仮想プライベート LAN サービス（VPLS）および VPLS BGP ベースの自動検出の設定

VPLS の制約事項

- レイヤ 2 プロトコルトンネリングの設定はサポートされていません。
- Integrated Routing and Bridging（IRB）の設定はサポートされていません。
- 明示的 null の仮想回線接続検証（VCCV）ping はサポートされていません。
- スイッチは、ハブとしてではなく、階層型仮想プライベート LAN サービス（VPLS）でスポークとして設定されている場合にのみサポートされます。
- レイヤ 2 VPN インターワーキング機能はサポートされていません。
- **ip unnumbered** コマンドは、マルチプロトコル ラベル スイッチング（MPLS）構成ではサポートされていません。
- フラッドトラフィックの場合、仮想回線（VC）統計情報は、**show mpls l2 vc vcid detail** コマンドの出力に表示されません。
- 接続回線では、Dot1q トンネル構成はサポートされていません。

VPLS、VPLS BGP ベースの自動検出、および Flow Aware Transport に関する情報

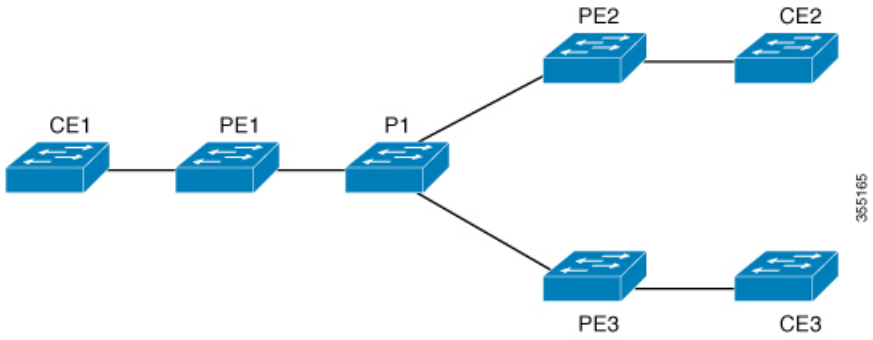
次の項では、VPLS、VPLS BGP ベースの自動検出、および Flow Aware Transport について説明します。

VPLS の概要

VPLS により、企業は、サービスプロバイダーから提供されるインフラストラクチャを介して、複数サイトからのイーサネットベースの LAN をまとめてリンクできます。企業の側からは、サービスプロバイダーのパブリックネットワークは、1 つの大きなイーサネット LAN のように見えます。サービス プロバイダーからすると、VPLS は、大規模な設備投資なしで、既存のネットワーク上に収益を生み出す新たなサービスを導入するチャンスになります。オペレータは、ネットワークでの機器の運用年数を延長できます。

VPLS はプロバイダーコアを使用して複数の接続回線をまとめ、複数の接続回線間の仮想ブリッジをシミュレートします。VPLS のトポロジは、カスタマーからは認識されません。すべてのカスタマーエッジ（CE）デバイスは、プロバイダーコアによってエミュレートされた論理ブリッジに接続されているように見えます。

図 *1 : VPLS* トポロジ



フルメッシュ構成について

フルメッシュ構成では、VPLS に参加するすべてのプロバイダーエッジ（PE）デバイス間でトンネルラベルスイッチパス（LSP）のフルメッシュが必要です。フルメッシュ構成では、シグナリングのオーバーヘッドと、PE デバイス上でプロビジョニング対象の VC に対するパケット複製の要件が多くなります。

フルメッシュ構成の場合、参加している各 PE デバイスに仮想転送インスタンス（VFI）が必要です。VFI には、VPLS ドメインの VPN ID、そのドメインの他の PE デバイスのアドレス、トンネルシグナリングのタイプ、各ピア PE デバイスのカプセル化のメカニズムが含まれます。

VPLS インスタンスは、エミュレート VC の相互接続によって形成される一連の VFI を構成します。VPLS インスタンスは、パケット交換ネットワーク上の論理ブリッジを形成します。VPLS インスタンスには、一意の VPN ID が割り当てられます。

PE デバイスは、VFI を使用して、エミュレートされた VC から VPLS インスタンスの他のすべての PE デバイスまでのフルメッシュ LPS を確立します。PE デバイスは、Cisco IOS CLI を使用して、スタティック設定を通じた VPLS インスタンスのメンバーシップを取得します。

フルメッシュ構成では、PE デバイスが単一のブロードキャストドメインを維持できます。そのため、接続回線でブロードキャスト、マルチキャスト、または未知のユニキャストパケットを受信すると、PE デバイスは、他のすべての接続回線およびエミュレート回線のパケットを、その VPLS インスタンスに参加している他のすべての CE デバイスへに送信します。CE デバイスでは、VPLS インスタンスを、エミュレート LAN として認識します。

プロバイダーコアでのパケットループの問題を回避するために、PE デバイスは、エミュレート VC に「スプリットホライズン」の原則を適用します。スプリットホライズンの原則により、エミュレート VC でパケットを受信したパケットは、他のいずれのエミュレート VC にも転送されなくなります。

VFI を定義したら、CE デバイスへの接続回線にバインドする必要があります。

パケット転送の判断は、特定の VPLS ドメインのレイヤ 2 VFI を検索することによって行われます。

特定の PE デバイスの VPLS インスタンスは、特定の物理または論理ポートに着信するイーサネットフレームを受信し、イーサネットスイッチによる動作同様に、MAC アドレステーブルに入力します。PE デバイスは、この MAC アドレスを使用して、リモートサイトにある別の PE デバイスに配布するために、このようなフレームを適切な LSP に切り替えます。

MAC アドレスが MAC アドレステーブルにない場合、PE デバイスは、イーサネットフレームを複製し、イーサネットフレームが入力された入力ポートを除く、その VPLS インスタンスに関連付けられたすべての論理ポートにフラッディングします。PE デバイスは、特定のポートでパケットを受信したときに MAC アドレステーブルを更新し、一定期間使用されていないアドレスを削除します。

VPLS BGP ベースの自動検出について

VPLS 自動検出を使用すると、各 PE デバイスで、同じ VPLS ドメインの一部である他の PE デバイスを検出できます。VPLS 自動検出は、PE デバイスが VPLS ドメインに追加、またはドメインから削除されたタイミングも追跡します。VPLS 自動検出を有効にすると、VPLS ドメインを手動で設定したり、PE デバイスが追加または削除されたときに設定を維持したりする必要がなくなります。VPLS 自動検出は、ボーダー ゲートウェイ プロトコル（BGP）を使用して、VPLS メンバーを検出し、VPLS ドメイン内の擬似回線（PW）をセットアップおよび解除します。

BGP では、エンドポイント プロビジョニング情報を保存する際にレイヤ 2 VPN ルーティング情報ベース（RIB）が使用されます。これは、レイヤ 2 VFI が設定されるたびにアップデートされます。プレフィックスおよびパス情報はレイヤ 2 VPN データベースに保存され、ベストパスが BGP により決定されるようになります。BGP により、更新メッセージですべての BGP ネ

イバーにエンドポイントプロビジョニング情報が配布される場合、レイヤ 2 VPN ベースのサービスをサポートするために、このエンドポイント情報を使用して擬似回線メッシュが設定されます。

BGP 自動検出のメカニズムにより、VPLS 機能に必要不可欠なレイヤ 2 VPN サービスの設定が簡易化されます。VPLS は、高速イーサネットを使用した堅牢でスケーラブルな IP MPLS ネットワークによる大規模な LAN として、地理的に分散した拠点間を接続することで柔軟なサービスの展開を実現します。

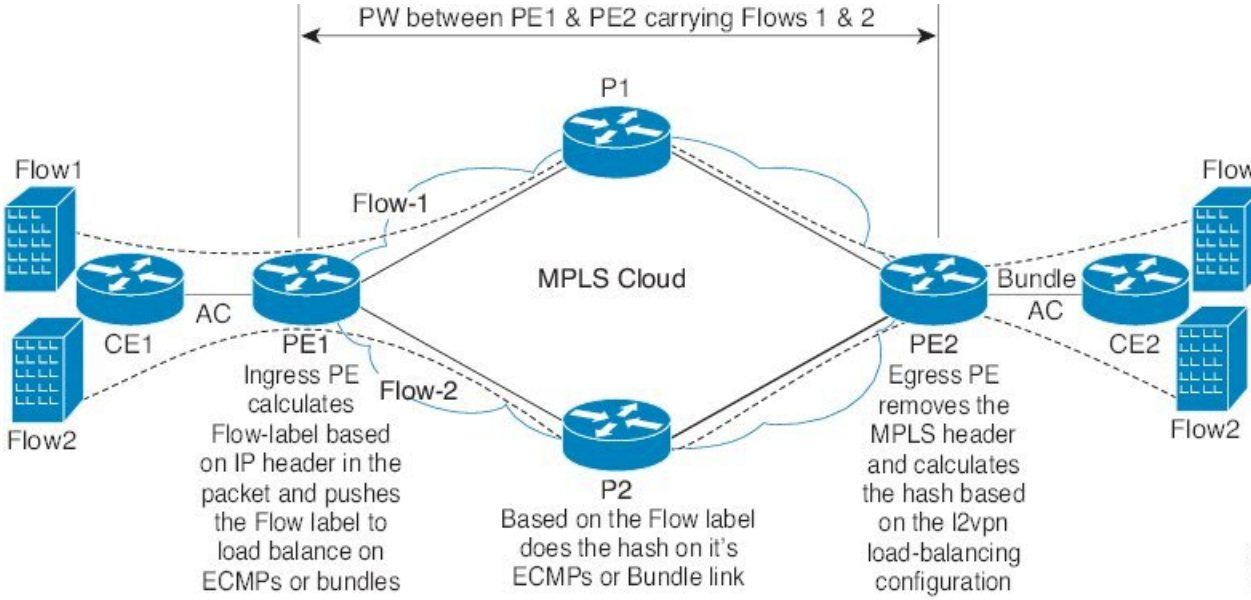
Flow Aware Transport 疑似回線について

デバイスは通常、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）デバイスから宛先 PE デバイスに転送されます。

Flow Aware Transport PW は、PW 内の個々のフローを識別する機能を提供します。また、そ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デバイスに提供します。Equal Cost Multipath（ECMP; 等コストマルチパス）が使用されている場合、Flow Aware Transport 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。PW 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。デバイスは、フローラベルをロードバランシングに使用でき、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行われます。

図 2 : Flow Aware Transport PW と、ECMP およびバンドルされたリンクへ分配される 2 つのフローに、Flow Aware Transport PW と、ECMP およびバンドルされたリンクへ分配される 2 つのフローの例を示します。

図 *2 : Flow Aware Transport 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



追加のラベルは、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。Flow Aware Transport PW コンフィギュレーションは、フローラベルを有効にします。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。

すべてのコアデバイスが、Flow Aware Transport 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。

Flow Aware Transport PW は、ポートチャネル ロードバランシング アルゴリズムのみに基づいて動作します。

Cisco Catalyst 6000 シリーズ スイッチと Cisco Catalyst 9000 シリーズ スイッチ間の相互運用性

次の項では、Cisco Catalyst 6000 シリーズ スイッチと Cisco Catalyst 9000 シリーズ スイッチ間でフローラベルを送受信できるようにする方法について説明します。

Flow Aware Transport PW（Advanced VPLS を使用）で設定された Cisco Catalyst 6000 シリーズ スイッチでは、フローラベルのネゴシエーションはサポートされていません。Cisco Catalyst 6000 シリーズ スイッチが Cisco Catalyst 9000 シリーズ スイッチなどのリモート PE デバイスと相互運用可能な場合、Cisco Catalyst 9000 シリーズ スイッチはデータトラフィックのフローラベルを送受信できません。Cisco Catalyst 9000 シリーズ スイッチで **load-balance flow-label both static** コマンドを設定すると、Cisco Catalyst 6000 シリーズ スイッチがフローラベルのネゴシエーションをサポートしていない場合でも、Cisco Catalyst 9000 シリーズ スイッチでフローラベルを送受信できます。

次に、フローラベルの送受信を有効にする設定例を示します。

```
Device> enable
Device# configure terminal
Device(config)# template type pseudowire mpls
Device(config-template)# encapsulation mpls
Device(config-template# load-balance flow ip dst-ip
Device(config-template)# load-balance flow-label both static
Device(config-template# end
```

VPLS、VPLS BGP ベースの自動検出、および Flow Aware Transport の設定方法

次の項では、VPLS、VPLS BGP ベースの自動検出、および Flow Aware Transport に関する設定情報について説明します。

CE デバイスへのレイヤ 2 PE デバイスインターフェイスの設定

CE デバイスへのレイヤ 2 PE デバイスインターフェイスを設定する必要があります。次の項では、VPLS を設定する前に完了する必要があるさまざまな設定作業について説明します。

CE デバイスからのタグ付きトラフィックを受け取る PE デバイスの 802.1Q トランクの設定

PE デバイスで 802.1Q トランクを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 3	**interface** *interface-id* 例： `Device(config)# interface TenGigabitEthernet1/0/24`	トランクとして設定するインターフェイスを定義し、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 4	**no ip address** *ip_address mask* [**secondary**] 例： `Device(config-if)# no ip address`	IP 処理をディセーブルにして、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 5	**switchport** 例： `Device(config-if)# switchport`	レイヤ 2 スイッチドインターフェイスのスイッチング特性を変更します。
ステップ 6	**switchport trunk encapsulation dot1q** 例： `Device(config-if)# switchport trunk encapsulation dot1q`	スイッチ ポートのカプセル化形式を 802.1Q に設定します。
ステップ 7	**switchport trunk allow vlan** *vlan_ID* 例：	許可 VLAN のリストを設定します。

	コマンドまたはアクション	目的
	`Device(config-if)# switchport trunk allow vlan 2129`	
ステップ 8	**switchport mode trunk** 例： `Device(config-if)# switchport mode trunk`	トランキング VLAN レイヤ 2 インターフェイスへのインターフェイスを設定します。
ステップ 9	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

CE デバイスからのタグなしトラフィックを受け取る PE デバイスの 802.1Q アクセスポートの設定

PE デバイスで 802.1Q アクセスポートを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 3	**interface** *interface-id* 例： `Device(config)# interface TenGigabitEthernet1/0/24`	トランクとして設定するインターフェイスを定義し、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 4	**no ip address** *ip_address mask* [**secondary**] 例： `Device(config-if)# no ip address`	IP 処理をディセーブルにします。
ステップ 5	**switchport** 例：	レイヤ 2 スイッチドインターフェイスのスイッチング特性を変更します。

	コマンドまたはアクション	目的
	`Device(config-if)# switchport`	
ステップ 6	**switchport mode access** 例： `Device(config-if)# switchport mode access`	インターフェイスタイプを、非トランキング、タグなし、シングル VLAN レイヤ 2 インターフェイスとして設定します。
ステップ 7	**switchport access vlan** *vlan_ID* 例： `Device(config-if)# switchport access vlan 2129`	インターフェイスがアクセス モードのときに VLAN を設定します。
ステップ 8	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

PE デバイスでのレイヤ 2 VLAN インスタンスの設定

PE デバイスにレイヤ 2 VLAN インターフェイスを設定すると、VLAN データベースへの PE デバイス上のレイヤ 2 VLAN インスタンスで、VPLS と VLAN 間のマッピングを設定できます。

PE デバイスでレイヤ 2 VLAN インスタンスを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 3	**vlan** *vlan-id* 例： `Device(config)# vlan 2129`	特定の VLAN を設定します。
ステップ 4	**interface vlan** *vlan-id* 例：	この VLAN に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
	`Device(config-vlan)# interface vlan 2129`	
ステップ 5	**end** 例： `Device(config-vlan)# end`	特権 EXEC モードに戻ります。

VPLS の設定

VPLS は、Xconnect モードまたはプロトコル CLI 方式を使用して設定できます。次の項では、VPLS の設定方法について説明します。

Xconnect モードでの VPLS の設定

次の項では、Xconnect モードでの VPLS の設定について説明します。

PE デバイス上での MPLS の設定

PE デバイスで MPLS を設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 3	**mpls ip** 例： `Device(config)# mpls ip`	MPLS ホップバイホップ転送を設定します。
ステップ 4	**mpls label protocol ldp** 例： `Device(config)# mpls label protocol ldp`	プラットフォームの Label Distribution Protocol（LDP; ラベル配布プロトコル）を指定します。

	コマンドまたはアクション	目的
ステップ 5	**mpls ldp logging neighbor-changes** 例： `Device(config)# mpls ldp logging neighbor-changes`	（任意）ネイバーの変更の記録を指定します。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

PE デバイスでの VFI の設定

VFI によって VPLS ドメインの VPN ID、そのドメインの他の PE デバイスのアドレス、トンネルのシグナリングのタイプ、各ピアデバイスのカプセル化のメカニズムが指定されます。

PE デバイスで VFI および関連する VC を設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 3	**l2 vfi** *vfi-name* **manual** 例： `Device(config)# l2 vfi 2129 manual`	レイヤ 2 VFI 手動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 4	**vpn id** *vpn-id* 例： `Device(config-vfi)# vpn id 2129`	VPLS ドメインの VPN ID を設定します。このレイヤ 2 Virtual Routing Forwarding（VRF）にバインドされたエミュレート VC でシグナリングにこの VPN ID が使用されます。 （注） *vpn-id* は *vlan-id* と同じです。
ステップ 5	**neighbor** *router-id* {**encapsulation mpls**} 例：	リモートピアリングルータ ID と、エミュレート VC をセットアップするために使用されるトンネルカプ

	コマンドまたはアクション	目的
	`Device(config-vfi)# neighbor remote-router-id encapsulation mpls`	セル化タイプまたは疑似回線（PW）プロパティを指定します。
ステップ 6	**end** 例： `Device(config-vfi)# end`	特権 EXEC モードに戻ります。

PE デバイスでの VFI への接続回線の関連付け

VFI を定義したら、1 つ以上の接続回線に関連付ける必要があります。

接続回線を VFI に関連付けるには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 3	**interface vlan** *vlan-id* 例： `Device(config)# interface vlan 2129`	動的なスイッチ仮想インターフェイス（SVI）を作成するか、使用します。 （注）　*vlan-id* は *vpn-id* と同じです。
ステップ 4	**no ip address** 例： `Device(config-if)# no ip address`	IP 処理をディセーブルにします。（IP アドレスを設定する場合は、VLAN のレイヤ 3 インターフェイスを設定できます）。
ステップ 5	**xconnect vfi** *vfi-name* 例： `Device(config-if)# xconnect vfi 2129`	VLAP ポートにバインドするレイヤ 2 VFI を指定します。
ステップ 6	**end** 例：	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	`Device(config-if)# end`	

プロトコル CLI モードでの VPLS の設定

次の項では、プロトコル CLI モードでの VPLS の設定について説明します。

プロトコル CLI モードでの VPLS の設定

プロトコル CLI モードで VPLS を設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 3	**l2vpn vfi context** *vfi-name* 例： `Device(config)# l2vpn vfi context vpls1`	レイヤ 2 VPN VFI コンテキストを確立して、レイヤ 2 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 4	**vpn id** *vpn-id* 例： `Device(config-vfi)# vpn id 10`	VPLS ドメインの VPN ID を設定します。
ステップ 5	**member** *ip-address* **encapsulation mpls** 例： `Device(config-vfi)# member 2.2.2.2 encapsulation mpls`	ポイントツーポイントレイヤ 2 VPN VFI 接続を形成するデバイスを指定します。
ステップ 6	**exit** 例： `Device(config-vfi)# exit`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的

ステップ 7	次のいずれかを選択します。
	• **vlan configuration** *vlan-id*
	• **interface vlan** *vlan-id*
	例：
	`Device(config)# vlan configuration 100`
	`OR`
	`Device(config)# interface vlan 100`
		VLAN またはインターフェイスに適用する設定を適用し、VLAN またはインターフェイス コンフィギュレーション モードを開始します。

ステップ 8	**member vfi** *vfi-name* 例： `Device(config-vlan-config)# member vfi vpls1`	VFI インスタンスを VLAN またはインターフェイスにバインドします。
ステップ 9	**end** 例： `Device(config-vlan-config)# end`	特権 EXEC モードに戻ります。

疑似回線インターフェイスを使用した VPLS Flow Aware Transport の設定（プロトコル CLI モード）

擬似回線インターフェイスを使用して VPLS Flow Aware Transport を設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 3	**interface pseudowire** *number* 例： `Device(config)# interface pseudowire 1001`	指定した名前で PW を確立して、疑似回線インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**encapsulation mpls** 例：	トンネリング カプセル化を MPLS として指定します。

	コマンドまたはアクション	目的
	`Device(config-if)# encapsulation mpls`	
ステップ 5	**neighbor** *peer-address vcid-value* 例： `Device(config-if)# neighbor 10.1.1.200 200`	レイヤ 2 VPN PW のピア IP アドレスと VC ID 値を指定します。
ステップ 6	**load-balance flow** 例： `Device(config-if)# load-balance flow`	ロードバランシングがフロー単位で実行されるように、PW機能を使用したロードバランシングを有効にします。
ステップ 7	**load-balance flow-label** 例： `Device(config-if)# load-balance flow-label both`	MPLS PW 機能の Flow Aware Transport を有効にして、フローラベルの使用方法を指定します。
ステップ 8	**exit** 例： `Device(config-if)# exit`	特権 EXEC モードに戻ります。
ステップ 9	**l2vpn vfi context** *vfi-name* 例： `Device(config)# l2vpn vfi context vpls1`	レイヤ 2 VPN VFI コンテキストを確立して、レイヤ 2 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 10	**vpn id** *vpn-id* 例： `Device(config-vfi)# vpn id 10`	VPLS ドメインの VPN ID を設定します。
ステップ 11	**member pseudowire** *number* 例： `Device(config-vfi)# member pseudowire 1001`	疑似回線インターフェイスを VFI のメンバーとして追加します。
ステップ 12	**exit** 例： `Device(config-vfi)# exit`	特権 EXEC モードに戻ります。
ステップ 13	次のいずれかを選択します。 • **vlan configuration** *vlan-id* • **interface vlan** *vlan-id*	VLAN またはインターフェイスに適用する設定を適用し、VLAN またはインターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的

	例：
	`Device(config)# vlan configuration 100`
	OR
	`Device(config)# interface vlan 100`	

ステップ 14	**member vfi** *vfi-name* 例： `Device(config-vlan-config)# member vfi vpls1`	VFI インスタンスを VLAN またはインターフェイスにバインドします。
ステップ 15	**end** 例： `Device(config-vlan-config)# end`	特権 EXEC モードに戻ります。

テンプレートを使用した VPLS Flow Aware Transport の設定（プロトコル CLI モード）

テンプレートを使用して VPLS Flow Aware Transport を設定すると、複数の PW が同じ設定を共有できます。

テンプレートを使用して VPLS Flow Aware Transport を設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 3	**template type pseudowire** [*template-name*] 例： `Device(config)# template type pseudowire mpls`	レイヤ 2 PW の名前を指定し、擬似回線テンプレート 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 mpls** 例： `Device(config-template)# encapsulation mpls`	トンネリング カプセル化を MPLS として指定します。

	コマンドまたはアクション	目的
ステップ 5	**load-balance flow** 例： `Device(config-template)# load-balance flow`	ロードバランシングがフロー単位で実行されるように、PW機能を使用したロードバランシングを有効にします。
ステップ 6	**load-balance flow-label** 例： `Device(config-template)# load-balance flow-label both`	MPLS PW 機能の Flow Aware Transport を有効にして、フローラベルの使用方法を指定します。
ステップ 7	**exit** 例： `Device(config-template)# exit`	特権 EXEC モードに戻ります。
ステップ 8	**l2vpn vfi context** *vfi-name* 例： `Device(config)# l2vpn vfi context vpls1`	レイヤ 2 VPN VFI コンテキストを確立して、レイヤ 2 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 9	**vpn id** *vpn-id* 例： `Device(config-vfi)# vpn id 10`	VPLS ドメインの VPN ID を設定します。
ステップ 10	**member** *ip-address* **template** *template-name* 例： `Device(config-vfi)# member 102.102.102.102 template mpls`	ポイントツーポイントレイヤ 2 VPN VFI 接続を形成するデバイスを指定します。 • **ip-address**：VFI ネイバーの IP アドレス。 • template-name：テンプレート方式としてテンプレート名 mpls を指定します。**template**
ステップ 11	**exit** 例： `Device(config-vfi)# exit`	特権 EXEC モードに戻ります。
ステップ 12	次のいずれかを選択します。 • **vlan configuration** *vlan-id* • **interface vlan** *vlan-id* 例： `Device(config)# vlan configuration 100`	VLAN またはインターフェイスに適用する設定を適用し、VLAN またはインターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	OR `Device(config)# interface vlan 100`	
ステップ 13	**member vfi** *vfi-name* 例： `Device(config-vlan-config)# member vfi vpls1`	VFI インスタンスを VLAN またはインターフェイスにバインドします。
ステップ 14	**end** 例： `Device(config-vlan-config)# end`	特権 EXEC モードに戻ります。

疑似回線とテンプレートを使用した VPLS Flow Aware Transport の設定（プロトコル CLI モード）

PW とテンプレートの両方を使用して VPLS Flow Aware Transport を設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 3	**template type pseudowire** [*template-name*] 例： `Device(config)# template type pseudowire mpls`	レイヤ 2 PW の名前を指定し、擬似回線テンプレート 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 mpls** 例： `Device(config-template)# encapsulation mpls`	トンネリング カプセル化を MPLS として指定します。
ステップ 5	**load-balance flow** 例： `Device(config-template)# load-balance flow`	ロードバランシングがフロー単位で実行されるように、PW 機能を使用したロードバランシングを有効にします。

	コマンドまたはアクション	目的
ステップ 6	**load-balance flow-label** 例： `Device(config-template)# load-balance flow-label both`	MPLS PW 機能の Flow Aware Transport を有効にして、フローラベルの使用方法を指定します。
ステップ 7	**exit** 例： `Device(config-template)# exit`	特権 EXEC モードに戻ります。
ステップ 8	**interface pseudowire** *number* 例： `Device(config)# interface pseudowire 1001`	指定した名前でPWを確立して、疑似回線インターフェイス コンフィギュレーション モードを開始します。
ステップ 9	**source template type pseudowire** [*template-name*] 例： `Device(config-if)# source template type pseudowire mpls`	mpls という名前のタイプ擬似回線のソーステンプレートを設定します。
ステップ 10	**neighbor** *peer-address vcid-value* 例： `Device(config-if)# neighbor 10.1.1.200 200`	レイヤ 2 VPN PW のピア IP アドレスと VC ID 値を指定します。
ステップ 11	**exit** 例： `Device(config-if)# exit`	特権 EXEC モードに戻ります。
ステップ 12	**l2vpn vfi context** *vfi-name* 例： `Device(config)# l2vpn vfi context vpls1`	レイヤ 2 VPN VFI コンテキストを確立して、レイヤ 2 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 13	**vpn id** *vpn-id* 例： `Device(config-vfi)# vpn id 10`	VPLS ドメインの VPN ID を設定します。
ステップ 14	**member pseudowire** *number* 例： `Device(config-vfi)# member pseudowire 1001`	疑似回線インターフェイスを VFI のメンバーとして追加します。

	コマンドまたはアクション	目的
ステップ 15	**exit** 例： `Device(config-vfi)# exit`	特権 EXEC モードに戻ります。

ステップ 16	次のいずれかを選択します。
	• **vlan configuration** *vlan-id*
	• **interface vlan** *vlan-id*
	例：
	`Device(config)# vlan configuration 100`
	`OR`
	`Device(config)# interface vlan 100`
		VLAN またはインターフェイスに適用する設定を適用し、VLAN またはインターフェイス コンフィギュレーション モードを開始します。

ステップ 17	**member vfi** *vfi-name* 例： `Device(config-vlan-config)# member vfi vpls1`	VFI インスタンスを VLAN またはインターフェイスにバインドします。
ステップ 18	**end** 例： `Device(config-vlan-config)# end`	特権 EXEC モードに戻ります。

VPLS BGP ベースの自動検出の設定

次の項では、VPLS BGP ベースの自動検出の設定方法について説明します。

VPLS BGP ベースの自動検出のイネーブル化

VPLS BGP ベースの自動検出を有効にするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例：	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。

	コマンドまたはアクション	目的
	`Device# configure terminal`	
ステップ 3	**l2 vfi** *vfi-name* **autodiscovery** 例： `Device(config)# l2 vfi 2128 autodiscovery`	PE デバイス上で VPLS 自動検出を有効にして、L2 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 4	**vpn id** *vpn-id* 例： `Device(config-vfi)# vpn id 2128`	VPLS ドメインの VPN ID を設定します。
ステップ 5	**end** 例： `Device(config-vfi)# end`	特権 EXEC モードに戻ります。

VPLS 自動検出を有効にする BGP の設定

VPLS 自動検出を有効にするように BGP を設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 3	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 1000`	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。

	コマンドまたはアクション	目的
ステップ 4	**no bgp default ipv4-unicast** 例： `Device(config-router)# no bgp default ipv4-unicast`	BGP ルーティング プロセスで使用される IPv4 ユニキャスト アドレス ファミリを無効にします。 （注） IPv4 ユニキャスト アドレス ファミリのルーティング情報は、**neighbor remote-as router** コマンドを使用して設定された各 BGP ルーティング セッションに対して、デフォルトでアドバタイズされます。ただし、 **neighbor remote-as** コマンドを設定する前に、**no bgp default ipv4-unicast** コマンドを設定した場合は除きます。既存のネイバー コンフィギュレーションは影響されません。
ステップ 5	**bgp log-neighbor-changes** 例： `Device(config-router)# bgp log-neighbor-changes`	BGP ネイバー リセットのロギングを有効にします。
ステップ 6	**neighbor remote-as** { *ip-address* | *peer-group-name* } **remote-as** *autonomous-system-number* 例： `Device(config-router)# neighbor 44.254.44.44 remote-as 1000`	指定された自律システム内のネイバーの IP アドレスまたはピア グループ名を、ローカル デバイスの IPv4 マルチプロトコル BGP ネイバー テーブルに追加します。 • *autonomous-system-number* 引数が、**router bgp** コマンドで指定された自律システム番号と一致する場合、ネイバーは内部ネイバーになります。 • *autonomous-system-number* 引数が、**router bgp** コマンドで指定された自律システム番号と一致しない場合、ネイバーは外部ネイバーになります。
ステップ 7	**neighbor** { *ip-address* | *peer-group-name* } **update-source** *interface-type interface-number* 例： `Device(config-router)# neighbor 44.254.44.44 update-source Loopback300`	（任意）ルーティング テーブル アップデートを受信するための特定のソースまたはインターフェイスを選択するようにデバイスを設定します。
ステップ 8	他の BGP ネイバーを設定する場合は、ステップ 6 と 7 を繰り返します。	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。

	コマンドまたはアクション	目的
ステップ 9	**address-family l2vpn** [**vpls**] 例： `Device(config-router)# address-family l2vpn vpls`	レイヤ 2 VPN 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。 オプションの **vpls** キーワードは、VPLS エンドポイント プロビジョニング情報が BGP ピアに配布されるように指定します。
ステップ 10	**neighbor** { *ip-address* | *peer-group-name* } **activate** 例： `Device(config-router-af)# neighbor 44.254.44.44 activate`	BGP ネイバーとの情報交換を有効にします。
ステップ 11	**neighbor** { *ip-address* | *peer-group-name* } **send-community** { **both** | **standard** | **extended** } 例： `Device(config-router-af)# neighbor 44.254.44.44 send-community both`	コミュニティ属性が BGP ネイバーに送信されるように指定します。
ステップ 12	ステップ 10 と 11 を繰り返して、L2VPN アドレスファミリ内の他の BGP ネイバーをアクティブにします。	
ステップ 13	**exit-address-family** 例： `Device(config-router-af)# exit-address-family`	アドレス ファミリ コンフィギュレーション モードを終了し、ルータ コンフィギュレーション モードに戻ります。
ステップ 14	**end** 例： `Device(config-router)# end`	ルータ コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。

プロトコル CLI モードでの VPLS BGP ベースの自動検出の設定

次の項では、プロトコル CLI モードでの VPLS BGP ベースの自動検出の設定について説明します。

プロトコル CLI モードでの VPLS BGP ベースの自動検出の設定

プロトコル CLI モードで VPLS BGP ベースの自動検出を設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 3	**l2vpn vfi context** *vfi-name* 例： `Device(config)# l2vpn vfi context vpls1`	レイヤ 2 VPN VFI コンテキストを確立して、レイヤ 2 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 4	**vpn id** *vpn-id* 例： `Device(config-vfi)# vpn id 10`	VPLS ドメインの VPN ID を設定します。
ステップ 5	**autodiscovery bgp signaling ldp** 例： `Device(config-vfi)# autodiscovery bgp signaling ldp`	BGP シグナリングと LDP シグナリングを有効にします。
ステップ 6	**exit** 例： `Device(config-vfi-autodiscovery)# exit`	特権 EXEC モードに戻ります。
ステップ 7	**exit** 例： `Device(config-vfi)# exit`	特権 EXEC モードに戻ります。
ステップ 8	次のいずれかを選択します。 • **vlan configuration** *vlan-id* • **interface vlan** *vlan-id*	VLAN またはインターフェイスに適用する設定を適用し、VLAN またはインターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的

	例：
	
	`Device(config)# vlan configuration 100`
	`OR`
	`Device(config)# interface vlan 100`	

ステップ 9	**member vfi** *vfi-name* 例： `Device(config-vlan-config)# member vfi vpls1`	VFI インスタンスを VLAN またはインターフェイスにバインドします。
ステップ 10	**end** 例： `Device(config-vlan-config)# end`	特権 EXEC モードに戻ります。

テンプレートを使用した VPLS BGP ベースの自動検出 Flow Aware Transport の設定（プロトコル CLI モード）

テンプレートを使用して VPLS BGP ベースの自動検出 Flow Aware Transport を設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# 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 3	**template type pseudowire** [*template-name*] 例： `Device(config)# template type pseudowire mpls`	レイヤ 2 PW の名前を指定し、擬似回線テンプレート 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 mpls** 例： `Device(config-template)# encapsulation mpls`	トンネリング カプセル化を MPLS として指定します。
ステップ 5	**load-balance flow** 例：	ロードバランシングがフロー単位で実行されるように、PW 機能を使用した Any Transport over MPLS

	コマンドまたはアクション	目的
	`Device(config-template)# load-balance flow`	（AToM）ロードバランシング機能を有効にします。
ステップ 6	**load-balance flow-label** 例： `Device(config-template)# load-balance flow-label both`	MPLS PW 機能の Flow Aware Transport を有効にして、フローラベルの使用方法を指定します。
ステップ 7	**exit** 例： `Device(config-template)# exit`	特権 EXEC モードに戻ります。
ステップ 8	**l2vpn vfi context** *vfi-name* 例： `Device(config)# l2vpn vfi context vpls1`	レイヤ 2 VPN VFI コンテキストを確立して、レイヤ 2 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 9	**vpn id** *vpn-id* 例： `Device(config-vfi)# vpn id 10`	VPLS ドメインの VPN ID を設定します。
ステップ 10	**autodiscovery bgp signaling ldp template** *name* 例： `Device(config-vfi)# autodiscovery bgp signaling ldp template mpls`	BGP シグナリングと LDP シグナリングを有効にします。
ステップ 11	**exit** 例： `Device(config-vfi)# exit`	特権 EXEC モードに戻ります。

ステップ 12	次のいずれかを選択します。
	• **vlan configuration** *vlan-id*
	• **interface vlan** *vlan-id*
	例：
	`Device(config)# vlan configuration 100`
	`OR`
	`Device(config)# interface vlan 100`
		VLAN またはインターフェイスに適用する設定を適用し、VLAN またはインターフェイス コンフィギュレーション モードを開始します。

ステップ 13	**member vfi** *vfi-name* 例：	VFI インスタンスを VLAN またはインターフェイスにバインドします。

	コマンドまたはアクション	目的
	`Device(config-vlan-config)# member vfi vpls1`	
ステップ 14	**end** 例： `Device(config-vlan-config)# end`	特権 EXEC モードに戻ります。

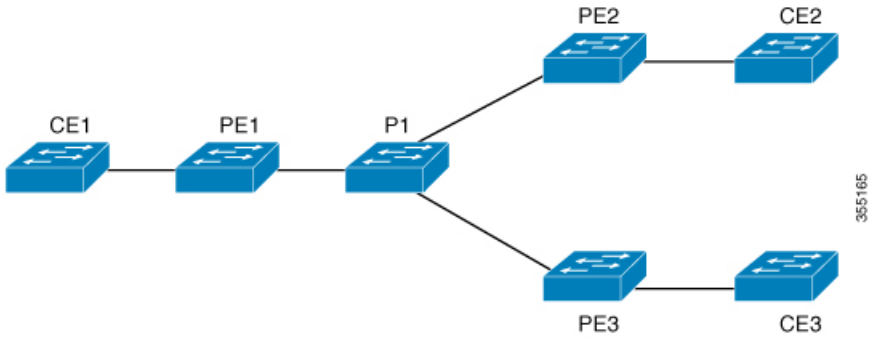
VPLS および VPLS BGP ベースの自動検出の設定例

この項では、VPLS および VPLS BGP ベースの自動検出の設定例を示します。

例：Xconnect モードでの VPLS の設定

次に、PE1 および PE2 デバイスで VPLS を設定する例を示します。

図 *3 : VPLS* トポロジ



PE1 の設定

```
Device> enable
Device# configure terminal
Device(config)# pseudowire-class vpls2129
Device(config-if)# encapsulation mpls
Device(config-if)# exit
Device(config)# l2 vfi 2129 manual
Device(config-vfi)# vpn id 2129
Device(config-vfi)# neighbor 44.254.44.44 pw-class vpls2129
Device(config-vfi)# neighbor 188.98.89.98 pw-class vpls2129
Device(config-vfi)# exit
Device(config)# interface TenGigabitEthernet1/0/24
Device(config-if)# switchport trunk allowed vlan 2129
Device(config-if)# switchport mode trunk
Device(config-if)# exit
Device(config)# interface vlan 2129
Device(config-vlan-config)# no ip address
Device(config-vlan-config)# xconnect vfi 2129
```

例：Xconnect モードで設定されたVPLS の確認

次に、**show mpls 12transport vc detail** コマンドの出力例を示します。このコマンドの出力には、仮想回線に関する情報が表示されます。

```
Device# show mpls 12transport vc detail
 Local interface: VFI 2129 vfi up
  Interworking type is Ethernet
  Destination address: 44.254.44.44, VC ID: 2129, VC status: up
    Output interface: Gi1/0/9, imposed label stack {18 17}
    Preferred path: not configured
    Default path: active
    Next hop: 177.77.177.2
  Create time: 19:09:33, last status change time: 09:24:14
    Last label FSM state change time: 09:24:14
  Signaling protocol: LDP, peer 44.254.44.44:0 up
    Targeted Hello: 1.1.1.72(LDP Id) -> 44.254.44.44, LDP is UP
    Graceful restart: configured and enabled
    Non stop routing: not configured and not enabled
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
Last BFD dataplane     status rcvd: Not sent
      Last BFD peer monitor  status rcvd: No fault
      Last local AC  circuit status rcvd: No fault
      Last local AC  circuit status sent: No fault
      Last local PW i/f circ status rcvd: No fault
      Last local LDP TLV     status sent: No fault
      Last remote LDP TLV    status rcvd: No fault
      Last remote LDP ADJ    status rcvd: No fault
MPLS VC labels: local 512, remote 17
    Group ID: local n/a, remote 0
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
```

```
  Control Word: Off
  SSO Descriptor: 44.254.44.44/2129, local label: 512
  Dataplane:
    SSM segment/switch IDs: 20498/20492 (used), PWID: 2
  VC statistics:
    transit packet totals: receive 0, send 0
    transit byte totals:   receive 0, send 0
    transit packet drops:  receive 0, seq error 0, send 0
```

次に、**show l2vpn atom vc** コマンドの出力例を示します。このコマンドの出力には、ATM over MPLS が VC に設定されていることが示されます。

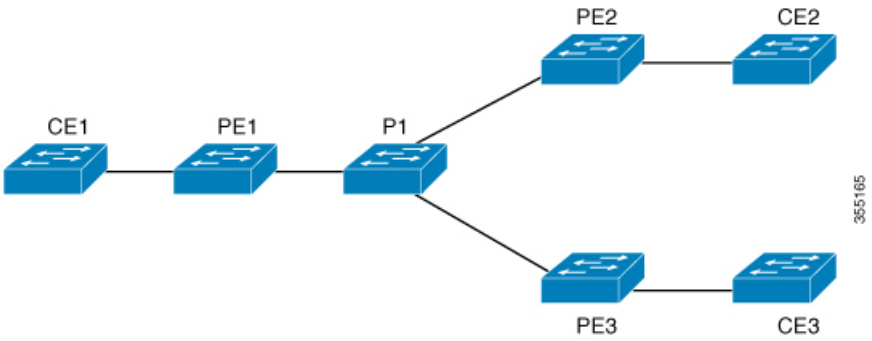
```
Device# show l2vpn atom vc detail

pseudowire100005 is up, VC status is up PW type: Ethernet
  Create time: 19:25:56, last status change time: 09:40:37
    Last label FSM state change time: 09:40:37
  Destination address: 44.254.44.44 VC ID: 2129
    Output interface: Gi1/0/9, imposed label stack {18 17}
    Preferred path: not configured
    Default path: active
    Next hop: 177.77.177.2
  Member of vfi service 2129
    Bridge-Domain id: 2129
    Service id: 0x32000003
  Signaling protocol: LDP, peer 44.254.44.44:0 up
    Targeted Hello: 1.1.1.72(LDP Id) -> 44.254.44.44, LDP is UP
    Graceful restart: configured and enabled
    Non stop routing: not configured and not enabled
     PWid FEC (128), VC ID: 2129
    Status TLV support (local/remote)         : enabled/supported
      LDP route watch                         : enabled
      Label/status state machine              : established, LruRru
      Local dataplane status received         : No fault
      BFD dataplane status received           : Not sent
      BFD peer monitor status received        : No fault
      Status received from access circuit     : No fault
      Status sent to access circuit           : No fault
      Status received from pseudowire i/f     : No fault
Status sent to network peer             : No fault
      Status received from network peer       : No fault
      Adjacency status of remote peer         : No fault
  Sequencing: receive disabled, send disabled
  Bindings
    Parameter    Local                          Remote
    ------------ ------------------------------ ------------------------------
    Label        512                            17
    Group ID     n/a                            0
    Interface
    MTU          1500                           1500
    Control word off                            off
    PW type      Ethernet                       Ethernet
    VCCV CV type 0x02                           0x02
                   LSPV [2]                       LSPV [2]
    VCCV CC type 0x06                           0x06
                   RA [2], TTL [3]                RA [2], TTL [3]
    Status TLV   enabled                        supported
  SSO Descriptor: 44.254.44.44/2129, local label: 512
  Dataplane:
    SSM segment/switch IDs: 20498/20492 (used), PWID: 2
  Rx Counters
```

```
  0 input transit packets, 0 bytes
  0 drops, 0 seq err
Tx Counters
  0 output transit packets, 0 bytes
  0 drops
```

例：テンプレートを使用した VPLS Flow Aware Transport の設定（プロトコル CLI モード）

次に、PE1 および PE2 デバイスで VPLS を設定する例を示します。

図 *4 : VPLS* トポロジ



PE1 の設定

```
Device> enable
Device# configure terminal
Device(config)# template type pseudowire mpls
Device(config-template)# encapsulation mpls
Device(config-template)# load-balance flow ip dst-ip
Device(config-template)# load-balance flow-label both
Device(config-template)# exit
Device(config)# interface Loopback0
Device(config-if)# ip address 1.1.1.30 255.255.255.255
Device(config-if)# ip ospf 1 area 0
Device(config-if)# exit
Device(config)# interface TwentyFiveGigE1/0/9
Device(config-if)# no switchport
Device(config-if)# ip address 80.0.0.30 255.255.255.0
Device(config-if)# ip ospf 1 area 0
Device(config-if)# mpls ip
Device(config-if)# exit
Device(config)# l2vpn vfi context foo
Device(config-vfi)# vpn id 2129
Device(config-vfi)# member 1.1.1.20 template mpls
Device(config-vfi)# exit
Device(config)# interface TwentyFiveGigE1/0/2
Device(config-if)# switchport mode access
Device(config-if)# switchport access vlan 100
Device(config-if)# exit
Device(config)# interface vlan 100
Device(config-vlan-config)# member vfi foo
Device(config-vlan-config)# end
```

例：VPLS BGP 自動検出の設定

次に、PE デバイスで VPLS を設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# router bgp 1000
Device(config-router)# bgp log-neighbor-changes
Device(config-router)# bgp graceful-restart
Device(config-router)# neighbor 44.254.44.44 remote-as 1000
Device(config-router)# neighbor 44.254.44.44 update-source Loopback300
Device(config-router)# address-family l2vpn vpls
Device(config-router-af)# neighbor 44.254.44.44 activate
Device(config-router-af)# neighbor 44.254.44.44 send-community both
Device(config-router-af)# exit-address-family
Device(config-router-af)# end
Device(config)# l2 vfi 2128 autodiscovery
Device(config-vfi)# vpn id 2128
Device(config-vfi)# exit
Device(config)# interface vlan 2128
Device(config-vlan-config)# no ip address
Device(config-vlan-config)# xconnect vfi 2128
!
```

例：VPLS BGP 自動検出の確認

次に、**show platform software fed sw 1 matm macTable vlan 2000** コマンドの出力例を示します。

```
Device# show platform software fed sw 1 matm macTable vlan 2000

VLAN  MAC               Type      Seq#   macHandle          siHandle           diHandle
          *a_time *e_time   ports
2000  2852.6134.05c8   0X8002    0      0xffbba312c8       0xffbb9ef938       0x5154
          0        0        Vlan2000
2000  0000.0078.9012   0X1       32627  0xffbb665ec8       0xffbb60b198       0xffbb653f98
       300      278448   Port-channel11
2000  2852.6134.0000   0X1       32651  0xffba15e1a8       0xff454c2328       0xffbb653f98
       300      63       Port-channel11
2000  0000.0012.3456   0X2000001 32655  0xffba15c508       0xff44f9ec98       0x0
          300      1        2000:33.33.33.33
Total Mac number of addresses:: 4
*a_time=aging_time(secs)  *e_time=total_elapsed_time(secs)
Type:
MAT_DYNAMIC_ADDR     0x1       MAT_STATIC_ADDR       0x2
MAT_CPU_ADDR         0x4       MAT_DISCARD_ADDR      0x8
MAT_ALL_VLANS        0x10      MAT_NO_FORWARD        0x20
MAT_IPMULT_ADDR      0x40      MAT_RESYNC            0x80
MAT_DO_NOT_AGE       0x100     MAT_SECURE_ADDR       0x200
MAT_NO_PORT          0x400     MAT_DROP_ADDR         0x800
MAT_DUP_ADDR         0x1000    MAT_NULL_DESTINATION  0x2000
MAT_DOT1X_ADDR       0x4000    MAT_ROUTER_ADDR       0x8000
MAT_WIRELESS_ADDR    0x10000   MAT_SECURE_CFG_ADDR   0x20000
MAT_OPQ_DATA_PRESENT 0x40000   MAT_WIRED_TUNNEL_ADDR 0x80000
MAT_DLR_ADDR         0x100000  MAT_MRP_ADDR          0x200000
```

```
MAT_MSRP_ADDR        0x400000  MAT_LISP_LOCAL_ADDR   0x800000
MAT_LISP_REMOTE_ADDR 0x1000000 MAT_VPLS_ADDR         0x2000000
```

次に、**show bgp l2vpn vpls all** コマンドの出力例を示します。

```
Device# show bgp l2vpn vpls all

BGP table version is 6, local router ID is 222.5.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
  r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
  x best-external, a additional-path, c RIB-compressed,
  t secondary path,
 Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1000:2128
*>   1000:2128:1.1.1.72/96
                0.0.0.0                              32768 ?
*>i  1000:2128:44.254.44.44/96
                44.254.44.44             0    100      0 ?
```

VPLS および VPLS BGP ベースの自動検出の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

表 *1 : VPLS* および *VPLS BGP* ベースの自動検出の機能情報

機能名	リリース	機能情報
VPLS および VPLS BGP ベースの自動検出の設定	Cisco IOS XE Everest 16.5.1a	VPLSにより、企業は、サービスプロバイダーから提供されるインフラストラクチャを介して、複数サイトからのイーサネットベースのLANをまとめてリンクできます。 VPLS自動検出を使用すると、各 PE デバイスで、同じ VPLS ドメインの一部である他の PE デバイスを検出できます。
VPLS レイヤ2スヌーピング：IGMP（IPv4）	Cisco IOS XE Amsterdam 17.1.1	IGMP スヌーピングは、VPLS が設定されたネットワークでサポートされます。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。