総称ルーティング カプセル化 の実装

総称ルーティング カプセル化（GRE）は、シスコが開発したトンネリング プロトコルであり、インターネット プロトコルのインターネットワーク上の仮想ポイントツーポイント リンク内にさまざまなネットワーク層プロトコルをカプセル化します。

Cisco IOS XR ソフトウェアでのリンク バンドル設定機能の履歴

リリース	変更内容
リリース 4.3.0	これらの機能は、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータでサポートされています。 • ASR 9000 Enhanced Ethernet 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 上の MPLS/L3VPNoGRE • ASR 9000 Enhanced Ethernet 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 上の RSVP/TEoGRE • ASR 9000 Enhanced Ethernet 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 上の VRF 対応 GRE • ASR 9000 Enhanced Ethernet ラインカードのみの GRE 上の L2VPN（VPWS および VPLS）

内容

この章で説明する内容は、次のとおりです。

総称ルーティング カプセル化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。これらのコマンド リファレンス ガイドには、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。

総称ルーティング カプセル化に関する情報

GRE 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。

- 「GRE の概要」（P.LSC-128）
- 「GRE の機能」（P.LSC-128）

GRE の概要

総称ルーティング カプセル化（GRE）トンネリング プロトコルでは、カプセル化によって、1 つのプロトコルから別のプロトコルにパケットを転送する、簡易で一般的なアプローチを提供します。

GRE は、ペイロード（外側の IP パケット内部の、宛先ネットワークに渡す必要がある内側のパケット）をカプセル化します。GRE トンネルのエンドポイントは、介在する IP ネットワークを通じてカプセル化パケットをルーティングすることによって、GRE トンネルを介してペイロードを送信します。途中の IP ルータは、ペイロード（内側のパケット）を解析しません。これらのルータは、GRE トンネル エンドポイントにパケットを転送する際に、外側の IP パケットだけを解析します。トンネル エンドポイントに到達すると、GRE カプセル化が削除され、ペイロードは最終的な宛先に転送されます。

MPLS ネットワークは、パブリック ネットワークを介し、ルーティング ラベルを使用してカスタマー データをトンネリングすることによって、VPN 機能を提供します。サービス プロバイダー（SP）は、相互接続されたプライベート ネットワークを持つカスタマーに MPLS L3VPN、6PE/6VPE および L2VPN サービスを提供します。

MPLS および L3VPN は、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータ上の通常のインターフェイスでサポートされます。MPLS のサポートは、プロバイダー コアが完全に MPLS に対応していない可能性があるため、ルータ間の GRE トンネルに拡張されます。

GRE の機能

サポートされる機能は次のとおりです。

- 「MPLS/L3VPN over GRE」（P.LSC-129）
- 「6PE/6VPE over GRE」（P.LSC-131）

MPLS/L3VPN over GRE

MPLS VPN over GRE 機能は、非 MPLS ネットワーク経由でマルチプロトコル ラベル スイッチング（MPLS）パケットのトンネリングを行うためのメカニズムを提供します。この機能は、MPLS over Generic Routing Encapsulation（MPLSoGRE）を使用して、MPLS パケットを IP トンネルの内部にカプセル化します。IP トンネル内の MPLS パケットのカプセル化は、非 MPLS ネットワーク上でのポイントツーポイント リンクを作成します。

L3VPN over GRE は、ゼロ個以上の MPLS ラベルを付加した後で、GRE ヘッダーおよびトンネルの宛先と送信元 IP アドレスを含むその外側の IPv4 ヘッダーに L3VPN トラフィックをカプセル化し、トンネルを通じてリモート トンネル エンドポイントに転送することを基本的に意味します。着信パケットは、純粋な IPv4 パケットまたは MPLS パケットです。着信パケットが IPv4 の場合、パケットは VRF インターフェイスを通じてトンネルに渡され、着信パケットが MPLS の場合は、MPLS インターフェイスにパケットが渡されます。IPv4 の場合、外側の IPv4 ヘッダーと GRE ヘッダーでカプセル化する前に、VRF プレフィックスに対応する VPN ラベルと、GRE トンネルの宛先の IGP プレフィックスに対応する IGP ラベルがパケットに付加されます。MPLS の場合、最上位の IGP ラベルは、GRE トンネルの宛先アドレスに対応する任意のラベルと交換されます。

PE-to-PE トンネリング

プロバイダー エッジ間（PE-to-PE）トンネリング設定によって、非 MPLS ネットワーク間の複数のカスタマー ネットワークをスケーラブルな方法で接続できます。この設定を使用して、複数のカスタマー ネットワーク宛のトラフィックは、単一の GRE トンネルから多重化されます。

（注） 類似したスケーラブルではない代替方法は、別個の GRE トンネルから各カスタマー ネットワークに接続することです（たとえば、1 つのカスタマー ネットワークを各 GRE トンネルに接続します）。

図 8 に示すように、PE デバイスは、VPN ルーティングおよび転送（VRF）番号を非 MPLS ネットワークの各側にあるカスタマー エッジ（CE）デバイスに割り当てます。

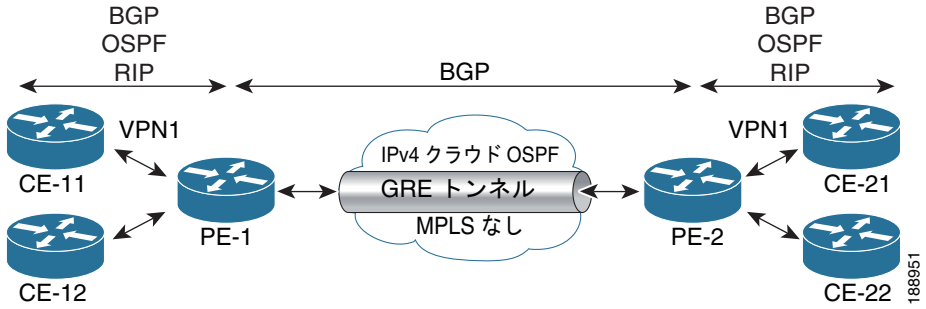
PE デバイスは、ボーダー ゲートウェイ プロトコル（BGP）、Open Shortest Path First（OSPF）、または Routing Information Protocol（RIP）などのルーティング プロトコルを、CE デバイスの背後にある IP ネットワークを学習するために使用します。CE デバイスの背後にある IP ネットワークへのルートは、関連する CE デバイスの VRF ルーティング テーブルに格納されます。

非 MPLS ネットワークのいずれかの側にある PE デバイスは、（非 MPLS ネットワーク内で動作している）ルーティング プロトコルを使用して、非 MPLS ネットワークのもう一方の側にある PE デバイスについて学習します。PE デバイス間に確立された学習ルートは、メインまたはデフォルトのルーティング テーブルに格納されます。

反対方向の PE デバイスは、BGP を使用して、PE デバイスの背後にあるカスタマー ネットワークに関連付けられたルートについて学習します。これらの学習ルートは、非 MPLS ネットワークには認識されません。

図 8 は、非 MPLS ネットワークにまたがる GRE トンネル経由で BGP ネイバー（反対方向の PE デバイス）へのスタティック ルートを定義する BGP を示しています。BGP ネイバーによって学習されたルートには GRE トンネルのネクスト ホップが含まれているため、すべてのカスタマー ネットワーク トラフィックが GRE トンネルを使用して送信されます。

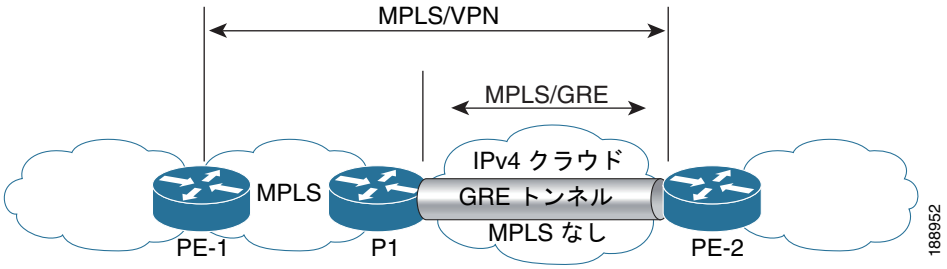
図 8　PE-to-PE トンネリング



P-to-PE トンネリング

図 9 に示すように、Provider-to-Provider Edge（P-to-PE）トンネリング設定によって、非 MPLS ネットワークで PE デバイス（P1）を MPLS セグメント（PE-2）に接続できます。この設定では、非 MPLS ネットワークの一方の側宛の MPLS トラフィックは、単一の GRE トンネル経由で送信されます。

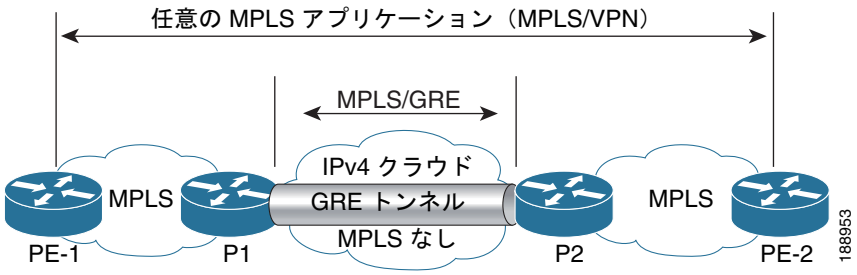
図 9　P-to-PE トンネリング



P-to-P トンネリング

図 10 に示すように、Provider-to-Provider（P-to-P）設定によって、非 MPLS ネットワークで 2 つの MPLS セグメント（P1 から P2）を接続できます。この設定では、非 MPLS ネットワークの一方の側宛の MPLS トラフィックは、単一の GRE トンネル経由で送信されます。

図 10　P-to-P トンネリング



6PE/6VPE

サービス プロバイダー（SP）は、IPv4 VPN サービスを提供するために、IPv4/MPLS バックボーンによって安定し確立されたコアを使用します。6PE/6VPE 機能は、IPv6 コアなしでこのバックボーンを介して IPv6 VPN サービスを提供するために、SP を利用します。プロバイダー エッジ（PE）ルータは v6 到達可能性および v6 ラベル配布をアドバタイズするために、MP-iBGP（マルチ プロトコル iBGP）を実行します。6PE の場合、ラベルは接続されたカスタマー エッジ（CE）ルータから学習した IPv6 プレフィックスごとに割り当てられ、6VPE の場合は、プレフィックス単位または CE/VRF レベル単位でラベルを割り当てるように PE ルータを設定できます。

6PE/6VPE over GRE

IPv4/MPLS を使用すると、SP は、IPv4 コア（IPv6 非対応）上で IPv6 トラフィックを転送できるようになりますが、MPLS over GRE は、MPLS 非対応のネットワークを介して MPLS トラフィックをトンネリングできるようにします。これらの 2 つの機能により、IPv6 コア セグメントおよび MPLS 非対応のコア セグメントを介して IPv6 トラフィックを転送できます。PE ルータのみが MPLS および IPv6（デュアル スタック）を認識する必要があります。

6PE/6VPE over GRE 機能を使用すると、IPv4 GRE トンネルを使用して、MPLS および IPv6 非対応コアを経由して、BGP ネクスト ホップを介して宛先 v6 のプレフィックスに到達するように、IPv6 VPN over MPLS 機能を提供できます。

MPLS 転送

1 つのカスタマー サイトから IPv6 トラフィックを受信すると、入力 PE デバイスは MPLS を使用して、BGP ネクスト ホップとして識別された出力 PE デバイスに向けて、バックボーンを介して IPv6 VPN パケットをトンネリングします。入力 PE デバイスは、一般的に IPv6 パケットの先頭に外部ラベルおよび内部ラベルを付加してから、出力インターフェイスにパケットを配置します。

通常の動作では、転送パス上の P デバイスは最初のラベルの先にあるフレームを調べません。P デバイスは着信ラベルを発信ラベルと交換するか、または次のデバイスが PE デバイスの場合には着信ラベルを削除します。着信ラベルの削除は、最後から 2 番めのホップのポッピングと呼ばれます。残りのラベル（BGP ラベル）は、カスタマー サイトへの出力 PE インターフェイスを識別するために使用されます。また、ラベルは、最後の P デバイスからプロトコル バージョン（IPv6）を隠します。そうしない場合、IPv6 パケットを転送することが必要になります。

P デバイスは IPv6 VPN ルートを知りません。IPv6 ヘッダーは 1 つ以上の MPLS ラベルの下に隠されたままになります。P デバイスで、送達できない MPLS カプセル化 IPv6 パケットを受信した場合のオプションは 2 つあります。P デバイスが IPv6 対応の場合、IPv6 ヘッダーを公開し、IPv6 メッセージ用のインターネット制御メッセージ プロトコル（ICMP）を構築して、MPLS カプセル化メッセージを元のパケットの送信元に送信します。P デバイスが IPv6 対応でない場合、パケットをドロップします。

6PE/6VPE over GRE

前述したように、6PE/6VPE over GRE は、GRE 上の MPLS 上で IPv6/IPv6 VPN を有効にすることを基本的に意味します。

入力 PE デバイスは、MPLS を介した 6PE/6VPE と組み合わせた IPv4 総称ルーティング カプセル化（GRE）トンネルを使用して、BGP ネクスト ホップとして識別された出力 PE デバイスに向けて、バックボーンを介して IPv6 VPN パケットをトンネリングします。

PE デバイスは、6PE/6VPE の場合と同様に、MP-iBGP セッションおよび MPLS LDP セッションを確立します。ここでの違いは、これらのセッションが GRE トンネル上で確立されることです。これは、PE が 1 IGP ホップしか離れていないことを意味します。トンネル パス内の P ルータは、単に IPv4 アドレスであるトンネル宛先にトラフィックを転送する必要があります。

IPv6 トラフィックをラベル スイッチングするために、IPv6 LSP が次のように設定されます。

- LDP セッションおよび BGP セッションの確立後、IPv4 VPN の場合と同様に、PE は CE から学習する IPv6 プレフィックスと対応する IPv6 ラベルを交換します。
- IPv6 ラベルは、ラベル スタックの最も内部の位置を占めます。
- PE IPv4 アドレスに対応する IPv4 ラベルはスタックの外側の位置を占めます。
- IPv6 トラフィックを PE1 から PE2 に転送する必要がある場合、外側の PE2 IPv4 ラベルは、トラフィックを PE2 にラベル スイッチングするために使用され、内側の IPv6 ラベルは、CE に接続されているインターフェイスからパケットを送信するために使用されます。

総称ルーティングカプセル化の設定方法

ここでは、GRE の実装に必要なタスクについて説明します。

- 「GRE トンネルの設定」（P.LSC-132）
- 「グローバル VRF の設定」（P.LSC-134）
- 「VRF インターフェイスの設定」（P.LSC-136）
- 「VRF ルーティング プロトコルの設定」（P.LSC-138）
- 「リモート PE の到達可能性のための IGP の設定」（P.LSC-139）
- 「GRE トンネル上の LDP の設定」（P.LSC-141）
- 「VPN-IPv4 ルートを交換するための MP-iBGP の設定」（P.LSC-143）

GRE トンネルの設定

GRE トンネルを設定するには、この作業を実行します。

手順の概要

1. **configure**
2. **interface tunnel-ip** *number*
3. **ipv4 address** *ipv4-address mask*
4. **tunnel source type** *path-id*
5. **tunnel destination** *ip-address*
6. **end**
   または
   commit

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** **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	**interface tunnel-ip** *number* **例：** `RP/0/RSP0/CPU0:router(config)# interface tunnel-ip 4000`	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 3	**ipv4 address** *ipv4-address subnet-mask* **例：** `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 10.1.1.1 255.255.255.0`	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 4	**tunnel source** *type path-id* **例：** `RP/0/RSP0/CPU0:router(config-if)# tunnel source TenGigE0/2/0/1`	トンネル インターフェイスの送信元を指定します。

	コマンドまたはアクション	目的
ステップ 5	**tunnel destination** *ip-address* **例：** `RP/0/RSP0/CPU0:router(config-if)# tunnel destination 145.12.5.2`	トンネルの宛先を指定します。

ステップ 6	**end**
	または
	commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-if)# end`
	または
	`RP/0/RSP0/CPU0:router(config-if)# 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、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション 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、コンフィギュレーション 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。
		• 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。

グローバル VRF の設定

グローバル VRF 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. **vrf** *vrf-name*
3. **address-family { ipv4 | ipv6 } unicast**
4. **import route-target** [*as-number:nn* | *ip-address:nn*]
5. **export route-target** [*as-number:nn* | *ip-address:nn*]
6. **exit**
7. **exit**
8. **router bgp** *as-number*
9. **vrf** *vrf-name*
10. **rd {as-number:nn | ip-address:nn | auto}**

11. **end**
    または
    commit

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** **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	**vrf** *vrf-name* **例：** `RP/0/RSP0/CPU0:router(config)# vrf vpn1`	VRF インスタンスを設定します。
ステップ 3	**address-family { ipv4 | ipv6 } unicast** **例：** `RP/0/RSP0/CPU0:router(config-vrf)# address-family { ipv4 | ipv6 } unicast`	IPv4 または IPv6 のいずれかのアドレス ファミリを指定し、アドレス ファミリの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	**import route-target** [*as-number:nn* | *ip-address:nn*] **例：** `RP/0/RSP0/CPU0:router(config-vrf-af)# import route-target 2:1`	ルート ターゲット（RT）拡張コミュニティのリストを指定します。指定されたインポート ルート ターゲット拡張コミュニティと関連付けられているプレフィックスだけが VRF にインポートされます。
ステップ 5	**export route-target** [*as-number:nn* | *ip-address:nn*] **例：** `RP/0/RSP0/CPU0:router(config-vrf-af)# export route-target 1:1`	ルート ターゲット拡張コミュニティのリストを指定します。エクスポート ルート ターゲット コミュニティは、リモート PE にアドバタイズされる際にプレフィックスと関連付けられます。リモート PE は、これらのエクスポート ルート ターゲット コミュニティと一致するインポート RT を持つ VRF に、これらのプレフィックスをインポートします。
ステップ 6	**exit** **例：** `RP/0/RSP0/CPU0:router(config-vrf-af)# exit`	VRF アドレス ファミリ設定モードを終了し、ルータを VRF 設定モードに戻します。
ステップ 7	**exit** **例：** `RP/0/RSP0/CPU0:router(config-vrf)# exit`	VRF 設定モードを終了し、ルータをグローバル コンフィギュレーション モードに戻します。
ステップ 8	**router bgp** *as-number* **例：** `RP/0/RSP0/CPU0:router(config)# router bgp 1`	自律システム番号を指定し、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。

	コマンドまたはアクション	目的
ステップ 9	**vrf** *vrf-name* **例：** `RP/0/RSP0/CPU0:router(config-bgp)# vrf vpn1`	VRF インスタンスを設定します。
ステップ 10	**rd {as-number:nn | ip-address:nn | auto}** **例：** `RP/0/RSP0/CPU0:router(config-bgp-vrf)#rd auto`	ルート識別子を設定します。

ステップ 11	**end**
	または
	commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-bgp-vrf)# end`
	または
	`RP/0/RSP0/CPU0:router(config-bgp-vrf)# 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、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション 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、コンフィギュレーション 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。
		• 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。

VRF インターフェイスの設定

VRF 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。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **vrf** *vrf-name*
4. **ipv4 address** *ipv4-address mask*
5. **end**
   または
   commit

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** **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	**interface** *type interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface tunnel-ip 100`	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	**vrf** *vrf-name* **例：** `RP/0/RSP0/CPU0:router(config-if)# vrf vrf_A`	VRF インスタンスを設定し、VRF 設定モードを開始します。
ステップ 4	**ipv4 address** *ipv4-address mask* **例：** `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 192.168.1.27 255.255.255.0`	指定したインターフェイスのプライマリ IPv4 アドレスを設定します。

ステップ 5	**end**
	または
	commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-if)# end`
	または
	`RP/0/RSP0/CPU0:router(config-if)# 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、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション 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、コンフィギュレーション 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。
		• 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。

VRF ルーティング プロトコルの設定

VRF ルーティング 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. **router ospf** *process-name*
3. **vrf** *vrf-name*
4. **router-id** {*router-id* | *type interface-path-id*}
5. **area** *area-id*
6. **interface** *type interface-path-id*
7. **end**
   または
   commit

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** **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	**router ospf** *process-name* **例：** `RP/0/RSP0/CPU0:router(config)# router ospf 109`	OSPF 設定モードを開始します。このモードでは、OSPF ルーティング プロセスの設定を行えます。
ステップ 3	**vrf** *vrf-name* **例：** `RP/0/RSP0/CPU0:router(config-ospf)# vrf vrf_1`	VPN ルーティングおよび転送（VRF）インスタンスを設定し、OSPF ルーティングの VRF 設定モードを開始します。
ステップ 4	**router-id** {*router-id* | type interface-path-id} **例：** `RP/0/RSP0/CPU0:router(config-ospf-vrf)# router-id 172.20.10.10`	OSPF ルーティング プロセスのルータ ID を設定します。
ステップ 5	**area** *area-id* **例：** `RP/0/RSP0/CPU0:router(config-ospf-vrf)# area 0`	OSPF エリアをエリア 0 として設定します。

	コマンドまたはアクション	目的
ステップ 6	**interface** type interface-path-id **例：** RP/0/RSP0/CPU0:router(config-ospf-vrf-ar)# interface GigabitEthernet 0/3/0/0	インターフェイス GigabitEthernet 0/3/0/0 をエリア 0 に関連付けます。
ステップ 7	**end** または **commit** **例：** RP/0/RSP0/CPU0:router(config-ospf-vrf-ar)# end または RP/0/RSP0/CPU0:router(config-ospf-vrf-ar)# 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、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション 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、コンフィギュレーション 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。 • 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。

リモート PE の到達可能性のための IGP の設定

リモート PE の到達可能性のために IGP 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 ospf** *process-name*
3. **router-id** {*router-id*}
4. **area** *area-id*
5. **interface tunnel-ip** *number*
6. **end**
   または
   commit

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** **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	**router ospf** *process-name* **例：** `RP/0/RSP0/CPU0:router(config)# router ospf 1`	指定したルーティング プロセスに OSPF ルーティングをイネーブルにし、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 3	**router-id** {*router-id*} **例：** `RP/0/RSP0/CPU0:router(config-ospf)# router-id 1.1.1.1`	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 4	**area** *area-id* **例：** `RP/0/RSP0/CPU0:router(config-ospf)# 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。

	コマンドまたはアクション	目的
ステップ 5	**interface tunnel-ip** *number* **例：** `RP/0/RSP0/CPU0:router(config-ospf-ar)# interface tunnel-ip 4`	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 6	**end**
	または
	commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-ospf-ar-if)# end`
	または
	`RP/0/RSP0/CPU0:router(config-ospf-ar-if)# 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、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション 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、コンフィギュレーション 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。
		• 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。

GRE トンネル上の LDP の設定

GRE トンネル上で LDP 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. **mpls ldp**
3. **router-id** {*router-id*}
4. **interface tunnel-ip** *number*
5. **end**
   または
   commit

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** **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	**mpls ldp** **例：** `RP/0/RSP0/CPU0:router(config)# 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 3	**router-id** {*router-id*} **例：** `RP/0/RSP0/CPU0:router(config-ldp)# router-id 1.1.1.1`	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 4	**interface tunnel-ip** *number* **例：** `RP/0/RSP0/CPU0:router(config-ldp)# interface tunnel-ip 4`	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 5	**end**
	または
	commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-ldp-if)# end`
	または
	`RP/0/RSP0/CPU0:router(config-ldp-if)# 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、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション 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、コンフィギュレーション 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。
		• 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。

VPN-IPv4 ルートを交換するための MP-iBGP の設定

VPN-IPv4 ルートを交換するために MP-iBGP 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** *process-name*
3. **router-id** *ip-address*
4. **neighbor** *ip-address*
5. **remote-as** *as-number*
6. **update-source** *type interface-path-id*
7. **address-family {vpnv4 | vpnv6} unicast**
8. **end**
   または
   commit

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** **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	**router bgp** *as-number* **例：** `RP/0/RSP0/CPU0:router(config)# router bgp 1`	自律システム番号を指定し、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	**router-id** *ip-address* **例：** `RP/0/RSP0/CPU0:router(config-bgp)# router-id 1.1.1.1`	指定したルータ ID で、ローカル ルータを設定します。
ステップ 4	**neighbor** *ip-address* **例：** `RP/0/RSP0/CPU0:router(config-bgp)# neighbor 4.4.4.4`	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 5	**remote-as** *as-number* **例：** `RP/0/RSP0/CPU0:router(config-bgp-nbr)#remote-as 1`	ネイバーを作成し、リモート自律システム番号を割り当てます。

	コマンドまたはアクション	目的
ステップ 6	**update-source** *type interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config-bgp-nbr)#update-source Loopback0`	ネイバーでセッションを形成するとき、特定のインターフェイスからのプライマリ IP アドレスをローカル アドレスとしてセッションで使用できます。
ステップ 7	**address-family {vpnv4 | vpnv6} unicast** **例：** `RP/0/RSP0/CPU0:router(config-bgp-nbr)# address-family vpnv4 unicast`	指定されたアドレス ファミリのアドレス ファミリ 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 8	**end**
	または
	commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-bgp-nbr)# end`
	または
	`RP/0/RSP0/CPU0:router(config-bgp-nbr)# 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、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション 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、コンフィギュレーション 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。
		• 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。

総称ルーティング カプセル化の設定例

ここでは、GRE の設定例を示します。

GRE トンネルの設定：例

次に、GRE トンネルを設定する例を示します。

```
configure
  interface tunnel-ip1
   ipv4 address 12.0.0.1 255.255.255.0
   tunnel source Loopback0
   tunnel destination 200.200.200.1
end
```

グローバル VRF の設定：例

次に、グローバル VRF を設定する例を示します。

```
configure
 vrf VRF1
  address-family ipv4 unicast
  import route-target 120.1
  export route-target 120.2
exit
exit
  router bgp120
  vrf VRF1
  rd auto
end
```

VRF インターフェイスの設定：例

次に、VRF インターフェイスを設定する例を示します。

```
configure
 interface tunnel-ip 100
  vrf VRF1
  ipv4 address 1.1.1.1 255.255.255.0
end
```

VRF ルーティング プロトコルの設定：例

次に、VRF ルーティング プロトコルを設定する例を示します。

```
configure
  router ospf109
   vrf VRF1
   router-id 172.20.10.10
   area0
   interface GigabitEthernet0/3/0/0
end
```

リモート PE の到達可能性のための IGP の設定：例

次に、リモートのプロバイダー エッジ（PE）の到達可能性のために IGP を設定する例を示します。

```
configure
  router ospf109
   router-id 172.20.10.10
   area0
   interface tunnel-ip1
end
```

GRE トンネル上の LDP の設定：例

次に GRE トンネル上で LDP を設定する例を示します。

```
configure
 mpls ldp
 router-id 172.20.10.10
  interface tunnel-ip1
 end
```

VPN-IPv4 ルートを交換するための MP-iBGP の設定：例

次に、VPN-IPv4 ルートを交換するために MP-iBGP を設定する例を示します。

```
configure
 router bgp100
  router-id 172.20.10.10
  neighbor 2.2.2.2 remote-as 100
  update-source Loopback0
  address-family vpnv4 unicast
end
```

その他の関連資料

VPLS の実装に関する詳細情報については、次を参照してください。

関連資料

関連項目	マニュアル タイトル
Cisco IOS XRL2VPN コマンド	『*Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Command Reference*』の「*Point to Point Layer 2 Services Commands*」
MPLS VPLS-related コマンド	『*Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Command Reference*』の「*Multipoint Layer 2 Services Commands*」
スタートアップ資料	*『Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide』*
VPLS ブリッジにおけるトラフィック ストーム制御	『*Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide*』の「*Traffic Storm Control under VPLS Bridges on Cisco ASR 9000 Series Routers*」
VPLS ブリッジのレイヤ 2 マルチキャスト	『*Cisco ASR 9000 Series Aggregation Services Router Multicast Configuration Guide*』の「*Layer 2 Multicast Using IGMP Snooping*」

標準

標準[1]	タイトル
draft-ietf-l2vpn-vpls-ldp-09	*『Virtual Private LAN Services Using LDP』*

1. サポートされている規格がすべて記載されているわけではありません。

MIB

MIB	MIB のリンク
—	Cisco IOS XR ソフトウェアを使用している MIB を特定してダウンロードするには、次の URL にある Cisco MIB Locator を使用し、[Cisco Access Products] メニューからプラットフォームを選択します。http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
RFC 2784	『*Generic Routing Encapsulation (GRE)*』
RFC 4448	『*Encapsulation Methods for Transport of Ethernet over MPLS Networks*』2006 年 4 月
RFC 4762	『*Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*』

シスコのテクニカル サポート

説明	リンク
シスコのテクニカル サポート Web サイトでは、製品、テクノロジー、ソリューション、技術的なヒント、およびツールへのリンクなどの、数千ページに及ぶ技術情報が検索可能です。Cisco.com に登録済みのユーザは、このページから詳細情報にアクセスできます。	http://www.cisco.com/en/US/support/index.html