：

```
RP/0/RSP0/cpu 0: router(config-mstp)# forward-delay 20
```

ブリッジの転送遅延パラメータを設定します。

ブリッジ転送遅延時間に使用できる秒値は、4 ～ 30 です。

**ステップ 8** **maximum** { **age** *seconds* | **hops** *hops* }

例：

```
RP/0/RSP0/cpu 0: router(config-mstp)# max age 40
RP/0/RSP0/cpu 0: router(config-mstp)# max hops 30
```

ブリッジの最大経過時間および最大ホップ パフォーマンス パラメータを設定します。

ブリッジの最大経過時間に使用できる秒値は、6 ～ 40 です。

ブリッジの最大ホップ数に使用できる秒値は、6 ～ 40 です。

**ステップ 9** **transmit hold-count** *count*

例：

```
RP/0/RSP0/cpu 0: router(config-mstp)# transmit hold-count 8
```

伝送保留カウントのパフォーマンス パラメータを設定します。

指定できる値は 1 ～ 10 です。

**ステップ 10** **provider-bridge**

例：

```
RP/0/RSP0/cpu 0: router(config-mstp)# provider-bridge
```

プロトコルの現在のインスタンスを 802.1ad モードにします。

**ステップ 11** **instance** *id*

例：

```
RP/0/RSP0/cpu 0: router(config-mstp)# instance 101
RP/0/RSP0/cpu 0: router(config-mstp-inst)#
```

MSTI コンフィギュレーション サブモードを開始します。

MSTI ID に使用できる値は、0 ～ 4094 です。

**ステップ 12** **priority** *priority*

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstp-inst)# priority 8192
```

現在の MSTI のブリッジ プライオリティを設定します。

指定できる値は、0 ～ 61440（4096 の倍数）です。

**ステップ 13** **vlan-id** *vlan-range* [,*vlan-range* ][,*vlan-range* ] [,*vlan-range* ]

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstp-inst)# vlan-id 2-1005
```

現在の MSTI と一連の VLAN ID を関連付けます。

VLAN のリストの範囲は、a-b、c、d、e-f、g などです。

（注） 各 MSTI に対してステップ 11 ～ 13 を繰り返します。

**ステップ 14** **interface** { **Bundle-Ether** | **GigabitEthernet** | **TenGigE** | **FastEthernet** } *instance*

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstp)# interface FastEthernet 0/0/0/1
RP/0/RSP0/cpu 0: router(config-mstp-if)#
```

MSTP インターフェイス コンフィギュレーション サブモードを開始し、特定のポートの STP をイネーブルにします。

ラック、スロット、インスタンス、またはポート形式でインターフェイスを転送します。

**ステップ 15** **instance** *id* **port-priority** *priority*

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstp-if)# instance 101 port-priority 160
```

MSTI にポート プライオリティのパフォーマンス パラメータを設定します。

MSTI ID に使用できる値は、0 ～ 4094 です。

ポート プライオリティに使用できる値は、0 ～ 240（16 の倍数）です。

**ステップ 16** **instance** *id* **cost** *cost*

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstp-if)# instance 101 cost 10000
```

現在のポートの特定のインスタンスに関する内部パス コストを設定します。

MSTI ID に使用できる値は、0 ～ 4094 です。

ポート コストに使用できる値は、1 ～ 200000000 です。

各インターフェイスの MSTI ごとにステップ 15 および 16 を繰り返します。

**ステップ 17** **external-cost** *cost*

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstp-if)# external-cost 10000
```

現在の外部ポート パス コストを設定します。

ポート コストに使用できる値は、1 ～ 200000000 です。

**ステップ 18** **link-type** { **point-to-point** | **multipoint** }

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstp-if)# link-type point-to-point
```

ポートのリンク タイプをポイントツーポイントまたはマルチポイントに設定します。

**ステップ 19** **hello-time** *seconds*

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstp-if)# hello-time 1
```

ポートの hello タイムを秒単位で設定します。

使用できる値は 1 および 2 です。

**ステップ 20** **portfast** [ **bpdu-guard** ]

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstp-if)# portfast
RP/0/RSP0/cpu 0: router(config-mstp-if)# portfast bpduguard
```

ポート上で PortFast をイネーブルにし、任意で BPDU ガードをイネーブルにします。

**ステップ 21** **guard root**

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstp-if)# guard root
```

ポート上で RootGuard をイネーブルにします。

**ステップ 22** **guard topology-change**

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstp-if)# guard topology-change
```

ポート上で TopologyChangeGuard をイネーブルにします。

（注） インターフェイスごとにステップ 14 ～ 22 を繰り返します。

**ステップ 23** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## MSTP の確認

次の show コマンドを使用して、MSTP の動作を確認できます。

- **show spanning-tree mst** *mst-name*
- **show spanning-tree mst** *mst-name* **interface** *interface-name*
- **show spanning-tree mst** *mst-name* **errors**
- **show spanning-tree mst** *mst-name* **configuration**
- **show spanning-tree mst** *mst-name* **bpdu interface** *interface-name*
- **show spanning-tree mst** *mst-name* **topology-change flushes**

# MSTAG または REPAG の設定

ここでは、MSTAG を設定する手順を説明します。

（注） REPAG の設定手順は同じです。

ここでは、データのスイッチングを設定する方法については説明しません。詳細については、「マルチポイント レイヤ 2 サービスの実装」モジュールを参照してください。

# タグなしサブインターフェイスの設定

物理またはバンドル イーサネット インターフェイスで MSTAG をイネーブルにするには、最初に encapsulation untagged コマンドを使用して、タグなしパケットと一致する L2 サブインターフェイスを設定する必要があります。L2 サブインターフェイスの設定の詳細については、「The Cisco ASR 9000 Series Routers Carrier Ethernet Model」モジュールを参照してください。

## MSTAG のイネーブル化

MSTAG は、対応するタグなしサブインターフェイス上で明示的に設定することによって、物理インターフェイスまたはバンドル イーサネット インターフェイスでイネーブルにします。MSTAG はタグなしサブインターフェイスで設定されている場合、物理またはバンドル イーサネット インターフェイスと、その物理またはバンドル イーサネット サブインターフェイス上の他のすべてのサブインターフェイスで自動的にイネーブルになります。

## MSTAG パラメータの設定

MSTAG パラメータは各インターフェイスで個別に設定され、MSTAG は各インターフェイスで完全に独立して動作します。（ルートを同じアクセス ネットワークに接続している場合を除き）異なるインターフェイスの MSTAG パラメータ間の対話はありません。

これらのパラメータは、インターフェイスごとに設定できます。

- リージョン名およびリビジョン
- ブリッジ ID
- ポート ID
- 外部ポート パス コスト
- 最大経過時間
- プロバイダー ブリッジ モード
- Hello タイム

次の MSTAG パラメータは、各スパニングツリーインスタンスのインターフェイスごとに設定可能です。

- VLAN IDs
- ルート ブリッジ プライオリティおよび ID
- ブリッジプライオリティ
- ポートプライオリティ
- 内部ポート パス コスト

アクセス ネットワーク全体に一貫した動作を確保するには、設定時に次のガイドラインを使用する必要があります。

- アクセス ネットワーク内のデバイスのルート ブリッジ プライオリティおよび ID よりもよい（低い）ルート ブリッジ プライオリティおよび ID を（スパニングツリー インスタンスごとに）使用して両方のゲートウェイ デバイスを設定する必要があります。ゲートウェイ デバイスでは、ルート ブリッジ プライオリティおよび ID を 0 に設定することを推奨します。

（注） アクセスデバイスで検出されたSTPの矛盾を回避するには、両方のゲートウェイ デバイスで同じルート プライオリティおよび ID を設定する必要があります。

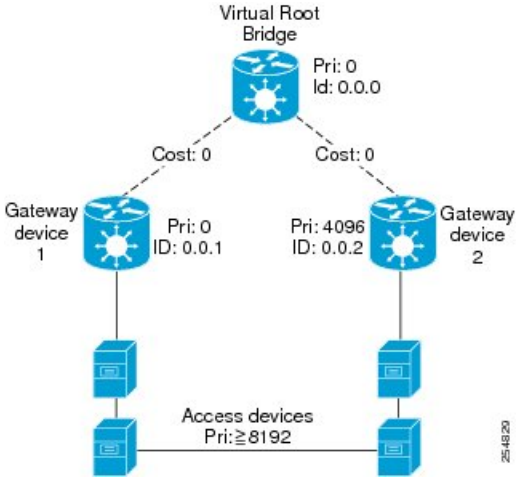
- ゲートウェイ デバイスは両方とも、ポート パス コストを 0 にして設定する必要があります。
- 各スパニングツリー インスタンスでは、ルート ブリッジ プライオリティおよび ID よりも高いが、ネットワーク内の他のデバイス（他のゲートウェイデバイスを含む）のブリッジ プライオリティおよび ID よりも低いブリッジ プライオリティおよび ID を使用して、1 つのゲートウェイ デバイスを設定する必要があります。ブリッジプライオリティを 0 に設定することを推奨します。
- スパニングツリー インスタンスごとに、ルート ブリッジ プライオリティおよびID、最初のゲートウェイデバイスブリッジプライオリティおよびIDよりも高いが、アクセス ネットワーク内のデバイスのブリッジ プライオリティおよびIDよりも低いブリッジ プライオリティおよびIDを使用して、2番目のゲートウェイ デバイスを設定する必要があります。ブリッジ プライオリティを 4096 に設定することを推奨します（これは、0 よりも大きい最低許容値です）。
- ゲートウェイ デバイスよりも高いブリッジ プライオリティを使用してすべてのアクセス デバイスを設定する必要があります。8192 以上の値を使用することを推奨します。
- スパニングツリーインスタンスごとに、すべてのリンクがアップすると目的のポートがブロック状態になるように、アクセスデバイスでポートパスコストおよびその他のパラメータを設定する場合があります。

注意 MSTAG 設定のチェックはありません。設定ミスによって、アクセス デバイスの MSTP プロトコルの誤った動作が発生する可能性があります（たとえば、STP の矛盾が検出されます）。

下図で、上記のガイドラインについて説明します。

（注） トポロジの変更がシグナリングされると、アクセス デバイスはゲートウェイ デバイスから受信した情報を無視する場合のように、これらのガイドラインは、REPAG には適用されません。

図 *62 : MSTAG* のガイドライン




（注） 次の項にリストされている設定手順では、設定可能なパラメータがすべて表示されます。ただし、通常、そのほとんどではデフォルト値を保持できます。

## 手順の概要

1. **configure**
2. **spanning-tree mstag** *protocol instance identifier*
3. **preempt delay for** interval{ **seconds** | **minutes** | **hours** }
4. **interface** { **Bundle-Ether** | **GigabitEthernet** | **TenGigE** | **FastEthernet** } *instance.subinterface*
5. **name** *name*
6. **revision** *revision* -number
7. **max age** *seconds*
8. **provider-bridge**
9. **bridge-id** *id*
10. **port-id** *id*
11. **external-cost** *cost*
12. **hello-time** *seconds*
13. **instance** *id*
14. **edge mode**
15. **vlan-id** *vlan-range* [, *vlan-range* ] [,*vlan-range* ] [,*vlan-range* ]
16. **priority** *priority*
17. **port-priority** *priority*
18. **cost** *cost*
19. **root-bridge** *id*
20. **root-priority** *priority*
21. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **spanning-tree mstag** *protocol instance identifier*

例：

```
RP/0/RSP0/cpu 0: router(config)# spanning-tree mstag a
RP/0/RSP0/cpu 0: router(config-mstag)#
```

MSTAG コンフィギュレーション サブモードを開始します。

**ステップ 3** **preempt delay for** interval{ **seconds** | **minutes** | **hours** }

例：

```
RP/0/RSP0/cpu 0: router(config-mstag)# preempt delay for 10 seconds
```

プリエンプション処理を行うまでに起動 BPDU を送信する遅延時間を指定します。

**ステップ 4** **interface** { **Bundle-Ether** | **GigabitEthernet** | **TenGigE** | **FastEthernet** } *instance.subinterface*

例：

```
RP/0/RSP0/cpu 0: router(config-mstag)# interface GigabitEthernet0/2/0/30.1
RP/0/RSP0/cpu 0: router(config-mstag-if)#
```

MSTAG インターフェイス コンフィギュレーション サブモードを開始し、特定のポートの MSTAG をイネーブルにします。

**ステップ 5** **name** *name*

例：

```
RP/0/RSP0/cpu 0: router(config-mstag-if)# name leo
```

MSTP 領域の名前を設定します。

デフォルト値は、IEEE 規格 802 で指定する 16 進数表記を使用してテキスト文字列としてフォーマットされたスイッチの MAC アドレスです。

**ステップ 6** **revision** *revision* -number

例：

```
RP/0/RSP0/cpu 0: router(config-mstag-if)# revision 1
```

MSTP 領域のリビジョン レベルを設定します。

指定できる値は 0 ～ 65535 です。

**ステップ 7** **max age** *seconds*

例：

```
RP/0/RSP0/cpu 0: router(config-mstag-if)# max age 20
```

ブリッジの最大経過時間のパフォーマンス パラメータを設定します。

ブリッジの最大経過時間に使用できる秒値は、6 ～ 40 です。

**ステップ 8** **provider-bridge**

例：

```
RP/0/RSP0/cpu 0: router(config-mstag-if)# provider-bridge
```

プロトコルの現在のインスタンスを 802.1ad モードにします。

**ステップ 9** **bridge-id** *id*

例：

```
RP/0/RSP0/cpu 0: router(config-mstag-if)# bridge-id 001c.0000.0011
```

現在のスイッチのブリッジ ID を設定します。

**ステップ 10** **port-id** *id*

例：

```
RP/0/RSP0/cpu 0: router(config-mstag-if)# port-id 111
```

現在のスイッチのポート ID を設定します。

**ステップ 11** **external-cost** *cost*

例：

```
RP/0/RSP0/cpu 0: router(config-mstag-if)# external-cost 10000
```

現在の外部ポート パス コストを設定します。

ポート コストに使用できる値は、1 ～ 200000000 です。

**ステップ 12** **hello-time** *seconds*

例：

```
RP/0/RSP0/cpu 0: router(config-mstag-if)# hello-time 1
```

ポートの hello タイムを秒単位で設定します。

指定できる値は 1 ～ 2 です。

**ステップ 13** **instance** *id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstag-if)# instance 1
```

MSTI コンフィギュレーション サブモードを開始します。

MSTI ID に使用できる値は、0 ～ 4094 です。

**ステップ 14** **edge mode**

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstag-if-inst)# edge mode
```

この MSTI のアクセス ゲートウェイ エッジ モードをイネーブルにします。

**ステップ 15** **vlan-id** *vlan-range* [, *vlan-range* ] [,*vlan-range* ] [,*vlan-range* ]

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstag-if-inst)# vlan-id 2-1005
```

現在の MSTI と一連の VLAN ID を関連付けます。

VLAN のリストの範囲は、a-b、c、d、e-f、g などです。

**ステップ 16** **priority** *priority*

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstag-if-inst)# priority 4096
```

現在の MSTI のブリッジ プライオリティを設定します。

指定できる値は、0 ～ 61440（4096 の倍数）です。

**ステップ 17** **port-priority** *priority*

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstag-if-inst)# port-priority 160
```

MSTI にポート プライオリティのパフォーマンス パラメータを設定します。

ポート プライオリティに使用できる値は、0 ～ 240（16 の倍数）です。

**ステップ 18** **cost** *cost*

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstag-if-inst)# cost 10000
```

現在のポートの特定のインスタンスに関する内部パス コストを設定します。

ポート コストに使用できる値は、1 ～ 200000000 です。

**ステップ 19** **root-bridge** *id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstag-if-inst)# root-id 001c.0000.0011
```

現在のポートから送信された BPDU のルート ブリッジ ID を設定します。

**ステップ 20** **root-priority** *priority*

**例**：

```
RP/0/RSP0/cpu 0: router(config-mstag-if-inst)# root-priority 4096
```

このポートから送信された BPDU のルート ブリッジ プライオリティを設定します。

（注） 各インターフェイスを設定するにはステップ 4 ～ 19 を繰り返し、インターフェイスごとに各 MSTI を設定するにはステップ 13 ～ 19 を繰り返します。

**ステップ 21** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## MSTAG トポロジ変更の伝播の設定

MSTAG トポロジ変更の伝播は、単に 2 台のゲートウェイ デバイスの MSTAG 対応インターフェイス間の接続を設定することによって設定されます。

1. MSTAG を上記のように設定します。使用するタグなしサブインターフェイスに留意してください。
2. ゲートウェイ デバイス間の接続を設定します。これは、MPLS 疑似回線経由で接続するか、直接物理リンクが存在する場合は VLAN サブインターフェイスになります。

3. 他のゲートウェイ デバイスへのタグなしサブインターフェイスおよびリンク（PW またはサブインターフェイス）が含まれている各ゲートウェイデバイスでポイントツーポイント（P2P）の相互接続を設定します。

MSTAG 用に設定されたタグなしサブインターフェイスが P2P の相互接続に追加されると、MSTAG トポロジ変更の伝播が自動的にイネーブルになります。MSTAG は、トポロジの変更の検出時に信号通知するよう、その他のゲートウェイ デバイスへの相互接続によって BDPU を転送します。

MPLS 疑似回線または P2P の相互接続の設定の詳細については、「ポイントツーポイント レイヤ 2 サービスの実装」モジュールを参照してください。

## MSTAG の確認

次の show コマンドを使用して、MSTAG の動作を確認できます。

- **show spanning-tree mstag** *mst-name*
- **show spanning-tree mstag** *mst-name* **bpdu interface** *interface-name*
- **show spanning-tree mstag** *mst-name* **topology-change flushes**

REPAG では類似するコマンドを使用できます。

# PVSTAG または PVRSTAG の設定

ここでは、PVSTAG を設定する手順を説明します。

PVRSTAG の設定手順は同じです。

（注） ここでは、データのスイッチングを設定する方法については説明しません。詳細については、「マルチポイント レイヤ 2 サービスの実装」モジュールを参照してください。

## PVSTAG のイネーブル化

PVSTAG は、PVSTAG 用の物理インターフェイスおよび VLAN を明示的に設定することで、その物理インターフェイスで特定の VLAN に対してイネーブルになります。

## PVSTAG パラメータの設定

次に、各 VLAN のインターフェイスごとに設定可能な PVSTAG パラメータを示します。

- ルート プライオリティおよび ID
- ルート コスト
- ブリッジ プライオリティおよび ID
- ポート プライオリティおよび ID

- 最大経過時間
- Hello タイム

正常に動作するには、PVSTAG の設定時に次のガイドラインに従う必要があります。

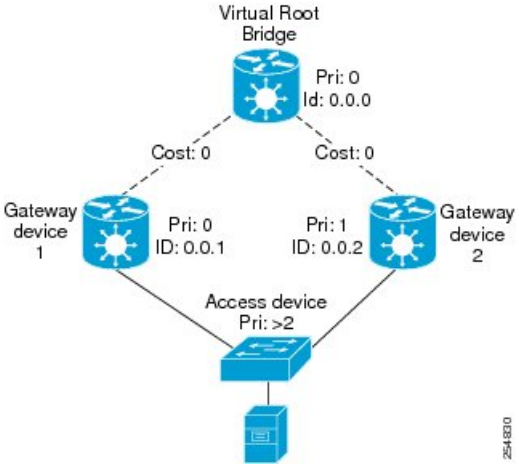
- アクセス ネットワーク内のデバイスのブリッジ プライオリティおよび ID よりもよい（低い）ルート ブリッジ プライオリティおよび ID を使用して両方のゲートウェイ デバイスを設定する必要があります。ゲートウェイ デバイスでは、ルートブリッジプライオリティおよび ID を 0 に設定することを推奨します。
- ゲートウェイ デバイスは両方とも、ルート コストを 0 にして設定する必要があります。
- ルート ブリッジ プライオリティおよび ID よりも高いが、ネットワーク内の他のデバイス（他のゲートウェイ デバイスを含む）のブリッジ プライオリティおよび ID よりも低いブリッジ プライオリティおよび ID を使用して、1 つのゲートウェイ デバイスを設定する必要があります。ブリッジ プライオリティを 0 に設定することを推奨します。
- ルート ブリッジ プライオリティおよび ID、最初のゲートウェイ デバイス ブリッジ プライオリティおよび ID よりも高いが、アクセス ネットワーク内のデバイスのブリッジ プライオリティおよび ID よりも低いブリッジ プライオリティおよび ID を使用して、2 番目のゲートウェイ デバイスを設定する必要があります。ブリッジプライオリティは、PVSTAG では 1、PVRSTAG では 4096 に設定することを推奨します。（PVRSTAG の場合、これは、0 よりも大きい最低許容値です）。
- ゲートウェイ デバイスよりも高いブリッジ プライオリティを使用してすべてのアクセス デバイスを設定する必要があります。PVSTAG では 2 以上の値、PVRSTAG では 8192 以上の値を使用することを推奨します。
- スパニングツリーインスタンスごとに、すべてのリンクがアップすると目的のポートがブロック状態になるように、アクセスデバイスでポートパス コストおよびその他のパラメータを設定する場合があります。

⚠

**注意** PVSTAG 設定のチェックはありません。設定ミスによって、アクセス デバイスの PVST プロトコルの誤った動作が発生する可能性があります（たとえば、STP の矛盾が検出されます）。

下図で、上記のガイドラインについて説明します。

図 *63 : PVSTAG* のガイドライン




（注） 次の項にリストされている設定手順では、設定可能なパラメータがすべて表示されます。ただし、通常、そのほとんどではデフォルト値を保持できます。

### PVSTAG トポロジの制約事項

次の制約事項が PVSTAG トポロジに適用されます。

- 1 つのアクセス デバイスだけをゲートウェイ デバイスに接続できます。
- 1 つの VLAN のトポロジ変更通知は、その物理インターフェイスのすべての VLAN およびブリッジ ドメインに影響します。

## 手順の概要

1. **configure**
2. **spanning-tree mstag pvstag** *protocol instance identifier*
3. **preempt delay for** *interval* { **seconds | minutes | hours** }
4. **interface type** *interface-path-id* または **interface Bundle-Ether** *bundle-id*
5. **vlan** *vlan-id*
6. **root-priority** *priority*
7. **root-id** *id*
8. **root-cost** *cost*
9. **priority** *priority*
10. **bridge-id** *id*
11. **port-priority** *priority*
12. **port-id** *id*
13. **hello-time** *seconds*
14. **max age** *seconds*
15. **commit** コマンドまたは **end** コマンドを使用します。

手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **spanning-tree mstag pvstag** *protocol instance identifier*

例：

```
RP/0/RSP0/cpu 0: router(config)# spanning-tree pvstag a
RP/0/RSP0/cpu 0: router(config-pvstag)#
```

PVSTAG コンフィギュレーション サブモードを開始します。

**ステップ 3** **preempt delay for** *interval* { **seconds | minutes | hours** }

例：

```
RP/0/RSP0/cpu 0: router(config-pvstag)# preempt delay for 10 seconds
```

プリエンプション処理を行うまでに起動 BPDU を送信する遅延時間を指定します。

**ステップ 4** **interface type** *interface-path-id* または **interface Bundle-Ether** *bundle-id*

例：

```
RP/0/RSP0/cpu 0: router(config-pvstag)# interface GigabitEthernet0/2/0/30.1
RP/0/RSP0/cpu 0: router(config-pvstag-if)#

or

RP/0/RSP0/cpu 0: router(config-pvstag)# interface Bundle-Ether 100
RP/0/RSP0/cpu 0: router(config-pvstag-if)#
```

PVSTAG インターフェイス コンフィギュレーション サブモードを開始し、特定のポートの PVSTAG をイネーブルにします。

**ステップ 5** **vlan** *vlan-id*

例：

```
RP/0/RSP0/cpu 0: router(config-pvstag-if)# vlan 200
```

このインターフェイスで VLAN をイネーブルにして設定します。

**ステップ 6** **root-priority** *priority*

例：

```
RP/0/RSP0/cpu 0: router(config-pvstag-if-vlan)# root-priority 4096
```

このポートから送信された BPDU のルート ブリッジ プライオリティを設定します。

**ステップ 7** **root-id** *id*

例：

```
RP/0/RSP0/cpu 0: router(config-pvstag-if-vlan)# root-id 0000.0000.0000
```

ポートから送信された BPDU のルート ブリッジの ID を設定します。

**ステップ 8** **root-cost** *cost*

例：

```
RP/0/RSP0/cpu 0: router(config-pvstag-if-vlan)# root-cost 10000
```

このインターフェイスから BPDU で送信するルート パス コストを設定します。

**ステップ 9** **priority** *priority*

例：

```
RP/0/RSP0/cpu 0: router(config-pvstag-if-vlan)# priority 4096
```

現在の MSTI のブリッジ プライオリティを設定します。

PVSTAG の場合、使用できる値は 0 ～ 65535 で、PVRSTAG の場合、使用できる値は 0 ～ 61440（4096 の倍数）です。

**ステップ 10** **bridge-id** *id*

例：

```
RP/0/RSP0/cpu 0: router(config-pvstag-if-vlan)# bridge-id 001c.0000.0011
```

現在のスイッチのブリッジ ID を設定します。

**ステップ 11** **port-priority** *priority*

例：

```
RP/0/RSP0/cpu 0: router(config-pvstag-if-vlan)# port-priority 160
```

MSTI にポート プライオリティのパフォーマンス パラメータを設定します。

PVSTAG の場合、ポート プライオリティに使用できる値は 0 ～ 255 で、PVRSTAG の場合、使用できる値は 0 ～ 240（16 の倍数）です。

**ステップ 12** **port-id** *id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-pvstag-if-vlan)# port-id 111
```

現在のスイッチのポート ID を設定します。

**ステップ 13** **hello-time** *seconds*

**例**：

```
RP/0/RSP0/cpu 0: router(config-pvstag-if-vlan)# hello-time 1
```

ポートの hello タイムを秒単位で設定します。

指定できる値は 1 ～ 2 です。

**ステップ 14** **max age** *seconds*

**例**：

```
RP/0/RSP0/cpu 0: router(config-pvstag-if-vlan)# max age 20
```

ブリッジの最大経過時間のパフォーマンス パラメータを設定します。

ブリッジの最大経過時間に使用できる秒値は、6 ～ 40 です。

（注） 各インターフェイスを設定するにはステップ 4 ～ 14 を繰り返し、インターフェイスごとに各 VLAN を設定するにはステップ 5 ～ 14 を繰り返します。

**ステップ 15** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## サブインターフェイスの設定

インターフェイスの PVSTAG でイネーブルになっている VLAN ごとに、その VLAN のトラフィックと一致する対応するサブインターフェイスを設定する必要があります。これはデータのスイッチングと PVST BPDU の両方に使用されます。サブインターフェイスを設定するときには、次のガイドラインに従ってください。

- VLAN 1 は PVST のネイティブ VLAN として扱われます。したがって、VLAN 1 の場合は、タグなしパケット（**encapsulation untagged**）と一致するサブインターフェイスを設定する

必要があります。また、VLAN 1 を明示的にタグ付けされたパケット（**encapsulation dot1q 1**）と一致するサブインターフェイスを設定する必要が生じる場合があります。

- PVST では dot1q パケットだけが許可されます。Q-in-Q および dot1ad パケットはプロトコルでサポートされていないため、これらのカプセル化で設定されたサブインターフェイスは、PVSTAG で正しく動作しません。
- VLAN の範囲と一致するサブインターフェイスは PVSTAG でサポートされます。これがデータ スイッチングのプロビジョニングで望ましい場合を除き、VLAN ごとに個別のサブインターフェイスを設定する必要はありません。
- PVSTAG は次をサポートしていません。
  - L2 モードで設定された物理インターフェイス
  - デフォルトのカプセル化（**encapsulation default**）で設定されているサブインターフェイス
  - VLAN（**encapsulation dot1q any**）と一致するように設定されたサブインターフェイス

L2 サブインターフェイスの設定の詳細については、「ポイントツーポイント レイヤ 2 サービスの実装」モジュールを参照してください。

## PVSTAG の確認

次の show コマンドを使用して、PVSTAG または PVRSTAG の動作を確認できます。

- **show spanning-tree pvstag** *mst-name*
- **show spanning-tree pvstag** *mst-name*

特に、これらのコマンドは各 VLAN に使用するサブインターフェイスを表示します。

# PVRST の設定

### 始める前に

次の内容を確認してください。

- VLAN カプセル化を使用した L2 トランスポート サブインターフェイスが定義されている。
- スパニングツリーを実行しているすべての VLAN のブリッジグループにある L2VPN ブリッジドメインが設定され、対応する L2 トランスポート サブインターフェイスがブリッジドメインで設定されている。

### 手順の概要

1. **configure**
2. **spanning-tree pvrst** *protocol-instance-name*

3. **apply-group** *group_name group_name*
4. **forward-delay** *seconds*
5. **maximum age** *seconds*
6. **transmit hold-count** *count*
7. **vlan** *vlan_id*
8. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **spanning-tree pvrst** *protocol-instance-name*

例：

```
RP/0/RSP0/cpu 0: router(config)# spanning-tree pvrst stp
```

PVRST 設定サブモードを開始します。

**ステップ 3** **apply-group** *group_name group_name*

例：

```
RP/0/RSP0/cpu 0: router(config-pvrst)# apply-group groupA groupB
```

1 つのグループから別のグループに設定を適用できます。

**ステップ 4** **forward-delay** *seconds*

例：

```
RP/0/RSP0/cpu 0: router(config-pvrst)# forward-delay 10
```

ブリッジの転送遅延時間（秒単位）を設定できます。

転送遅延は、ポートがスパニングツリー ラーニングおよびリスニング ステートからフォワーディング ステートに変更するまでに待機する秒数です。時間遅延範囲は 4 ～ 30 秒です。

**ステップ 5** **maximum age** *seconds*

例：

```
RP/0/RSP0/cpu 0: router(config-pvrst)# maximum age 10
```

ブリッジの最大経過時間（秒単位）を設定します。

最大エージング タイムは、再構成を試行するまでにスイッチがスパニングツリー コンフィギュレーション メッセージを受信せずに待機する秒数です。最大経過時間の範囲は 6 ～ 40 秒です。

**ステップ 6** **transmit hold-count** *count*

**例**：

```
RP/0/RSP0/cpu 0: router(config-pvrst)# transmit hold-count 4
```

ブリッジ送信ホールドカウントを設定できます。ホールドカウントの範囲は 1 ～ 10 です。

**ステップ 7** **vlan** *vlan_id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-pvrst)#
```

VLAN に PVRST を設定できます。VLAN ID の範囲は 1 ～ 4094 です。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# MVRP-lite の設定

ここでは、MVRP-lite を設定する手順を説明します。

## MVRP-lite のイネーブル化

MVRP ライトが設定されている場合、MSTP がイネーブルであるすべてのインターフェイスで自動的にイネーブルになります。MSTP は、MVRP をイネーブルにする前に設定する必要があります。MSTP の設定の詳細については、「MSTP の設定」を参照してください。

## MVRP-lite パラメータの設定

次に、設定可能な MVRP-lite パラメータを示します。

- 定期的な送信
- Join 時間
- Leave 時間

- Leave-all 時間

## 手順の概要

1. **configure**
2. **spanning-tree mst** *protocol instance identifier*
3. **mvrp static**
4. **periodic transmit** [ **interval** *seconds* ]
5. **join-time** *milliseconds*
6. **leave-time** *seconds*
7. **leaveall-time** *seconds*
8. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モード を開始します。

**ステップ 2** **spanning-tree mst** *protocol instance identifier*

例：

```
RP/0/RSP0/cpu 0: router(config)#
spanning-tree mst aRP/0/RSP0/CPU0:router(config-mstp)#
```

MSTP コンフィギュレーション サブモードを開始します。

**ステップ 3** **mvrp static**

例：

```
RP/0/RSP0/cpu 0: router(config-mstp)#mvrp static
```

この MSTP プロトコル インスタンスを実行するように MVRP を設定します。

**ステップ 4** **periodic transmit** [ **interval** *seconds* ]

例：

```
RP/0/RSP0/cpu 0: router(config-mvrp)#
periodic transmit
```

すべてのアクティブ ポートで定期的なマルチ VLAN 登録プロトコル データ ユニット（MVRPDU）を送信します。

**ステップ 5** **join-time** *milliseconds*

例：

```
RP/0/RSP0/cpu 0: router(config-mvrp)#
 hello-time 1
```

すべてのアクティブ ポートの Join 時間を設定します。

**ステップ 6** **leave-time** *seconds*

例：

```
RP/0/RSP0/cpu 0: router(config-mvrp)# leave-time 20
```

すべてのアクティブ ポート Leave 時間を設定します。

**ステップ 7** **leaveall-time** *seconds*

例：

```
RP/0/RSP0/cpu 0: router(config-mvrp)# leaveall-time 20
```

権限をすべてのアクティブ ポートの Leave all 時間を設定します。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## MVRP-lite の確認

次の show コマンドを使用して、MVRP-lite の動作を確認できます。

- **show ethernet mvrp mad**
- **show ethernet mvrp status**
- **show ethernet mvrp statistics**

# MSTP の実装の設定例

ここでは、次の設定例を示します。

# MSTP の設定：例

次に、MSTPが単一のインターフェイスでイネーブルになっている単一スパニングツリーインスタンスの MSTP 設定例を示します。

```
config
spanning-tree mst example
    name m1
    revision 10
    forward-delay 20
    maximum hops 40
    maximum age 40
    transmit hold-count 8
    provider-bridge
    bringup delay for 60 seconds
    flush containment disable
    instance 101
        vlans-id 101-110
        priority 8192
    !
    interface GigabitEthernet0/0/0/0
        hello-time 1
        external-cost 10000
        link-type point-to-point
        portfast
        guard root
        guard topology-change
        instance 101 cost 10000
        instance 101 port-priority 160
    !
!
```

次に、スパニングツリープロトコルの状態の概要を生成する **show spanning-tree mst** コマンドの出力例を示します。

```
# show spanning-tree mst example
Role:  ROOT=Root, DSGN=Designated, ALT=Alternate, BKP=Backup, MSTR=Master
State: FWD=Forwarding, LRN=Learning, BLK=Blocked, DLY=Bringup Delayed

Operating in dot1q mode


MSTI 0 (CIST):

  VLANS Mapped: 1-9,11-4094

  CIST Root  Priority    4096
             Address     6262.6262.6262
             This bridge is the CIST root
             Ext Cost    0

  Root ID    Priority    4096
             Address     6262.6262.6262
             This bridge is the root
             Int Cost    0
             Max Age 20 sec, Forward Delay 15 sec


  Bridge ID  Priority    4096 (priority 4096 sys-id-ext 0)
             Address     6262.6262.6262
             Max Age 20 sec, Forward Delay 15 sec
             Max Hops 20, Transmit Hold count  6
```

```
Interface    Port ID           Role State Designated           Port ID
             Pri.Nbr Cost                 Bridge ID             Pri.Nbr
------------ ------- --------- ---- ----- -------------------- -------
Gi0/0/0/0    128.1   20000     DSGN FWD   4096  6262.6262.6262 128.1
Gi0/0/0/1    128.2   20000     DSGN FWD   4096  6262.6262.6262 128.2
Gi0/0/0/2    128.3   20000     DSGN FWD   4096  6262.6262.6262 128.3
Gi0/0/0/3    128.4   20000     ---- BLK   ----- -------------- -------


MSTI 1:

  VLANS Mapped: 10

  Root ID    Priority    4096
             Address     6161.6161.6161
             Int Cost    20000
             Max Age 20 sec, Forward Delay 15 sec


  Bridge ID  Priority    32768 (priority 32768 sys-id-ext 0)
             Address     6262.6262.6262
             Max Age 20 sec, Forward Delay 15 sec
             Max Hops 20, Transmit Hold count  6


Interface    Port ID           Role State Designated           Port ID
             Pri.Nbr Cost                 Bridge ID             Pri.Nbr
------------ ------- --------- ---- ----- -------------------- -------
Gi0/0/0/0    128.1   20000     ROOT FWD   4096  6161.6161.6161 128.1
Gi0/0/0/1    128.2   20000     ALT  BLK   4096  6161.6161.6161 128.2
Gi0/0/0/2    128.3   20000     DSGN FWD   32768 6262.6262.6262 128.3
Gi0/0/0/3    128.4   20000     ---- BLK   ----- -------------- -------

================================================================================
```

**show spanning-tree mst** の出力例では、最初の行は、MSTP が dot1q またはプロバイダーブリッジモードで動作しているかどうかを示し、この情報の後に各 MSTI の詳細が表示されます。

各 MSTI について、次の情報が表示されます。

- MSTI の VLAN のリスト。
- CIST の場合、CIST ルートのプライオリティおよびブリッジ ID、および CIST ルートに到達するための外部パス コスト。またこの出力は、このブリッジが CIST ルートであるかどうかを示します。
- この MSTI のルート ブリッジのプライオリティおよびブリッジ ID、およびルートに到達するための内部パス コスト。またこの出力は、このブリッジが MSTI のルートであるかどうかを示します。
- MSTI のルート ブリッジから受信した最大経過時間および転送遅延時間。
- この MSTI のこのブリッジのプライオリティおよびブリッジ ID。
- このブリッジの最大経過時間、転送遅延、最大ホップ、および転送保留カウント（すべての MSTI で同じです）。
- MSTP 対応インターフェイスのリスト。各インターフェイスについて、次の情報が表示されます。

- インターフェイス名。
- この MSTI のこのインターフェイスのポート プライオリティおよびポート ID。
- この MSTI のこのインターフェイスのポート コスト。
- 現在のポートの役割。
  - DSGN：指定：これは、この MSTI のこの LAN 上の指定ポートです。
  - ROOT：ルート：この MSTI のブリッジのルート ポートです。
  - ALT：代替：これは、この MSTI の代替ポートです。
  - BKP：バックアップ：これは、この MSTI のバックアップ ポートです。
  - MSTR：マスター：これは、CIST のルート ポートまたは代替ポートである境界ポートです。

  インターフェイスがダウンしているか、起動遅延タイマーが実行されていて、役割がまだ割り当てられていません。
- 現在のポート状態。
  - BLK：ポートはブロックされています。
  - LRN：ポートを学習中です。
  - FWD：ポートは転送中です。
  - DLY：起動遅延タイマーが実行中です。
- ポートが境界ポートであり、CIST はなく、ポートが指定されていない場合は、境界ポートだけが表示され、残りの情報は表示されません。
- ポートがアップしていないか、起動遅延タイマーが動作している場合、情報は残りのフィールドに表示されません。それ以外の場合は、インターフェイスが接続されている LAN の指定ブリッジのブリッジ プライオリティおよびブリッジ ID が表示され、その後に LAN 上の指定ポートのポート プライオリティおよびポート ID が表示されます。ポートの役割が指定されていない場合、このブリッジまたはポートの情報が表示されます。

次に、上述した標準コマンドよりもインターフェイスの状態に関する詳細な情報を生成する、**show spanning-tree mst** コマンドの出力例を示します。

```
# show spanning-tree mst a interface GigabitEthernet0/1/2/1
GigabitEthernet0/1/2/1
Cost: 20000
link-type: point-to-point
hello-time 1
Portfast: no
BPDU Guard: no
Guard root: no
Guard topology change: no
BPDUs sent 492, received 3
```

```
MST 3:
 Edge port:
 Boundary : internal
 Designated forwarding
 Vlans mapped to MST 3: 1-2,4-2999,4000-4094
 Port info port id 128.193 cost 200000
 Designated root address 0050.3e66.d000 priority 8193 cost 20004
 Designated bridge address 0002.172c.f400 priority 49152 port id 128.193
 Timers: message expires in 0 sec, forward delay 0, forward transitions 1
 Transitions to reach this state: 12
```

出力には、すべての MSTI に適用されるインターフェイスに関するインターフェイス情報が表示されます。

- コスト
- リンク タイプ
- hello-time
- portfast（BPDU ガードがイネーブルかどうかなど）
- ガードのルート
- ガードのトポロジ変更
- 送受信された BPDU

また、各 MSTI に固有の情報が含まれます。

- ポート ID、プライオリティ、コスト
- ルートからの BPDU 情報（ブリッジ ID、コスト、プライオリティ）
- このポートで送信される BPDU 情報（ブリッジ ID、コスト、プライオリティ）
- この状態に達するまでの状態遷移
- トポロジは、この状態になるように変更されます。
- この MSTI の Flush containment ステータス

次に、MSTP 用に設定されているが、MSTP が動作していないインターフェイスに関する情報を生成する、**show spanning-tree mst** errors の出力例を示します。これは主に、存在しないインターフェイスに関する情報を表示します。

```
# show spanning-tree mst a errors
Interface               Error
------------------------------
GigabitEthernet1/2/3/4   Interface does not exist.
```

次に、MSTI マッピングテーブルに VLAN ID を表示する、**show spanning-tree mst** configuration の出力例を示します。また、送信された BPDU に含まれる設定ダイジェストを表示します。これは、同じ MSTP リージョン内の他のブリッジから受信したダイジェストと一致する必要があります。

```
# show spanning-tree mst a configuration
Name          leo
Revision      2702
Config Digest 9D-14-5C-26-7D-BE-9F-B5-D8-93-44-1B-E3-BA-08-CE
Instance  Vlans mapped
--------  ------------------------------
0         1-9,11-19,21-29,31-39,41-4094
1         10,20,30,40
---------------------------------------
```

次に、特定のローカルインターフェイスで出力および受信される BPDU の詳細を生成する、**show spanning-tree mst** の出力例を示します。

（注） 共有 LAN 上で動作する MSTP の場合は、複数の受信パケットを保存できます。

```
# show spanning-tree mst a bpdu interface GigabitEthernet0/1/2/2
direction transmit
MSTI 0 (CIST):
 Root ID : 0004.9b78.0800
 Path Cost : 83
 Bridge ID : 0004.9b78.0800
 Port ID : 12
 Hello Time : 2
 ...
```

次に、各インターフェイスの MSTI ごとに発生したトポロジ変更の詳細を表示する、**show spanning-tree mst** の出力例を示します。

```
# show spanning-tree mst M topology-change flushes instance$
MSTI 1:

Interface     Last TC               Reason                           Count
------------  --------------------  -------------------------------- -----
Te0/0/0/1     04:16:05 Mar 16 2010  Role change: DSGN to ----           10
#
#
# show spanning-tree mst M topology-change flushes instance$
MSTI 0 (CIST):

Interface     Last TC               Reason                           Count
------------  --------------------  -------------------------------- -----
Te0/0/0/1     04:16:05 Mar 16 2010  Role change: DSGN to ----           10
#
```

# MSTAG の設定：例

次に、単一のインターフェイスでの単一スパニングツリーインスタンスのMSTAG設定例を示します。

```
config
interface GigabitEthernet0/0/0/0.1 l2transport
    encapsulation untagged
!
spanning-tree mstag example
    preempt delay for 60 seconds
    interface GigabitEthernet0/0/0/0.1
        name m1
```

```
        revision 10
        external-cost 0
        bridge-id 0.0.1
        port-id 1
        maximum age 40
        provider-bridge
        hello-time 1
        instance 101
              edge-mode
            vlans-id 101-110
            root-priority 0
            root-id 0.0.0
            cost 0
            priority 0
            port-priority 0
        !
    !
!
```

次に、MSTAG トポロジ変更の伝搬の追加設定例を示します。

```
l2vpn
    xconnect group example
        p2p mstag-example
            interface GigabitEthernet0/0/0/0.1
            neighbor 123.123.123.1 pw-id 100
        !
    !
!
```

次に、**show spanning-tree mstag** の出力例を示します。

```
# show spanning-tree mstag A
GigabitEthernet0/0/0/1
  Preempt delay is disabled.
  Name:            6161:6161:6161
  Revision:        0
  Max Age:         20
  Provider Bridge: no
  Bridge ID:       6161.6161.6161
  Port ID:         1
  External Cost:   0
  Hello Time:      2
  Active:          no
  BPDUs sent:      0
    MSTI 0 (CIST):
    VLAN IDs:         1-9,32-39,41-4094
    Role:             Designated
    Bridge Priority:  32768
    Port Priority:    128
    Cost:             0
    Root Bridge:      6161.6161.6161
    Root Priority:    32768
    Topology Changes: 123
  MSTI 2
    VLAN IDs:         10-31
    Role:             Designated
    Bridge Priority:  32768
    Port Priority:    128
    Cost:             0
    Root Bridge:      6161.6161.6161
    Root Priority:    32768
    Topology Changes: 123
  MSTI 10
 VLAN IDs:         40
```

```
    Role:             Root (Edge mode)
    Bridge Priority:  32768
    Port Priority:    128
    Cost:             200000000
    Root Bridge:      6161.6161.6161
    Root Priority:    61440
    Topology Changes: 0
```

次に、特定のローカルインターフェイスで出力および受信される BPDU の詳細を生成する、**show spanning-tree mstag bpdu interface** の出力例を示します。

```
RP/0/RSP0/CPU0:router#show spanning-tree mstag foo bpdu interface GigabitEthernet 0/0/0/0
Transmitted:
  MSTI 0 (CIST):
ProtocolIdentifier: 0
ProtocolVersionIdentifier: 3
BPDUType: 2
CISTFlags: Top Change Ack  0
           Agreement       1
           Forwarding      1
           Learning        1
           Role            3
           Proposal        0
           Topology Change 0
CISTRootIdentifier: priority 8, MSTI 0, address 6969.6969.6969
CISTExternalPathCost: 0
CISTRegionalRootIdentifier: priority 8, MSTI 0, address 6969.6969.6969
CISTPortIdentifierPriority: 8
CISTPortIdentifierId: 1
MessageAge: 0
MaxAge: 20
HelloTime: 2
ForwardDelay: 15
Version1Length: 0
Version3Length: 80
FormatSelector: 0
Name: 6969:6969:6969
Revision: 0
MD5Digest: ac36177f 50283cd4 b83821d8 ab26de62
CISTInternalRootPathCost: 0
CISTBridgeIdentifier: priority 8, MSTI 0, address 6969.6969.6969
CISTRemainingHops: 20
  MSTI 1:
MSTIFlags: Master          0
           Agreement       1
           Forwarding      1
           Learning        1
           Role            3
           Proposal        0
           Topology Change 0
MSTIRegionalRootIdentifier: priority 8, MSTI 1, address 6969.6969.6969
MSTIInternalRootPathCost: 0
MSTIBridgePriority: 1
MSTIPortPriority: 8
MSTIRemainingHops: 20
```

次に、インターフェイスごとに発生したトポロジ変更の詳細を表示する、**show spanning-tree mstag topology-change flushes** の出力例を示します。

```
#show spanning-tree mstag b topology-change flushes

MSTAG Protocol Instance b

Interface     Last TC             Reason                          Count
```

```
------------  ------------------  -------------------------------- -----
Gi0/0/0/1     18:03:24 2009-07-14  Gi0/0/0/1.10 egress TCN          65535
Gi0/0/0/2     21:05:04 2009-07-15  Gi0/0/0/2.1234567890 ingress TCN     2
```

# PVSTAG の設定：例

次に、単一のインターフェイスでの単一 VLAN の PVSTAG 設定例を示します。

```
config
spanning-tree pvstag example
    preempt delay for 60 seconds
    interface GigabitEthernet0/0/0/0
        vlan 10
            root-priority 0
            root-id 0.0.0
            root-cost 0
            priority 0
            bridge-id 0.0.1
            port-priority 0
            port-id 1
            max age 40
            hello-time 1
        !
    !
!
```

次に、**show spanning-tree pvstag** の出力例を示します。

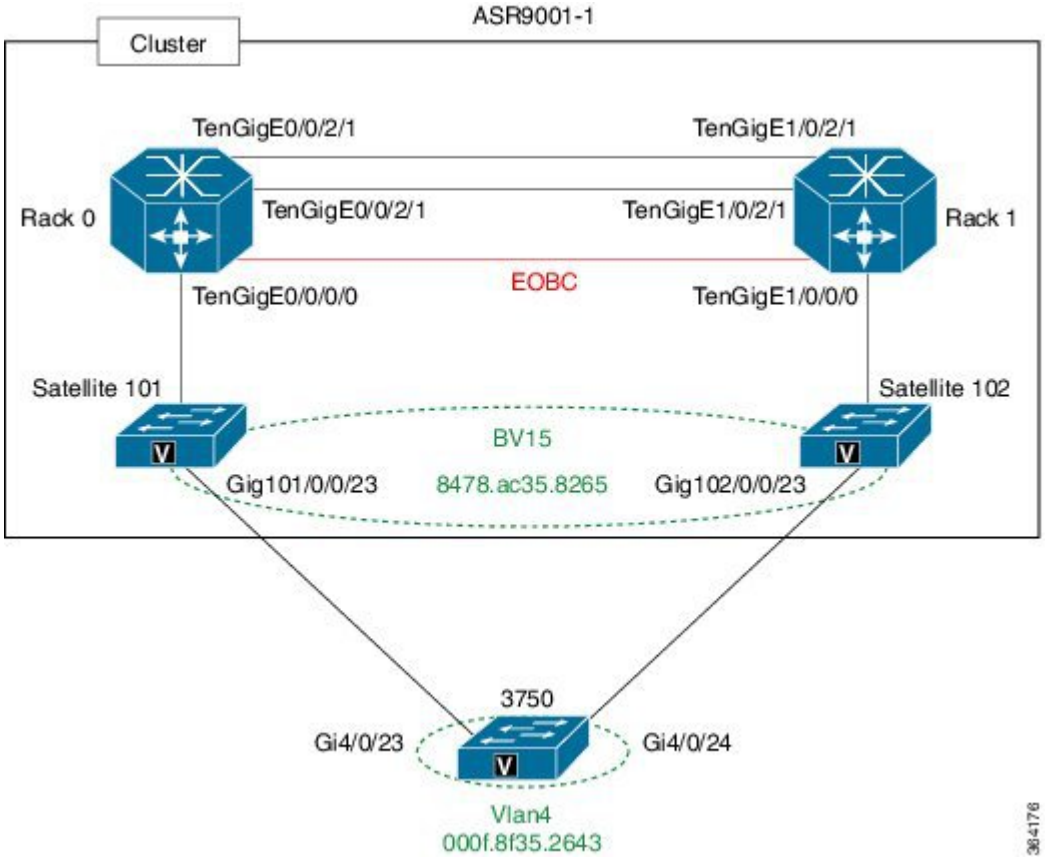
```
# show spanning-tree pvstag interface GigabitEthernet0/0/0/1
GigabitEthernet0/0/0/1
  VLAN 10
    Preempt delay is disabled.
    Sub-interface:    GigabitEthernet0/0/0/1.20 (Up)
    Max Age:          20
    Root Priority:    0
    Root Bridge:      0000.0000.0000
    Cost:             0
    Bridge Priority:  32768
    Bridge ID:        6161.6161.6161
    Port Priority:    128
    Port ID:          1
    Hello Time:       2
    Active:           no
    BPDUs sent:       0
    Topology Changes: 123
    VLAN 20
```

# サテライトを使用するクラスタでの PVSTAG の設定：例

この例は、次のネットワークトポロジ図に示されているクラスタでの PVSTAG の適用を示しています。大規模なファンアウトのために、このクラスタにはサテライトスイッチがあります。このクラスタは、ルーテッドインターフェイス（BVI）を介して L3 終端を提供します。コアに VPLS はありません。このシナリオでは、クラスタからスパニングツリー全体をなくすことのみを目的として、PVSTAG が設定されます。

この PVSTAG の例を簡略化するために、ネットワークトポロジには 1 台のアクセススイッチのみがあります。実際のシナリオでは、アクセススイッチのリングに置き換えることができます。

図 ***64 :*** サテライトを使用するクラスタでの ***PVSTAG***




クラスタの設定：

```
!
spanning-tree pvstag inst4
interface GigabitEthernet101/0/0/23
vlan 4
root-priority 4096
root-id 0011.0011.0011
priority 8192
bridge-id 0011.0011.0011
port-priority 128
!
!
interface GigabitEthernet102/0/0/23
vlan 4
root-priority 4096
root-id 0011.0011.0011
priority 8192
bridge-id 0022.0022.0022
port-priority 128
!
!
!
interface Loopback100
ipv4 address 8.8.8.8 255.255.255.255
!
interface Loopback201
ipv4 address 9.9.9.9 255.255.255.255
```

```
!
interface GigabitEthernet101/0/0/23
cdp
!
interface GigabitEthernet102/0/0/23
cdp
!
interface GigabitEthernet101/0/0/23.4 l2transport
encapsulation dot1q 4
rewrite ingress tag pop 1 symmetric
!
interface GigabitEthernet102/0/0/23.4 l2transport
encapsulation dot1q 4
rewrite ingress tag pop 1 symmetric
!
interface TenGigE0/0/0/0
ipv4 point-to-point
ipv4 unnumbered Loopback100
nv
satellite-fabric-link satellite 101
remote-ports GigabitEthernet 0/0/0-43
!
!
!
interface TenGigE0/0/2/0
nv
edge
interface
!
!
!
interface TenGigE0/0/2/1
nv
edge
interface
!
!
!
interface TenGigE1/0/0/0
ipv4 point-to-point
ipv4 unnumbered Loopback201
nv
satellite-fabric-link satellite 102
remote-ports GigabitEthernet 0/0/0-43
!
!
!
interface BVI5
ipv4 address 4.4.4.1 255.255.255.0
!
l2vpn
bridge group bg1
bridge-domain bd1
interface GigabitEthernet101/0/0/23.4
!
interface GigabitEthernet102/0/0/23.4
!
routed interface BVI5
!
!
!
nv
satellite 101
type asr9000v
```

```
serial-number CAT1641U0QV
ipv4 address 10.22.1.2
!
satellite 102
type asr9000v
serial-number CAT1635U14B
ipv4 address 10.23.1.2
!
```

**3750 の設定**：

```
!
interface GigabitEthernet4/0/23
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 4
switchport mode trunk
!
interface GigabitEthernet4/0/24
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 4
switchport mode trunk
!
interface Vlan4
ip address 4.4.4.2 255.255.255.0
!
```

# PVRST の設定：例

次に、PVRST の設定例を示します。

```
(config)# spanning-tree pvrst stp1
(config-pvrst)# forward-delay 6
(config-pvrst)# interface GigabitEthernet 0/1/1/2 hello-time 2
(config-pvrst)# maximum age 35
(config-pvrst)# transmit hold-count 9
(config-pvrst)# vlan 666 priority 4096
(config-pvrst)# commit
```

# MVRP-Lite の設定：例

次に、MVRP-lite の設定例を示します。

```
config
spanning-tree mst example
    mvrp static
        periodic transmit
        join-time 200
        leave-time 30
        leaveall-time 10
    !
!
```

次は、**show ethernet mvrp mad** の出力例を示しています。

```
RP/0/RSP0/CPU0:router# show ethernet mvrp mad interface GigabitEthernet 0/1/0/1
GigabitEthernet0/1/0/1
  Participant Type: Full; Point-to-Point: Yes
  Admin Control: Applicant Normal; Registrar Normal
```

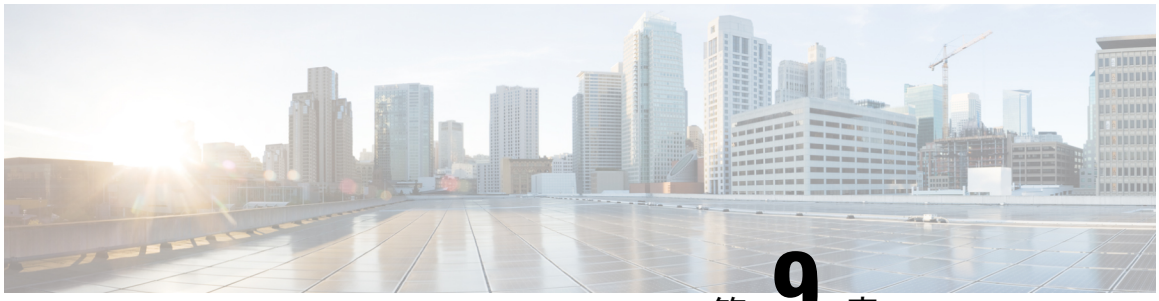
```
  LeaveAll Passive (next in 5.92s); periodic disabled
  Leave in 25.70s; Join not running
  Last peer 0293.6926.9585; failed registrations: 0

VID   Applicant              Registrar
----  ---------------------  ---------
   1  Very Anxious Observer  Leaving
 283  Quiet Passive          Empty
```

次は、**show ethernet mvrp status** の出力例を示しています。

```
RP/0/RSP0/CPU0:router# show ethernet mvrp status interface GigabitEthernet 0/1/0/1
GigabitEthernet0/1/0/1
  Statically declared:  1-512,768,980-1034
  Dynamically declared: 2048-3084
  Registered:           1-512
```

次は、**show ethernet mvrp statistics** の出力例を示しています。

```
RP/0/RSP0/CPU0:router# show ethernet mvrp statistics interface GigabitEthernet 0/1/0/1
GigabitEthernet0/1/0/1
  MVRPDUs TX:    1245
  MVRPDUs RX:       7
  Dropped TX:       0
  Dropped RX:      42
  Invalid RX:      12
```



第 9 章

# レイヤ 2 アクセスリストの実装

イーサネット サービス アクセス コントロール リスト（ACL）は、レイヤ 2 ネットワーク トラフィック プロファイルを集合的に定義する 1 つ以上のアクセス コントロール エントリ（ACE）で構成されます。このプロファイルは、Cisco IOS XR ソフトウェア機能で参照できます。各イーサネット サービス ACL には、送信元および宛先アドレス、サービス クラス（CoS）、または VLAN ID などの基準に基づいたアクション要素（許可または拒否）が含まれます。

このモジュールでは、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータでのイーサネット サービス アクセス リストの実装に必要なタスクについて説明します。

（注） このモジュールに記載されているイーサネット サービス アクセス リスト コマンドの詳細については、『Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference』の「Ethernet Services (Layer 2) Access List Commands on Cisco ASR 9000 Series Routers」モジュールを参照してください。この章で使用される他のコマンドの説明については、コマンド リファレンスのマスター索引を参照するか、またはオンラインで検索してください。

**Cisco ASR 9000 シリーズ ルータでのイーサネット サービス アクセス リスト実装の機能履歴**

| リリース | 変更内容 |
|---|---|
| リリース 3.7.2 | この機能は、Cisco ASR 9000 シリーズ ルータに追加されました。 |



# レイヤ 2 アクセス リスト実装の前提条件

この前提条件は、アクセスリストおよびプレフィックスリストの実装に適用されます。

適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。

ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

# レイヤ 2 アクセス リストの実装に関する情報

## イーサネット サービス アクセス リスト機能のハイライト

イーサネット サービス アクセス リストには、次の機能のハイライトがあります。

- 特定のシーケンス番号を使用してアクセス リストのカウンタをクリアする機能。
- 別のアクセス リストに既存のアクセス リストの内容をコピーする機能。
- ユーザがシーケンス番号を permit または deny ステートメントに追加し、そのようなステートメントのシーケンスの再設定、追加、または名前付きアクセス リストからの削除を行うことができるようにします。
- パケットを転送するためにインターフェイスでパケット フィルタリングを実行します。
- イーサネット サービス ACL は、インターフェイス、VLAN サブインターフェイス、バンドル イーサネット インターフェイス、EFP、バンドル イーサネット インターフェイスを介した EFP で適用できます。イーサネット サービス ACL のアトミック置換は、これらの物理インターフェイスでサポートされています。

## イーサネット サービス アクセス リストの目的

イーサネット サービス アクセス リストは、ACL ベースの転送（ABF）を使用して、ネットワークを介して移動するパケットおよび場所を制御するパケット フィルタリングを実行します。そのような制御は、着信および発信ネットワーク トラフィックを制限し、ポート レベルでネットワークにユーザおよびデバイスのアクセスを制限するために役立ちます。

## イーサネット サービス アクセス リストの仕組み

イーサネット サービス アクセス リストは、レイヤ 2 設定に適用される、permit および deny ステートメントで構成される順序付きリストです。アクセス リストには、参照に使用される名前があります。

アクセス リストを設定して名前を付けることは可能ですが、アクセス リストを受け取るコマンドによってアクセス リストが参照されるまで、有効にはなりません。複数のコマンドから同じアクセス リストを参照できます。アクセス リストで、ルータに到達するレイヤ 2 トラフィッ

ク、またはルータ経由で送信されるレイヤ2トラフィックは制御できますが、ルータが送信元のトラフィックは制御できません。

## イーサネット サービス アクセス リストのプロセスおよびルール

イーサネット サービス アクセス リストの設定時は、次のプロセスとルールを使用します。

- ソフトウェアは、アクセス リストの条件に対してフィルタされる各パケットの送信元アドレスや宛先アドレスをテストします。一度に 1 つの条件（permit または deny ステートメント）がテストされます。
- パケットがアクセス リストのステートメントに一致しないと、そのパケットはリスト内の次のステートメントに対してテストされます。
- パケットとアクセス リストのステートメントが一致すると、リスト内の残りのステートメントはスキップされ、パケットは一致したステートメントに指定されたとおりに許可または拒否されます。パケットが許可されるか拒否されるかは、パケットが一致する最初のエントリによって決まります。つまり、一致すると、それ以降のエントリは考慮されません。
- アクセス リストがアドレスまたはプロトコルを拒否する場合は、ソフトウェアはパケットを廃棄します。
- 各アクセス リストの最後には暗黙の deny ステートメントがあるため、一致する条件がない場合は、パケットはドロップされます。つまり、各ステートメントに対してテストするときまでにパケットを許可または拒否しないと、パケットは拒否されます。
- アクセス リストには permit ステートメントを 1 つ以上含める必要があります。そうしないと、パケットはすべて拒否されます。
- 最初に一致が見つかった後は条件のテストが終了するため、条件の順序は重要です。同じ permit ステートメントまたは deny ステートメントでも、順序が異なる場合、ある状況では通過し、別の状況では拒否されるパケットが生じる可能性があります。
- インバウンド アクセス リストは、ルータに到達するパケットを処理します。着信パケットの処理後に、アウトバウンドインターフェイスへのルーティングが行われます。インバウンドアクセス リストが効率的なのは、フィルタリングテストで拒否されたことでパケットが廃棄される場合、ルーティング検索のオーバーヘッドが抑えられるためです。パケットがテストで許可されると、そのパケットに対してルーティングの処理が実施されます。インバウンド リストの場合、permit とは、インバウンド インターフェイスで受信したパケットを引き続き処理することを意味します。deny とは、パケットを破棄することです。
- アウトバウンド アクセス リストの場合、パケットの処理後にルータから送信されます。着信パケットはアウトバウンドインターフェイスにルーティングされてから、アウトバウンドアクセス リストで処理されます。アウトバウンドリストの場合、許可とは、出力バッファに対して送信されることを示し、拒否とは、パケットが廃棄されることを示します。
- アクセス リストは、使用中のアクセス グループによって適用されている場合には削除できません。アクセス リストを削除するには、まずアクセス リストを参照しているアクセス グループを削除してから、アクセス リストを削除します。

- **ethernet-services access-group** コマンドを使用するには、アクセス リストが必要です。

## イーサネット サービス アクセス リストを作成する際に役立つヒント

イーサネット サービス アクセス リストの作成時は、次の点に注意してください。

- アクセス リストは、インターフェイスに適用する前に作成します。
- より具体的な参照が、より全般的な参照よりも前に出現するように、アクセス リストを構成します。

## 送信元アドレスと宛先アドレス

送信元 MAC アドレスと宛先 MAC アドレスの 2 つのフィールドは、アクセス リストの基礎として最も一般的なフィールドです。送信元 MAC アドレスを指定して、特定のネットワーキング デバイスまたはホストからのパケットを制御します。宛先 MAC アドレスを指定して、特定のネットワーキング デバイスまたはホストに送信されるパケットを制御します。

# イーサネット サービス アクセス リスト エントリのシーケンス番号

イーサネット サービス アクセス リスト エントリにシーケンス番号を適用する機能によって、アクセス リストの変更が簡単になります。アクセス リスト エントリのシーケンス番号機能を使用すると、アクセス リスト エントリにシーケンス番号を追加して、シーケンス番号を再設定できます。新しいエントリを追加する場合、アクセス リストの目的の位置に挿入されるようにシーケンス番号を選択します。必要に応じて、アクセス リストの現在のエントリを並べ替えて、新しいエントリを挿入できる場所を作成できます。

## シーケンス番号の動作

次に、シーケンス番号の動作について詳細に説明します。

- シーケンス番号のないエントリを複数適用すると、最初のエントリにシーケンス番号 10 が割り当てられ、それ以降のエントリには 10 ずつ増分したシーケンス番号が割り当てられます。最大シーケンス番号は 2147483646 です。生成したシーケンス番号がこの最大値を超えると、次のメッセージが表示されます。

  ```
  Exceeded maximum sequence number.
  ```

- シーケンス番号のないエントリを 1 つ指定すると、アクセス リストの最後のシーケンス番号に 10 を加えたシーケンス番号が割り当てられ、リストの末尾に配置されます。
- ACL エントリは、トラフィック フローにもハードウェアのパフォーマンスにも影響を及ぼすことなく追加できます。
- ルート スイッチ プロセッサ（RSP）とインターフェイス カードにあるエントリのシーケンス番号が常に同期されるよう、分散サポートが提供されます。

# レイヤ 2 アクセス リストの実装方法

## レイヤ 2 アクセス リスト実装の制約事項

次の制限事項が、イーサネット サービス アクセス リストの実装に適用されます。

- イーサネット サービス アクセス リストは、管理インターフェイスではサポートされていません。
- NetIO（ソフトウェア低速パス）は、イーサネットサービスアクセスリストではサポートされません。
- 内部 VLAN 0 と外部 VLAN 0 での一致は、Cisco ASR 9000 高密度 100GE イーサネットラインカードおよびASR 9000 拡張イーサネットラインカードではサポートされていません。

## イーサネット サービス アクセス リストの設定

このタスクでは、イーサネット サービス アクセス リストを設定します。

### 手順の概要

1. **configure**
2. **ethernet-service access-list** *name*
3. [ *sequence-number* ] **{ permit | deny }** { *src-mac-address src-mac-mask* | **any** | **host** } [ { *ethertype-number* } | **vlan** *min-vlan-ID* [ *max-vlan-ID* ] ] [ **cos** *cos-value* ] [ **dei** ] [ **inner-vlan** *min-vlan-ID* [ *max-vlan-ID*] ] **inner-cos** *cos-value* ] [ **inner-dei** ]
4. 必要に応じてステップ 3 を繰り返し、計画したシーケンス番号でステートメントを追加します。エントリを削除するには、**no** *sequence-number* コマンドを使用します。
5. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **ethernet-service access-list** *name*

例：

```
RP/0/RSP0/cpu 0: router(config)# ethernet-service access-list L2ACL2
```

イーサネット サービス アクセス リスト コンフィギュレーション モードを開始し、アクセス リスト L2ACL2 を設定します。

**ステップ 3** [ *sequence-number* ] { **permit** | **deny** } { *src-mac-address src-mac-mask* | **any** | **host** } [ { *ethertype-number* } | **vlan** *min-vlan-ID* [ *max-vlan-ID* ] ] [ **cos** *cos-value* ] [ **dei** ] [ **inner-vlan** *min-vlan-ID* [ *max-vlan-ID*] ] **inner-cos** *cos-value* ] [ **inner-dei** ]

例：

```
RP/0/RSP0/cpu 0: router(config-es-al)# 0 permit 1.2.3 3.2.1
or
RP/0/RSP0/cpu 0: router(config-es-al)# 30 deny any dei
```

パケットの通過またはドロップを決定する許可または拒否の条件を 1 つ以上指定します。

**ステップ 4** 必要に応じてステップ 3 を繰り返し、計画したシーケンス番号でステートメントを追加します。エントリを削除するには、**no** *sequence-number* コマンドを使用します。

アクセス リストは変更できます。

**ステップ 5** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

## 次の作業

イーサネット サービス アクセス リストの作成後に、インターフェイスに適用する必要があります。アクセスリストを適用する方法については、「イーサネット サービス アクセス リストの適用」セクションを参照してください。

# イーサネット サービス アクセス リストの適用

作成したアクセス リストを機能させるには、そのアクセス リストを参照する必要があります。アクセスリストは、発信インターフェイスまたは着信インターフェイスに適用できます。ここでは、端末回線とネットワーク インターフェイスの両方に対してこのタスクを実行するためのガイドラインを示します。

着信アクセスリストでは、パケットを受信した後で、Cisco IOS XR ソフトウェアはアクセスリストを参照してパケットの送信元 MAC アドレスをチェックします。アクセス リストがアドレ

スを許可している場合は、パケットの処理を継続します。アクセス リストがアドレスを拒否する場合は、ソフトウェアはパケットを廃棄します。

発信アクセス リストでは、パケットを受信して制御インターフェイスにルーティングした後で、ソフトウェアは、アクセス リストに対してパケットの送信元 MAC アドレスを検査します。アクセス リストがアドレスを許可している場合は、パケットを送信します。アクセス リストがアドレスを拒否する場合は、ソフトウェアはパケットを廃棄します。

（注） 空のアクセス リスト（アクセス コントロール エレメントが含まれていない）は、インターフェイスに適用できません。

## インターフェイスへのアクセスの制御

このタスクでは、アクセス リストをインターフェイスに適用して、そのインターフェイスへのアクセスを制限します。アクセス リストは、発信インターフェイスまたは着信インターフェイスに適用できます。

### 手順の概要

1. **configure**
2. **interface** *type instance*
3. **ethernet-services access-group** *access-list-name* { **ingress** | **egress** }
4. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface** *type instance*

例：

```
RP/0/RSP0/cpu 0: router(config)#  interface gigabitethernet 0/2/0/
```

インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。

- *type* 引数には、インターフェイスタイプを指定します。インターフェイス タイプの詳細については、疑問符（?）オンライン ヘルプ機能を使用してください。
- *instance* 引数には、物理インターフェイス インスタンスまたは仮想インスタンスを指定します。

- 物理インターフェイス インスタンスの表記方法は *rack/slot/module/port* です。値を区切るスラッシュ（/）は、表記の一部として必要です。
- 仮想インターフェイス インスタンスの数値範囲は、インターフェイス タイプによって異なります。

**ステップ 3** **ethernet-services access-group** *access-list-name* { **ingress** | **egress** }

例：

```
RP/0/RSP0/cpu 0: router(config-if)# ethernet-services access-group p-in-filter ingress
RP/0/RSP0/cpu 0: router(config-if)# ethernet-services access-group p-out-filter egress
```

インターフェイスへのアクセスを制御します。

- *access-list-name* 引数を使用して、特定のイーサネット サービス アクセス リストを指定します。
- *ingres* キーワードを使用すると着信パケットをフィルタリングできます。また、*egress* キーワードを使用すると発信パケットをフィルタリングできます。

この例では、GigabitEthernet 0/2/0/2 から発着信されるパケットにフィルタを適用します。

**ステップ 4** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# イーサネット サービス アクセス リストのコピー

このタスクでは、イーサネット サービス アクセス リストをコピーします。

## 手順の概要

1. **copy access-list ethernet-service** *source-acl destination-acl*
2. **show access-lists ethernet-services** [ *access-list-name* | **maximum** | **standby** | **summary** ]

## 手順の詳細

**ステップ 1** **copy access-list ethernet-service** *source-acl destination-acl*

例：

```
RP/0/RSP0/cpu 0: router# copy access-list ethernet-service list-1 list-2
```

既存のイーサネット サービス アクセス リストのコピーを作成します。

- コピーするアクセスリストの名前を指定するには、*source-acl* 引数を使用します。
- 送信元アクセスリストの内容のコピー先を指定するには、*destination-acl* 引数を使用します。
  - *destination-acl* 引数は一意の名前である必要があります。アクセスリストに destination-acl 引数名が存在する場合、そのアクセスリストはコピーされません。

**ステップ 2** **show access-lists ethernet-services** [ *access-list-name* | **maximum** | **standby** | **summary** ]

**例**：

```
RP/0/RSP0/cpu 0: router# show access-lists ethernet-services list-2
```

（任意）指定されたイーサネット サービス アクセス リストの内容を表示します。たとえば、コピー先の内容を検証して、宛先アクセス リスト list-2 に送信元アクセス リスト list-1 の情報がすべて含まれていることを確認できます。

# アクセス リスト エントリの並べ替え

ここでは、名前付きアクセスリストのエントリにシーケンス番号を再割り当てする例を示します。アクセス リストの並べ替えは任意です。

## 手順の概要

1. **resequence access-list ethernet-service** *access-list-name* [ *starting-sequence-number* [ *increment* ] ]
2. **commit** コマンドまたは **end** コマンドを使用します。
3. **show access-lists ethernet-services** [ *access-list-name* | **maximum** | **standby** | **summary** ]

## 手順の詳細

**ステップ 1** **resequence access-list ethernet-service** *access-list-name* [ *starting-sequence-number* [ *increment* ] ]

**例**：

```
RP/0/RSP0/cpu 0: router# resequence access-list ethernet-service L2ACL2 20 10
```

（任意）目的の開始シーケンス番号およびシーケンス番号の増分を使用して、指定されたイーサネット サービス アクセス リストを並べ替えます。

- 次の例では、L2ACL2 という名前のイーサネット サービス アクセス リストを並べ替えます。開始シーケンス番号は 20、増分は 10 です。増分値を選択しないと、デフォルトの増分値 10 が使用されます。

（注） 並べ替えプロセス中に終了番号が許可された最大シーケンス番号を超えることがわかった場合、設定は無効になり、拒否されます。シーケンス番号は変更されません。

**ステップ 2** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

**ステップ 3** **show access-lists ethernet-services** [ *access-list-name* | **maximum** | **standby** | **summary** ]

**例**：

```
RP/0/RSP0/cpu 0: router# show access-lists ethernet-services L2ACL2
```

（任意）指定されたイーサネット サービス アクセス リストの内容を表示します。

- 出力をレビューして、アクセス リストに最新情報が含まれていることを確認します。

# レイヤ 2 アクセス リストを実装するための設定例

## アクセス リストのエントリの並べ替え：例

次に、アクセスリストの並べ替え例を示します。並べ替えられたアクセスリストの先頭の値は 1、増分値は 2 です。後続のエントリはユーザ指定の増分値に基づいて並べられています。範囲は 1 ～ 2147483646 です。

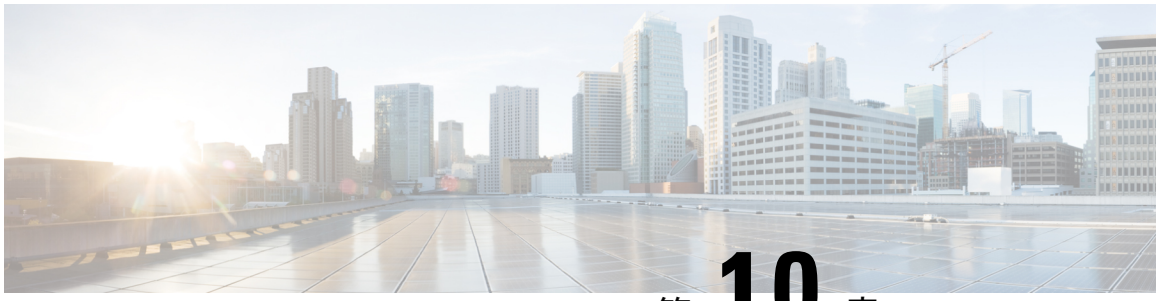
シーケンス番号のないエントリが入力されると、デフォルトで、アクセスリストの最後のエントリのシーケンス番号に 10 を加えたシーケンス番号が割り当てられます。

```
ethernet service access-list acl_1
10 permit 1.2.3 4.5.6
20 deny 2.3.4 5.4.3
30 permit 3.1.2 5.3.4 cos 5

resequence access-list ethernet service acl_1 10 20

show access-list ethernet-service acl1_1

ipv4 access-list acl_1
  10 permit 1.2.3 4.5.6
  30 deny 2.3.4 5.4.3
  50 permit 3.1.2 5.3.4 cos 5
```

## シーケンス番号を指定したエントリの追加：例

この例では、新しいエントリをイーサネット サービス アクセス リスト acl_5 に追加します。

```
ethernet-service access-list acl_5
2 permit 1.2.3 5.4.3
5 permit 2.3.4. 6.5.4 cos 3
10 permit any dei
20 permit 6.5.4 1.3.5 VLAN vlan3

configure
 ethernet-service access-list acl_5
 15 permit 1.5.7 7.5.1
 end

ethernet-service access-list acl_5
2 permit 1.2.3 5.4.3
5 permit 2.3.4. 6.5.4 cos 3
10 permit any dei
15 permit 1.5.7 7.5.1
20 permit 6.5.4 1.3.5 VLAN vlan3
```



第 10 章

# VXLAN の実装

このモジュールでは、一般的な VXLAN の概念情報と、Cisco ASR 9000 シリーズ ルータでのレイヤ 2 VXLAN の設定情報を示します。レイヤ 3 VXLAN の設定情報については、『Cisco ASR 9000 Series Aggregation Services Router MPLS Layer 3 VPN Configuration Guide』の「Implementing L3 VXLAN」の章を参照してください。VXLAN は、VLAN の場合と同じイーサネットレイヤ 2 ネットワークサービスを提供しますが、より優れた拡張性と柔軟性を備えています。

**表 *4 : VXLAN* の機能の履歴**

| リリース | 変更内容 |
|---|---|
| リリース 5.2.0 | この機能は、Cisco ASR 9000 シリーズ ルータで導入されました。 |
| リリース 5.3.1 | VXLAN エニーキャストゲートウェイ機能が導入されました。 |
| リリース 6.1.2 | 次の機能が追加されました。<br>• EVPN VXLAN レイヤ 2 Data Center Interconnect ゲートウェイ<br>• EVPN ESI マルチパス |

# VXLAN の実装の前提条件

この前提条件は、VXLAN の実装に適用されます。

適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。

ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

# VXLAN の実装に関する情報

VXLAN を実装するには、次の概念を理解している必要があります。

## VXLAN

VXLAN は、VLAN の場合と同じイーサネットレイヤ 2 ネットワークサービスを提供しますが、より優れた拡張性と柔軟性を備えています。VXLAN は、レイヤ 3 ネットワーク上のレイヤ 2 オーバーレイ方式です。VXLAN は MAC Address-in-User Datagram Protocol（MAC-in-UDP）のカプセル化を使用して、コアネットワークでレイヤ 2 セグメントを拡張する方法を提供します。VXLAN は、共有される共通の物理インフラストラクチャにおいて、柔軟で大規模なマルチテナント環境をサポートするためのソリューションです。コアネットワークでの転送プロトコルは IP と UDP です。VLAN と比較して、VXLAN には次の利点があります。

- データセンター全体にマルチテナントセグメントを柔軟に配置します。テナントのワークロードがデータセンター内の物理ポッド全域に配置されるように、基盤となる共有ネットワーク インフラストラクチャでレイヤ 2 セグメントを拡張するソリューションを提供します。
- より多くのレイヤ 2 セグメントをアドレス指定するための拡張性が高くなります。VLAN は 12 ビットの VLAN ID を使用してレイヤ 2 セグメントをアドレス指定します。このため、拡張性は制限され VLAN の数は最大 4094 個になります。VXLAN は、VXLAN ネットワーク識別子（VNID）と呼ばれる 24 ビットのセグメント ID を使用します。これにより、最大 1600 万の VXLAN セグメントを同じ管理ドメインに共存させることができます。
- 基盤となるインフラストラクチャで使用可能なネットワークパスの使用率が向上します。VLAN はループ防止のためにスパニングツリープロトコルを使用します。このため、冗長パスをブロックすることによってネットワーク内の半数のネットワークリンクを使用しません。一方、VXLAN パケットはレイヤ 3 ヘッダーに基づいて基盤となるネットワーク経由で転送されます。VXLAN では、レイヤ 3 ルーティング、Equal Cost Multipath（ECMP; 等コストマルチパス）ルーティング、およびリンク アグリゲーション プロトコルを活用して、すべての利用可能なパスを使用できます。

# VXLAN エニーキャストゲートウェイ

VXLAN エニーキャストゲートウェイ機能は、エニーキャスト機能を VXLAN に拡張します。これにより、アンダーレイ マルチキャスト ロードバランシングおよび冗長性のためにネットワーク上でエニーキャストルーティングを使用できるようになります。

VXLAN エニーキャスト ソリューションは次のとおりです。

- 完全なアクティブ-アクティブ ファーストホップ ゲートウェイを許可します（フロー単位でアクティブ-アクティブ）。
- 新しいコントロール プレーン プロトコルや管理プレーンプロトコル、またはどのような形式の外部 SDN コントローラや NMS もゲートウェイの調整や同期を行いません。

エニーキャストゲートウェイ機能は、次の基本的な概念に従います。

- 複数の VXLAN ゲートウェイ間で仮想レイヤ 3 ゲートウェイと仮想 VTEP を作成する。これらのゲートウェイは、オーバーレイ IP アドレス、オーバーレイ MAC アドレス、およびアンダーレイ VTEP IP アドレスと同じ設定を使用します。
- 特定のタイプのオーバーレイ制御パケットのデータプレーンミラーとして使用する、ゲートウェイ間のプライベート マルチキャスト グループを作成する。

（注） VXLAN エニーキャストゲートウェイ機能は、Cisco ASR 9000 高密度 100GE イーサネットラインカードのみでサポートされます。

### 推奨事項

VXLAN エニーキャストゲートウェイ機能を設定する前に、次の推奨事項を考慮する必要があります。

- BGP は、データセンター内の VXLAN エニーキャスト機能とは連動しません。
- IGP は、データセンター内のアンダーレイネットワークで動作します。
- BGP および IGP は、WAN 側で使用する必要があります。
- データセンターのトップオブラック（TOR）スイッチは、ルータのカスタマー IP とエニーキャストゲートウェイ間のスタティックルートを使用します。

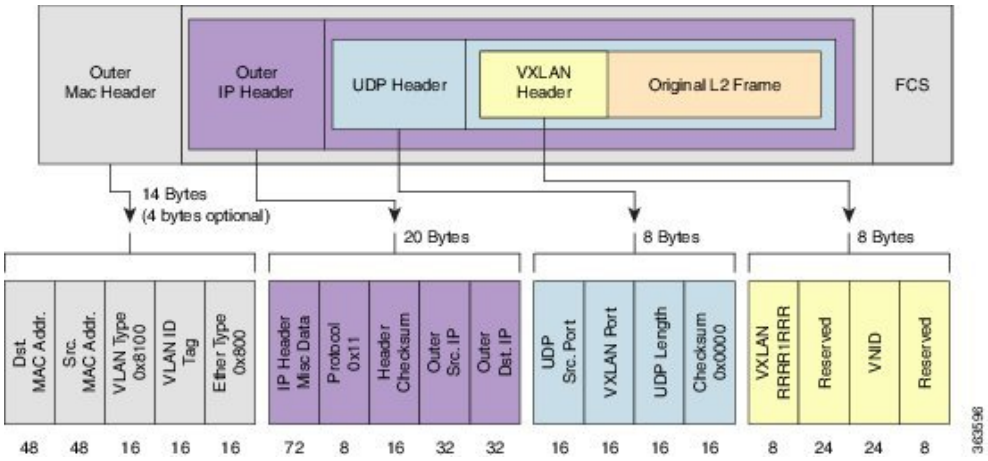
### VxLAN エニーキャストゲートウェイを展開するための要件

マルチキャストグループは制御フレームのミラーリングに使用されるため、IPv6 ネイバーアドバタイズメントの場合は、2 つのルータ（またはインターフェイス）間で同じアドレスが検出されることにより、重複アドレス検出（DAD）プロトコルがサービスをダウンさせます。したがって、BVI インターフェイスで IPv6 DAD を無効にし、不要ノード検出（ND）応答を有効にする必要があります。

# VXLAN のパケット形式

VXLAN は MAC-in-UDP のカプセル化方式を定義します。この方式において、元のレイヤ 2 フレームに VXLAN ヘッダーが追加され、UDP-IP パケットに置かれます。この MAC-in-UDP のカプセル化によって、VXLAN はレイヤ 3 ネットワーク上でレイヤ 2 ネットワークをトンネルします。VXLAN のパケット形式を次の図に示します。

**図 *65 : VXLAN* のパケット形式**



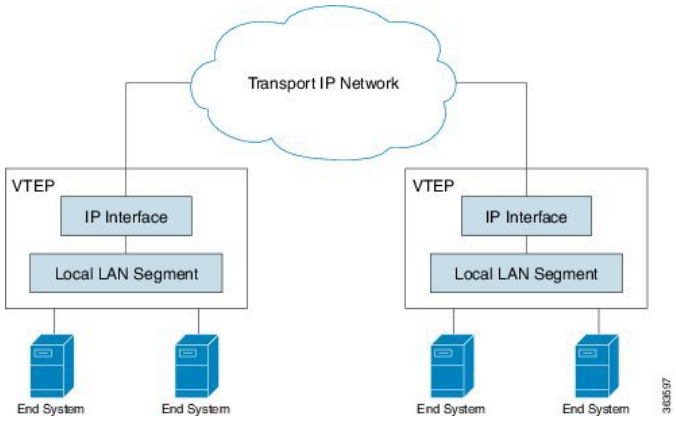

上図に示すように、VXLAN は 24 ビット VNID といくつかの予約ビットで構成される 8 バイト VXLAN ヘッダーを導入します。VXLAN ヘッダーおよび元のイーサネット フレームは、UDP ペイロードに入ります。24 ビット VNID は、レイヤ 2 セグメントを識別し、セグメント間でレイヤ 2 の分離を維持するために使用されます。VNID のすべての 24 ビットを使用して、VXLAN は約 1600 万個の LAN セグメントをサポートできます。

# VXLAN トンネル エンドポイント

VXLAN は VXLAN トンネルエンドポイント（VTEP）デバイスを使用してテナントのエンドデバイスを VXLAN セグメントへマッピングし、VXLAN のカプセル化およびカプセル化解除を実行します。各 VTEP 機能には 2 つのインターフェイスがあります。1 つはブリッジングを介してローカル エンドポイントの通信をサポートするためのローカル LAN セグメント上のスイッチ インターフェイスで、もう 1 つは、転送 IP ネットワークのための IP インターフェイスです。

IP インターフェイスには一意の IP アドレスがあります。これは、インフラストラクチャ VLAN として知られる、転送 IP ネットワーク上の VTEP を識別します。VTEP デバイスはこの IP アドレスを使用してイーサネット フレームをカプセル化し、カプセル化されたパケットを、IP インターフェイスを介して転送ネットワークへ送信します。また、VTEP デバイスはリモート VTEP で VXLAN セグメントを検出し、IP インターフェイスを介してリモートの MAC Address-to-VTEP マッピングについて学習します。次の図に、VTEP の機能コンポーネントとトランスポート IP ネットワークを介したレイヤ 2 接続用に作成された論理トポロジを示します。

図 *66 : VTEP*




VXLAN セグメントは基盤となるネットワーク トポロジに依存しません。逆に、VTEP 間の基盤となる IP ネットワークは、VXLAN オーバーレイに依存しません。これは送信元 IP アドレスとして開始 VTEP を持ち、宛先 IP アドレスとして終端 VTEP を持っており、外部 IP アドレス ヘッダーに基づいてパケットをカプセル化します。

# レイヤ 2 VXLAN ゲートウェイの設定

レイヤ 2 VXLAN ゲートウェイは、同じレイヤ 2 ネットワーク内の VXLAN セグメントと非 VXLAN セグメント（VLAN や VPLS など）の間のトラフィックをブリッジします。VXLAN レイヤ 2 ゲートウェイの動作は、データプレーン MAC アドレスラーニングと、IP マルチキャストによるマルチデスティネーショントラフィックのフラッディング（未知のユニキャスト、マルチキャスト、ブロードキャストフレームなど）に基づいています。次のセクションでは、ASR 9000 シリーズ ルータを、同じ L2 ドメイン内の VLAN および VXLAN セグメント間のレイヤ 2 VXLAN ゲートウェイとして設定する方法を示します。

## 前提条件

VXLAN レイヤ 2 ゲートウェイとして Cisco ASR 9000 シリーズ ルータを設定するための前提条件を、次に示します。

- ループバック インターフェイスを設定します。これは、ローカル VTEP の送信元インターフェイスとして機能します。
- リモート VTEP へのユニキャストの到達可能性を設定します。
- Bidirectional Protocol Independent Multicast（Bidir PIM）または PIM スパースモードを設定します。詳細については、 *Multicast Configuration Guide for Cisco ASR 9000 Series Routers*を参照してください。

# 機能制限

VXLAN の設定時には、次の制限事項を考慮してください。

- VXLAN は、オーバーレイトランスポート仮想化（OTV）および VXLAN UDP ポートでのみ設定します。
- ループバック インターフェイスのみを送信元インターフェイスにできます。
- 複数の NVE インターフェイス間で VNI、マルチキャストグループ、またはソースインターフェイスを共有することはできません。
- VNI 範囲とマルチキャスト範囲は、どちらも連続した範囲のみを指定できます。カンマ区切り値を使用した連続していない範囲はサポートされていません。
- マルチキャストグループへの VNI のマッピングには、1:1 または N:1 のいずれかを使用できます。次に例を示します。
  - 「member vni 5000 mcast-group 239.1.1.1」コマンドは、有効な 1:1 のマッピングを設定します。
  - 「member vni 5000-5005 mcast-group 239.1.1.1」コマンドは、有効な N:1 のマッピングを設定します。
- VNI が VNI 範囲の一部として設定されている場合は、同じ範囲の一部としてのみ変更または削除できます。たとえば、「member vni 5000-5002 mcast-group 239.1.1.1」コマンドが設定されている場合は、「no member vni 5001」コマンドを使用して NVE インターフェイスから VNI 5001 の関連付けのみを解除することはできません。
- スタティック MAC 設定はサポートされていません。
- システムごとに最大 128k のレイヤ 2 およびレイヤ 3 サブインターフェイスを設定できます。この設定には、レイヤ 2 サブインターフェイスとレイヤ 3 サブインターフェイス両方の組み合わせを使用できます。または、すべてレイヤ 2 サブインターフェイスにすることや、すべてレイヤ 3 サブインターフェイスにすることもできます。

  システムでは、システムごとに 128k を超えるサブインターフェイスを設定できますが、サービスにこの設定を使用することはできません。128k のサブインターフェイスのしきい値に達すると、システムにより警告メッセージが表示されますが、設定は引き続き適用されます。ただし、サービスにこの設定を使用することはできません。

# ネットワーク仮想化エンドポイント（NVE）インターフェイスの作成と設定

NVE インターフェイスを作成し、VXLAN の VXLAN トンネルエンドポイント（VTEP）として設定するには、次の作業を実行します。

### 手順の概要

1. **interface nve** *nve-identifier*
2. （オプション） **overlay-encapsulation vxlan**
3. **source-interface loopback** *loopback-interface-identifier*
4. **member vni** *vni_number* [ *-end_vni_range* ] **mcast-group** *ip_address* [ *end_ip_address_range* ]
5. （オプション）**anycast source-interface loopback** *loopback-interface-identifier* **sync-group** *ip_address*
6. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **interface nve** *nve-identifier*

例：

```
RP/0/RSP0/cpu 0: router(config)# interface nve 1
```

NVE インターフェイスを作成し、NVE インターフェイス設定サブモードを開始します。

**ステップ 2** （オプション） **overlay-encapsulation vxlan**

例：

```
RP/0/RSP0/cpu 0: router(config-if)# overlay-encapsulation vxlan
```

NVE インターフェイスの VXLAN カプセル化を設定します。VXLAN は、NVE インターフェイスのデフォルトのカプセル化です。この手順は、カプセル化を変更していない場合はオプションです。

**ステップ 3** **source-interface loopback** *loopback-interface-identifier*

例：

```
RP/0/RSP0/cpu 0: router(config-if)# source-interface loopback 1
```

ループバック インターフェイスを VTEP の送信元インターフェイスとして設定します。

**ステップ 4** **member vni** *vni_number* [ *-end_vni_range* ] **mcast-group** *ip_address* [ *end_ip_address_range* ]

例：

```
RP/0/RSP0/cpu 0: router(config-if)# member vni 1-10 mcast-group 224.2.2.2 224.2.2.10
```

VXLAN ネットワーク識別子（VNI）を使用して単一の VXLAN または連続する範囲の VXLAN を NVE インターフェイスに関連付け、この VNI に関連付けられるマルチキャストアドレスまたは連続するマルチキャストアドレス範囲を指定します。

（注）
- VNI とマルチキャストグループとのマッピングは、1 対 1 または多対 1 です。
- 不連続な VXLAN または VXLAN 範囲を NVE インターフェイスに関連付けるには、VXLAN または VXLAN 範囲ごとに次の手順を実行します。たとえば、

```
RP/0/RSP0/cpu 0: router(config-if)# member vni 10 mcast-group 224.2.2.10
RP/0/RSP0/cpu 0: router(config-if)# member vni 23 mcast-group 224.2.2.23
RP/0/RSP0/cpu 0: router(config-if)# member vni 50-59 mcast-group 224.2.2.50 224.2.2.59
RP/0/RSP0/cpu 0: router(config-if)# member vni 100-120 mcast-group 224.2.2.100 224.2.2.120
```

**ステップ 5** （オプション）**anycast source-interface loopback** *loopback-interface-identifier* **sync-group** *ip_address*

例：

```
RP/0/RSP0/cpu 0: router(config-if)# anycast source-interface loopback 1 sync-group 192.23.2.20
```

この VTEP のエニーキャスト モード パラメータを設定します。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

### 次のタスク

設定された NVE インターフェイス情報を表示するには、**show nve interface** コマンドを使用します。

# レイヤ 2 サブインターフェイスの作成と設定

VLAN セグメントに関連付けられたレイヤ 2 サブインターフェイスを作成するには、次の作業を実行します。

### 手順の概要

1. **interface gigabitEthernet** *interface-identifier* **l2transport**
2. **dot1q vlan** *vlan-identifier*
3. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **interface gigabitEthernet** *interface-identifier* **l2transport**

例：

```
RP/0/RSP0/cpu 0: router(config)# interface gigabitEthernet 0/0/0/0.100 l2transport
```

レイヤ 2 サブインターフェイスを作成し、サブインターフェイス設定モードを開始します。

**ステップ 2** **dot1q vlan** *vlan-identifier*

例：

```
RP/0/RSP0/cpu 0: router(config-if)# dot1q vlan 100
```

インターフェイスの VLAN を設定します。

**ステップ 3** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# VLAN および VXLAN のブリッジドメインへの関連付け

VLAN および VXLAN をブリッジドメインに関連付けるには、次の作業を実行します。

## 手順の概要

1. **l2vpn**
2. **bridge group** *bridge-group-name*
3. **bridge-domain** *bridge-domain-name*
4. **member vni** *vxlan-identifier*
5. **interface gigabitEthernet** *sub-interface-identifier*
6. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

l2vpn コンフィギュレーション モードを開始します。

**ステップ 2** **bridge group** *bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group bridgegroup1
```

ブリッジグループ設定モードを開始します。

**ステップ 3** **bridge-domain** *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain bdomain1
```

ブリッジドメイン設定モードを開始します。

**ステップ 4** **member vni** *vxlan-identifier*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# member vni 100
```

VXLAN をブリッジドメインに関連付けます。

**ステップ 5** **interface gigabitEthernet** *sub-interface-identifier*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# interface gigabitEthernet 0/0/0/0.200
```

VLAN サブインターフェイスを使用して、VLAN をブリッジドメインに関連付けます。

**ステップ 6** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# VXLAN 送信元 UDP ポートの設定

これはオプションのタスクです。デフォルトでは、カプセル化 VXLAN セグメントの送信元 UDP ポートは、内部ペイロードのレイヤ 2 アドレスフィールドのハッシュ関数によって計算されます。内部ペイロードのレイヤ 2 またはレイヤ 3 アドレスフィールドのいずれかで実行されるハッシュ関数を設定するには、次の作業を実行します。

## 手順の概要

1. **l2vpn**
2. **load-balancing flow** [ *src-dst-mac* | *src-dst-ip* ]

## 手順の詳細

**ステップ 1** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

l2vpn コンフィギュレーション モードを開始します。

**ステップ 2** **load-balancing flow** [ *src-dst-mac* | *src-dst-ip* ]

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# load-balancing flow src-dst-mac
```

ハッシュ関数用に内部ペイロードのレイヤ 2 またはレイヤ 3 アドレスフィールドのいずれかを選択します。

# VXLAN 宛先 UDP ポートの設定

UDP ポート番号 4789 と 8472 はそれぞれ VXLAN と OTV に割り当てられます。カプセル化 VXLAN セグメントの宛先 UDP ポート番号を設定するには、次の作業を実行します。デフォルトでは、カプセル化 VXLAN データグラムの宛先 UDP ポート番号が 4789 に設定されているため、これはオプションのタスクです。宛先 VTEP が OTV ポートを使用して VXLAN をサポートしている場合は、宛先 UDP ポート番号を 8472 に設定する必要があります。

## 手順の概要

1. **vxlan udp port** *port-number*

## 手順の詳細

**vxlan udp port** *port-number*

例：

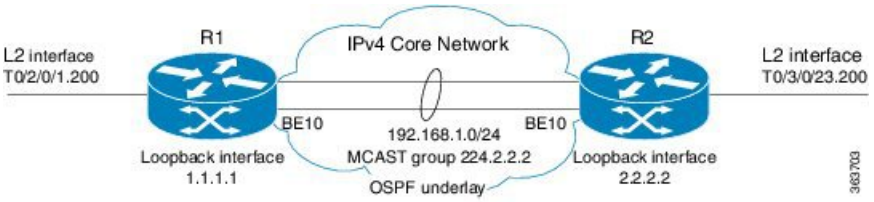
```
RP/0/RSP0/cpu 0: router(config)# vxlan udp port 4789
```

カプセル化 VXLAN セグメントの宛先 UDP ポート番号を設定します。

# レイヤ 2 VXLAN ゲートウェイの実装の設定例

次の例は、PE ルータ間のバンドルリンク接続としてコアネットワークが簡素化されたサンプル ネットワーク トポロジの、2 台のプロバイダーエッジ（PE）ルータ（R1 および R2）でのレイヤ 2 VXLAN ゲートウェイ設定を示しています。

**図 *67 :* レイヤ *2 VXLAN* ゲートウェイを使用するネットワーク**




**R1 での設定：**

```
interface Bundle-Ether10
  ipv4 address 192.168.1.1/24
!
interface Loopback0
  ipv4 address 1.1.1.1/32
!
interface T0/2/0/1
 no shut
!
interface T0/2/0/1.200 l2transport
 encapsulation dot1q 200
```

```
!
router ospf underlay
 router-id 1.1.1.1
 area 0
  interface Bundle-Ether10
  interface Loopback0
!
interface nve 1
  member vni 1 mcast-group 224.2.2.2 0.0.0.0
  overlay-encapsulation vxlan
  source-interface Loopback0
!
l2vpn
 bridge group vxlan
  bridge-domain vxlan
    interface T0/2/0/1.200
    member vni 1
  !
multicast-routing
 address-family ipv4
  interface Loopback0
    enable
  interface Bundle-Ether10
     enable
!
router pim
 address-family ipv4
  rp-address 1.1.1.1 bidir
```

**R2 での設定**：

```
interface Bundle-Ether10
  ipv4 address 192.168.1.2/24
!
interface Loopback0
  ipv4 address 2.2.2.2/32
!
interface T0/3/0/23
 no shut
!
interface T0/3/0/23.200 l2transport
 encapsulation dot1q 200
!
router ospf underlay
 router-id 2.2.2.2
 area 0
  interface Bundle-Ether10
  interface Loopback0
!
Interface nve 1
  member vni 1 mcast-group 224.2.2.2 0.0.0.0
  overlay-encapsulation vxlan
  source-interface Loopback0
!
l2vpn
 bridge group vxlan
  bridge-domain vxlan
    interface T0/3/0/23.200
    member vni 1
  !
multicast-routing
 address-family ipv4
  interface Loopback0
    enable
  interface Bundle-Ether10
```

```
     enable
!
router pim
 address-family ipv4
  rp-address 1.1.1.1 bidir
```

# EVPN VXLAN レイヤ 2 Data Center Interconnect ゲートウェイ

Cisco ASR 9000 シリーズ ルータはデータセンター相互接続（DCI）レイヤ 2 ゲートウェイとして機能し、MPLS ベースの L2VPN ネットワークを介して EVPN VXLAN ベースのデータセンター間にレイヤ 2 接続を提供します。データセンターは、中間サービス プロバイダー ネットワークを通じて接続されます。EVPN VXLAN 対応データセンターは、EVPN コントロール プレーンを使用して、1 つのデータセンターから別のデータセンターへのレイヤ 2 転送情報を配信します。この機能によって冗長性、復元力、プロビジョニング簡便性が得られます。

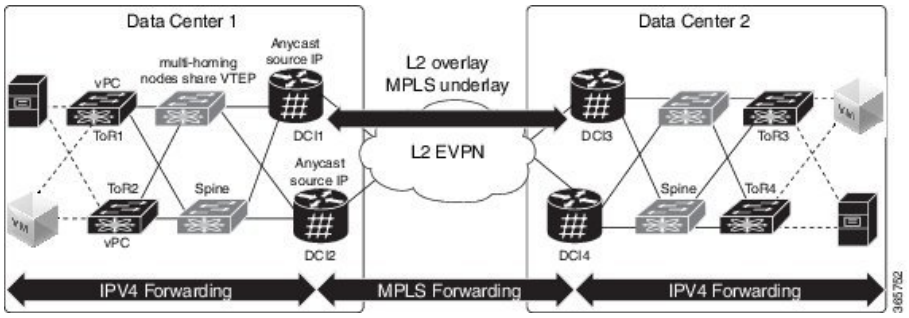
EVPN VXLAN レイヤ 2 DCI ゲートウェイ機能は次の機能をサポートしています。

- シングル ホーミングでの VXLAN アクセス
- エニーキャスト VXLAN 端末エンドポイント（VTEP）IP アドレスを使用したオールアクティブ マルチホーミングでの VXLAN アクセス
- 一意の VTEP IP アドレスを使用したオールアクティブ マルチホーミングでの VXLAN アクセス
- VXLAN カプセル化を使用した EVPN ESI マルチパス

## エニーキャスト VTEP IP アドレスを使用したオールアクティブ マルチホーミング

DCI は エニーキャスト VTEP IP アドレスを使用したオールアクティブ マルチホーミングに同じエニーキャスト VTE IP アドレスを使用します。Top of Rack（ToR）が複数のパスを使用して DCI に接続されており、トラフィックは ToR から DCI に複数の物理パスを通じて渡され、ロードバランシングにエニーキャスト IP アドレスが使用されているトポロジを考えてみます。DCI1 と DCI2 は、ネクストホップと同じエニーキャスト IP アドレスを使用して MAC ルートを ToR にアドバタイズします。つまり、ToR は DCI の同じエニーキャスト IP アドレスにトラフィックを送信し、ロードバランシングに IGP ECMP を使用します。仮想 PortChannel（vPC）では、ToR1 と ToR2 で同じ IP 設定を使用できます。ToR1 と ToR2 は、ネクストホップと同じ IP アドレスを使用して MAC ルートを DCI にアドバタイズします。そのため、DCI は ToR の同じ IP アドレスにトラフィックを送信し、ロード バランシングに IGP ECMP を使用します。DCI は、MPLS 転送を通じてリモート データセンターにトラフィックを送信します。

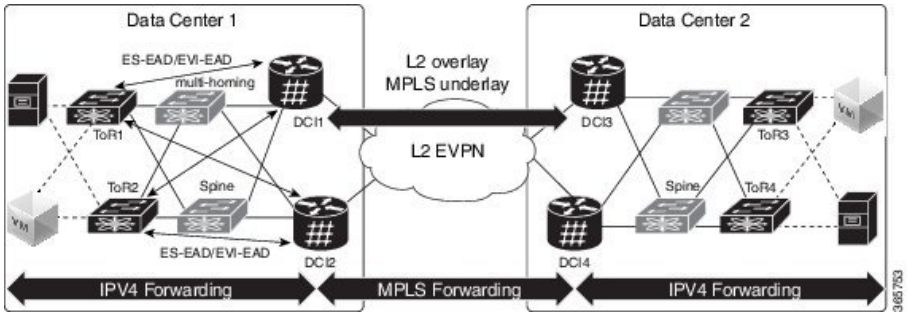
図 *68 :* エニーキャスト *VTEP IP* アドレスを使用したオールアクティブ マルチホーミング




# 一意の VTEP IP アドレスを使用したオールアクティブ マルチホーミング

DCI ではオールアクティブ マルチホーミングのエニーキャスト VTEP IP アドレスを一意の VTEP IP アドレスと共有しません。各 DCI は一意の VTEP IP アドレスを使用します。ToR が DCI から MAC ルートを受け取る次のトポロジを考えてみます。各 MAC ルートには一意のネクストホップがあります。DCI1 と DCI2 は両方とも異なるネクストホップを持つ同じ MAC のルートをアドバタイズするため、ToR には同じ MAC に 2 つの等コスト ネクストホップがあります。ToR は MAC にトラフィックを送信すると、ToR は両方のネクストホップ上でトラフィックのロード バランシングを実行します。

図 *69 :* 一意の *VTEP IP* アドレスを使用したオールアクティブ マルチホーミング

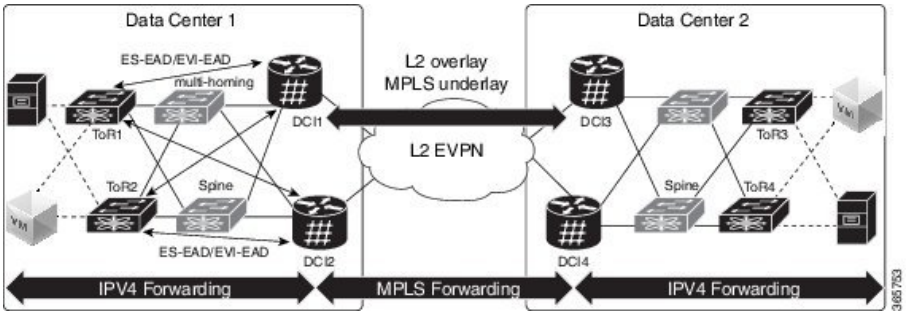



# VXLAN の EVPN ESI マルチパス：EVI ベースのロード バランシング

EVPN イーサネット セグメント識別子（ESI）マルチパス機能は、アクティブ-アクティブのデュアルホーム接続 ToR と DCI へのマルチパス トラフィックをサポートし、データセンター内に冗長接続を実現します。ESI マルチパスは、EVPN シグナリングを通じて ASR9k DCI ルータによって検出されます。パスは、イーサネット セグメント識別子（ESI）と EVPN インスタンス（EVI）に基づいて選択されます。受信した MAC ルートのパスを解決するには、RFC 7432 に指定されているとおり、ES ごとにイーサネット A-D ルート（ES-EAD）を、EVI ごとにイーサネット A-D（EVI-EAD）を使用します。

DCI が ToR から MAC ルートを受信し、各 MAC ルートに各 ToR のネクストホップがある次のトポロジを考えてみます。同様に、DCI は ToR へのさまざまなネクストホップを使用して MAC ルートをアドバタイズします。ToR のペアの背後にある VM へ DCI がトラフィックを送信する場合は、すべての MAC に 2 つのパスが存在します。DCI は、2 つのパス上でトラフィックをロード バランスします。パスの選択は、EVI に基づいています。たとえば、DCI1 と DCI2 は EVI1 で学習した MAC アドレス宛のすべてのトラフィックに ToR1 を選択します。また、DCI1 と DCI2 は EVI2 で学習した MAC アドレス宛のすべてのトラフィックに ToR2 を選択します。

**図 *70 : EVPN ESI* マルチパス**




# EVPN VXLAN レイヤ 2 Data Center Interconnect ゲートウェイの設定

EVPN VXLAN レイヤ 2 Data Center Interconnect ゲートウェイを設定するには、次のタスクを実行します。

EVPN ESI マルチパス機能を設定する場合は、エニーキャスト IP アドレスは設定しないでください。残りの設定タスクは同じです。

## BGP ルーティング プロセスでの L2 EVPN アドレス ファミリの設定

BGP ルーティング プロセスで EVPN アドレス ファミリを有効にするには、次のタスクを実行します。

### 手順の概要

1. **configure**
2. **router bgp** *asn_id*
3. **nsr**
4. **bgp graceful-restart**
5. **bgp router-id***ip-address*
6. **address-family l2vpn evpn**
7. **commit** コマンドまたは **end** コマンドを使用します。

手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **router bgp** *asn_id*

例：

```
RP/0/RSP0/cpu 0: router(config)# router bgp 100
```

BGP AS 番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。

**ステップ 3** **nsr**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# nsr
```

ノンストップルーティングを有効にします。

**ステップ 4** **bgp graceful-restart**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# bgp graceful-restart
```

ルータのグレースフル リスタートをイネーブルにします。

**ステップ 5** **bgp router-id***ip-address*

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# bgp router-id 209.165.200.227
```

指定したルータ ID で、ルータを設定します。

**ステップ 6** **address-family l2vpn evpn**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# address-fmaily l2vpn evpn
```

BGP ルーティングプロセスでグローバルに EVPN アドレスファミリを有効にし、EVPN アドレスファミリ設定サブモードを開始します。

**ステップ 7** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。

• [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# DCI と ToR 間のルーティング セッションの設定

DCI と ToR 間のルーティングセッションを設定するには、次の作業を実行します。

## 手順の概要

1. **configure**
2. **router bgp** *asn_id*
3. **neighbor***ip-address*
4. **remote-as** *autonomous-system-number*
5. **ebgp-multihop** *maximum hop count*
6. **update-source** *loopback*
7. **address-family l2vpn evpn**
8. **import stitching-rt reoriginate**
9. **route-policy** *route-policy-name* **in**
10. **encapsulation-type** *type*
11. **route-policy** *route-policy-name* **out**
12. **advertise l2vpn evpn re-originated stitching-rt**
13. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **router bgp** *asn_id*

例：

```
RP/0/RSP0/cpu 0: router(config)# router bgp 100
```

BGP AS 番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。

**ステップ 3** **neighbor***ip-address*

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 209.165.200.225
```

ルータを BGP ルーティングのネイバー設定モードにして、ネイバーの IP アドレス 209.165.200.225 を BGP ピアとして設定します。

**ステップ 4** **remote-as** *autonomous-system-number*

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 2000
```

ネイバーを作成し、そのネイバーをリモート自律システム番号に割り当てます。

**ステップ 5** **ebgp-multihop** *maximum hop count*

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# ebgp-multihop 255
```

外部 BGP ネイバーとのマルチホップ ピアリングをイネーブルにします。

**ステップ 6** **update-source** *loopback*

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# update-source loopback1
```

BGP セッションが、特定のインターフェイスのプライマリ IP アドレスをローカル アドレスとして使用できるようにします。

**ステップ 7** **address-family l2vpn evpn**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-fmaily l2vpn evpn
```

EVPN アドレスファミリを設定します。

**ステップ 8** **import stitching-rt reoriginate**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# import stitching-rt reoriginate
```

スティッチング ルート ターゲット識別子と一致するルートターゲット識別子を持つ BGP EVPN NLRI からのルーティング情報のインポートを有効にし、この再発信後のルーティング情報を L2VPN BGP ネイバーにエクスポートします。

**ステップ 9** **route-policy** *route-policy-name* **in**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# route-policy pass-all in
```

着信ユニキャストルートにルートポリシーを適用します。

**ステップ 10** **encapsulation-type** *type*

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# encapsulation-type vxlan
```

カプセル化タイプとして VXLAN を設定します。

**ステップ 11** **route-policy** *route-policy-name* **out**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# route-policy pass-all out
```

発信ユニキャストルートにルートポリシーを適用します。

**ステップ 12** **advertise l2vpn evpn re-originated stitching-rt**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# advertise l2vpn evpn re-originated stitching-rt
```

L2VPN BGP ネイバーから受信する L2VPN EVPN ルートのアドバタイズメントを設定します。

**ステップ 13** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずに設定セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# リモート DCI 接続の BGP セッションの設定

リモート DCI 接続に BGP セッションを設定するには、次のタスクを実行します。

## 手順の概要

1. **configure**
2. **router bgp** *asn_id*
3. **neighbor***ip-address*
4. **remote-as** *autonomous-system-number*
5. **update-source** *loopback*
6. **address-family l2vpn evpn**
7. **import re-originate stitching-rt**
8. **advertise l2vpn evpn re-originated**
9. **commit** コマンドまたは **end** コマンドを使用します。

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **router bgp** *asn_id*

例：

```
RP/0/RSP0/cpu 0: router(config)# router bgp 200
```

BGP AS 番号を指定し、BGP コンフィギュレーション モードを開始します。このモードでは、BGP ルーティング プロセスを設定できます。

**ステップ 3** **neighbor***ip-address*

例：

```
RP/0/RSP0/cpu 0: router(config-bgp)# neighbor 209.165.201.1
```

ルータを BGP ルーティングのネイバー設定モードにして、ネイバーの IP アドレス 209.165.201.1 を BGP ピアとして設定します。

**ステップ 4** **remote-as** *autonomous-system-number*

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# remote-as 100
```

ネイバーを作成し、そのネイバーをリモート自律システム番号に割り当てます。

**ステップ 5** **update-source** *loopback*

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# update-source loopback2
```

BGP セッションが、特定のインターフェイスのプライマリ IP アドレスをローカル アドレスとして使用できるようにします。

**ステップ 6** **address-family l2vpn evpn**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr)# address-fmaily l2vpn evpn
```

EVPN アドレスファミリを設定します。

**ステップ 7** **import re-originate stitching-rt**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# import re-originate stitching-rt
```

スティッチング ルート ターゲット識別子と一致するルートターゲット識別子を持つ BGP EVPN NLRI からのルーティング情報のインポートを有効にし、この再発信後のルーティング情報を L2VPN BGP ネイバーにエクスポートします。

**ステップ 8** **advertise l2vpn evpn re-originated**

例：

```
RP/0/RSP0/cpu 0: router(config-bgp-nbr-af)# advertise l2vpn evpn re-originated
```

L2VPN BGP ネイバーから受信する L2VPN EVPN ルートのアドバタイズメントを設定します。

**ステップ 9** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずに設定セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# ネットワーク仮想化エンドポイント（NVE）インターフェイスの設定

VNE インターフェイスを作成し、VxLAN の VXLAN トンネル エンドポイント（VTEP）として設定します。

## 手順の概要

1. **configure**
2. **interface nve** *nve-identifier*
3. **source-interface loopback** *loopback-interface-identifier*
4. **anycast source-interface loopback** *loopback-interface-identifier*
5. **redundancy**
6. **backbone vxlan**
7. **iccp group** *group number*
8. **exit**
9. **backbone mpls**
10. **iccp group** *group number*
11. **exit**
12. **exit**
13. **member vni** *vni_number*
14. **load-balance per-evi**
15. **suppress-unknown-unicast-flooding**
16. **mcast-group** *ip_address*
17. **host-reachability protocol** *protocol*
18. **commit** または **end** コマンドを使用します

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **interface nve** *nve-identifier*

例：

```
RP/0/RSP0/cpu 0: router(config)# interface nve 1
```

NVE インターフェイスを作成し、NVE インターフェイス設定サブモードを開始します。

**ステップ 3** **source-interface loopback** *loopback-interface-identifier*

例：

```
RP/0/RSP0/cpu 0: router(config-if)# source-interface loopback 1
```

ループバック インターフェイスを VTEP の送信元インターフェイスとして設定します。

**ステップ 4** **anycast source-interface loopback** *loopback-interface-identifier*

例：

```
RP/0/RSP0/cpu 0: router(config-if)# anycast source-interface loopback 1
```

エニーキャストモードのパラメータと、エニーキャストモードの送信元インターフェイスを設定します。

エニーキャスト IP アドレスは、ファブリック側の BGP ネクストホップに使用されます。ESI マルチパス機能を設定する場合は、エニーキャスト IP アドレスは設定しないでください。

**ステップ 5** **redundancy**

例：

```
RP/0/RSP0/cpu 0: router(config-if)# redundancy
```

冗長パスを設定します。

**ステップ 6** **backbone vxlan**

例：

```
RP/0/RSP0/cpu 0: router(config-nve-red)# backbone vxlan
```

シャーシ間通信プロトコル（ICCP）VXLAN バックボーンの設定

**ステップ 7** **iccp group** *group number*

例：

```
RP/0/RSP0/cpu 0: router(config-nve-red-backbone-vxlan)# iccp group 11
```

ICCP グループ番号を設定します。

**ステップ 8** **exit**

例：

```
RP/0/RSP0/cpu 0: router(config-nve-red-backbone-vxlan)# exit
```

バックボーン vxlan サブモードを終了し、冗長サブモードに戻ります。

**ステップ 9** **backbone mpls**

例：

```
RP/0/RSP0/cpu 0: router(config-nve-red)# backbone mpls
```

ICCP MPLS バックボーンを設定します。

**ステップ 10** **iccp group** *group number*

例：

```
RP/0/RSP0/cpu 0: router(config-nve-red-backbone-mpls)# iccp group 12
```

MPLS バックボーンの ICCP グループ番号を設定します。

**ステップ 11** **exit**

**例**：

```
RP/0/RSP0/cpu 0: router(config-nve-red-backbone-mpls)# exit
```

バックボーン mpls サブモードを終了し、冗長サブモードに戻ります。

**ステップ 12** **exit**

**例**：

```
RP/0/RSP0/cpu 0: router(config-nve-red)# exit
```

冗長サブモードを終了し、インターフェイスサブモードに戻ります。

**ステップ 13** **member vni** *vni_number*

**例**：

```
RP/0/RSP0/cpu 0: router(config-nve)# member vni 1
```

VxLAN ネットワーク識別子（VNI）を使用して単一の VxLAN を NVE インターフェイスに関連付け、この VNI に関連付けられるマルチキャストアドレスを指定します。

**ステップ 14** **load-balance per-evi**

**例**：

```
RP/0/RSP0/cpu 0: router(config-nve-vni)# load-balance per-evi
```

EVI 単位のロードバランスモードを設定します（デフォルトはフロー単位）。

**ステップ 15** **suppress-unknown-unicast-flooding**

**例**：

```
RP/0/RSP0/cpu 0: router(config-nve-vni)# suppress-unknown-unicast-flooding
```

不明なユニキャストフラッディングの抑制を設定します。

**ステップ 16** **mcast-group** *ip_address*

**例**：

```
RP/0/RSP0/cpu 0: router(config-nve-vni)# mcast-group 209.165.202.129
```

VNI に関連付けられるマルチキャストアドレスを指定します。

**ステップ 17** **host-reachability protocol** *protocol*

**例**：

```
RP/0/RSP0/cpu 0: router(config-nve-vni)# host-reachability protocol bgp
```

VxLAN トンネルエンドポイント到達可能性の BGP 制御プロトコルを設定します。

**ステップ 18** **commit** または **end** コマンドを使用します

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# ブリッジ ドメインの設定

次のステップを実行して DCI ゲートウェイ上にブリッジ ドメインを設定します。

### 手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group***bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **evi** *ethernet vpn id*
6. **exit**
7. **member vni** *vxlan-id*
8. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **l2vpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# l2vpn
```

l2vpn コンフィギュレーション モードを開始します。

**ステップ 3** **bridge group***bridge-group-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn)# bridge group bg1
```

ブリッジグループ設定モードを開始します。

**ステップ 4** **bridge-domain** *bridge-domain-name*

例：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg)# bridge-domain bd1
```

ブリッジドメイン設定モードを開始します。

**ステップ 5** **evi** *ethernet vpn id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# evi 1
```

イーサネット VPN ID を作成します。

**ステップ 6** **exit**

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd-evi)# exit
```

EVI 設定モードを終了して、ブリッジドメイン設定モードに戻ります。

**ステップ 7** **member vni** *vxlan-id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-l2vpn-bg-bd)# member vni 1
```

ブリッジドメインにメンバー VNI を関連付けます。

**ステップ 8** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# BGP ルート ターゲットのインポート/エクスポート ルールの設定

デフォルトでは、次のパラメータが DCI の設定から自動生成されます。

- グローバル イーサネット セグメント テーブルのルート識別（RD）

デフォルト：ループバック IP アドレスに基づく自動生成 RD

- EVI の BGP ルート識別子（RD）

デフォルト：ループバック IP アドレスに基づく自動生成 RD

- EVI の BGP ルート ターゲット。デフォルト：EVI ID に基づく自動生成 RT

次のタスクを実行して自動生成 BGP RD/RT 値を上書きし、転送情報のインポートとエクスポートに使用するルート ターゲットを定義します。

### 手順の概要

1. **configure**
2. **evpn**
3. **bgp**
4. **rd** { *2-byte as_number* | *4-byte as_number* | *IP_address* | **none** } : { *nn* }
5. **exit**
6. **evi** *evi_id*
7. **bgp**
8. **route-target import** { *2-byte as_number* | *4-byte as_number* | *IP_address* | **none** } : { *nn* } [**stitching**]
9. **route-target export** { *2-byte as_number* | *4-byte as_number* | *IP_address* | **none** } : { *nn* } [**stitching**]
10. **commit** コマンドまたは **end** コマンドを使用します。

### 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **evpn**

例：

```
RP/0/RSP0/cpu 0: router(config)# evpn
```

EVPN 設定モードを開始します。

**ステップ 3** **bgp**

例：

```
RP/0/RSP0/cpu 0: router(config-evpn)# bgp
```

EVPN BGP 設定モードを開始し、イーサネットセグメント ES:GLOBAL EVI（ES ルートの処理に使用）のスタティック BGP 設定を行います。

**ステップ 4** **rd** { *2-byte as_number* | *4-byte as_number* | *IP_address* | **none** } : { *nn* }

例：

```
RP/0/RSP0/cpu 0: router(config-evpn-bgp)# rd 200:50
```

ルート識別子を設定します。

**ステップ 5** **exit**

例：

```
RP/0/RSP0/cpu 0: router(config-evpn-bgp)# exit
```

現在の設定モードを終了し、EVPN サブモードに戻ります

**ステップ 6** **evi** *evi_id*

**例**：

```
RP/0/RSP0/cpu 0: router(config-evpn)# evi 1
```

イーサネット VPN ID を設定します。

EVI ID の範囲は 1 ～ 65534 です。

**ステップ 7** **bgp**

**例**：

```
RP/0/RSP0/cpu 0: router(config-evpn-evi)# bgp
```

特定の EVI の BGP 設定モードを開始します。

**ステップ 8** **route-target import** { *2-byte as_number* | *4-byte as_number* | *IP_address* | **none** } : { *nn* } [**stitching**]

**例**：

```
RP/0/RSP0/cpu 0: router(config-evpn-evi-bgp)# route-target import 101:1 stitching
```

一致するルートターゲット値を持つ L2 EVPN BGP NLRI からのルートのインポートを設定します。

**ステップ 9** **route-target export** { *2-byte as_number* | *4-byte as_number* | *IP_address* | **none** } : { *nn* } [**stitching**]

**例**：

```
RP/0/RSP0/cpu 0: router(config-evpn-evi-bgp)# route-target export 101:1 stitching
```

L2 EVPN BGP NLRI へのルートのエクスポートを設定し、指定されたルートターゲット識別子を BGP EVPN NLRI に割り当てます。

**ステップ 10** **commit** コマンドまたは **end** コマンドを使用します。

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# イーサネット セグメント識別子の設定

イーサネット セグメント識別子（ESI）を設定するには、次のタスクを実行します。

## 手順の概要

1. **configure**
2. **evpn**
3. **interface nve** *nve-identifier*

4. **ethernet-segment**
5. **identifier type** *esi-type esi-identifier*
6. **bgp route-target** *route target value*
7. **commit** または **end** コマンドを使用します

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **evpn**

例：

```
RP/0/RSP0/cpu 0: router# evpn
```

EVPN 設定モードを開始します。

**ステップ 3** **interface nve** *nve-identifier*

例：

```
RP/0/RSP0/cpu 0: router(config-evpn)# interface nve 1
```

NVE インターフェイスを作成し、NVE インターフェイス設定サブモードを開始します。

**ステップ 4** **ethernet-segment**

例：

```
RP/0/RSP0/cpu 0: router(config-evpn-ac)# ethernet-segment
```

EVPN イーサネットセグメント設定モードを開始します。

**ステップ 5** **identifier type** *esi-type esi-identifier*

例：

```
RP/0/RSP0/cpu 0: router(config-evpn-ac-es)# identifier type 0 88.00.00.00.00.00.00.00.01
```

イーサネットセグメント識別子を設定します。

**ステップ 6** **bgp route-target** *route target value*

例：

```
RP/0/RSP0/cpu 0: router(config-evpn-ac-es)# bgp route-target 8888.0000.0001
```

イーサネットセグメントの BGP インポートルートターゲットを設定します。

**ステップ 7** **commit** または **end** コマンドを使用します

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# ICCP グループの設定

シャーシ間通信プロトコル（ICCP）パラメータを設定するには、次のタスクを実行します。

コアインターフェイス トラッキングに ICCP グループを設定します。すべてのインターフェイスがダウンしている場合、DCI はコア/ファブリック ネットワークから分離されます。関連付けられている NVE インターフェイスがダウンし、BGP NLRI が撤回されます。

## 手順の概要

1. **configure**
2. **redundancy**
3. **iccp group** *group number*
4. **mode singleton**
5. **backbone**
6. **interface GigabitEthernet** *GigabitEthernet Interface Instance*
7. **commit** または **end** コマンドを使用します

## 手順の詳細

**ステップ 1** **configure**

例：

```
RP/0/RSP0/cpu 0: router# configure
```

グローバル コンフィギュレーション モードを開始します。

**ステップ 2** **redundancy**

例：

```
RP/0/RSP0/cpu 0: router(config)# redundancy
```

冗長コンフィギュレーション モードを開始します。

**ステップ 3** **iccp group** *group number*

例：

```
RP/0/RSP0/cpu 0: router(config-redundancy)# iccp group 11
```

ICCP グループ番号を設定します。

**ステップ 4** **mode singleton**

例：

```
RP/0/RSP0/cpu 0: router(config-redundancy-iccp-group)# mode singleton
```

グループをシングルトンモードで実行できるようにします。

**ステップ 5** **backbone**

例：

```
RP/0/RSP0/cpu 0: router(config-redundancy-iccp-group)# backbone
```

ICCP バックボーン インターフェイスを設定します。

**ステップ 6** **interface GigabitEthernet** *GigabitEthernet Interface Instance*

例：

```
RP/0/RSP0/cpu 0: router(config-redundancy-group-iccp-backbone)# interface GigabitEthernet 0/2/0/12
```

GigabitEthernet インターフェイスを設定します。

**ステップ 7** **commit** または **end** コマンドを使用します

**commit** ：設定の変更を保存し、コンフィギュレーション セッションに留まります。

**end** ：次のいずれかのアクションを実行することをユーザに要求します。

- [Yes]：設定変更を保存し、コンフィギュレーション セッションを終了します。
- [No]：設定変更をコミットせずにコンフィギュレーション セッションを終了します。
- [Cancel]：設定変更をコミットせずに、コンフィギュレーション モードに留まります。

# 例：エニーキャスト VTEP IP アドレス設定を使用したオールアクティブ マルチホーミングの設定

次に、エニーキャスト VTEP IP アドレス設定を使用したオールアクティブ マルチホーミングの例を示します。

```
interface nve1
source-interface loopback1
anycast source-interface loopback2
 member vni 5100
  mcast-address 239.1.1.1
  host-reachabilty protocol bgp
!

evpn
 evi 10
  bgp
   route-target import 100:10
   route-target import 200:5100 stitching
   route-target export 200:5100 stitching
```

```
!
!
l2vpn
 bridge group DCI
  bridge-domain V1
   evi 10
   member vni 5100
!
router bgp 100
 bgp router-id 209.165.200.226
 address-family l2vpn evpn

!
 neighbor 209.165.201.2
  remote-as 100
  description core-facing
  update-source Loopback1
  address-family l2vpn evpn
   import re-originate stitching-rt
   advertise l2vpn evpn re-originated
!
neighbor 209.165.202.130
  remote-as 200
  ebgp-multihop 255
  update-source Loopback1
  address-family l2vpn evpn
   import stitching-rt re-originate
   route-policy passall in
   encapsulation-type vxlan
   route-policy passall out
   advertise l2vpn evpn re-originated stitching-rt
!
```

# 例：一意の VTEP IP アドレス設定を使用したオールアクティブ マルチホーミングの設定

次に、一意の VTEP IP アドレス設定を使用したオールアクティブ マルチホーミングの例を示します。

```
interface nve1
source-interface loopback1
 member vni 5100
  mcast-address 239.1.1.1
  host-reachabilty protocol bgp
!
evpn
 evi 10
  bgp
   route-target import 100:10
   route-target import 200:5100 stitching
   route-target export 200:5100 stitching
!
!
l2vpn
 bridge group DCI
  bridge-domain V1
   evi 10
   member vni 5100
```

```
!
router bgp 100
 bgp router-id 209.165.200.226
 address-family l2vpn evpn

!
neighbor 209.165.201.2
 remote-as 100
 description core-facing
 update-source Loopback1
 address-family l2vpn evpn
   import re-originate stitching-rt
   multipath
   advertise l2vpn evpn re-originated
!
neighbor 209.165.202.130
  remote-as 200
  ebgp-multihop 255
  update-source Loopback1
  address-family l2vpn evpn
   import stitching-rt re-originate
    multipath
     route-policy passall in
     encapsulation-type vxlan
     route-policy passall out
     advertise l2vpn evpn re-originated stitching-rt
!
```