トンネルの実装

このモジュールでは、Cisco IOS XE ソフトウェアで使用可能な各種のトンネリング技術について説明します。物理インターフェイスまたは仮想インターフェイスを使用するトンネル タイプについては、設定の詳細および例が記載されています。多くのトンネリング技術は、テクノロジー固有のコマンドを使用して実装されており、該当するテクノロジー モジュールへのリンクが提供されます。

トンネリングを使用すると、トランスポート プロトコル内部の任意のパケットをカプセル化できます。トンネルは仮想インターフェイスとして実装され、簡単なインターフェイスを設定できるようになっています。トンネル インターフェイスは、特定の「パッセンジャ」プロトコルまたは「トランスポート」プロトコルに関連付けられるのではなく、任意の標準的なポイントツーポイント カプセル化スキームを実装するために必要なサービスを提供するアーキテクチャです。

機能情報の確認

最新の機能情報と注意事項については、ご使用のプラットフォームとソフトウェア リリースに対応したリリース ノートを参照してください。このモジュールで説明される機能に関する情報、および各機能がサポートされるリリースの一覧については、「トンネルの実装に関する機能情報」（P.38）を参照してください。

プラットフォームのサポートおよび Cisco IOS XE ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスします。Cisco.com のアカウントは必要ありません。

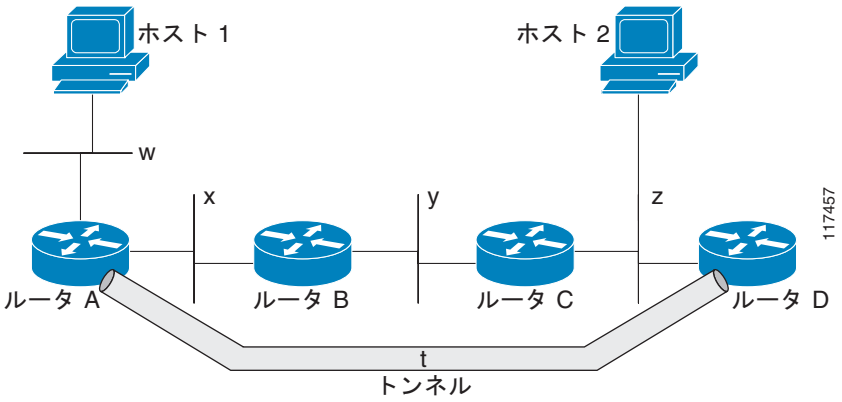
目次

トンネル実装の制約事項

- トンネル プロトコルがファイアウォールを通過でき、Access Control List（ACL; アクセス コントロール リスト）のチェックに合格できるようにすることが重要です。
- トンネル インターフェイスで帯域幅が正しく設定されていない場合、複数のポイントツーポイント トンネルがルーティング情報を使用して物理リンクを飽和させるおそれがあります。
- トンネルはシングル ホップに似ているので、ルーティング プロトコルはマルチホップ物理パスよりもトンネルを優先することがあります。ただし、このことはトンネルがシングル ホップに似ているとはいえ、マルチホップ リンクよりもスピードの遅いパスを通過する場合があるため、誤りを招くおそれがあります。トンネルは、実際に通過するリンクのように、堅牢で速いこともあれば、信頼性が低く遅いこともあります。ホップ カウントのみに基づいて決定するルーティング プロトコルでは、一連の物理リンクよりもトンネルを優先することが多くなります。トンネルはワンホップのポイントツーポイントリンクであり、最もコストの低いパスのように思われますが、実際には、もう 1 つの選択肢である物理トポロジよりも遅延に関して高コストである可能性があります。

  たとえば図 1 に示すトポロジでは、ホスト 1 からのパケットは、パス w、x、y、および z ではなく、トンネル ホップ カウントがより短いと思われるネットワーク w、t、および z を通ってホスト 2 に到達します。実際には、トンネルを通過するパケットは、ルータ A、B、および C を通って移動しますが、ルータ D まで移動してからルータ C に戻る必要があります。

図 1　　トンネルに関する注意事項：ホップ カウント



- ルーティングを慎重に設定しなければ、トンネルで再帰ルーティングの問題が発生する可能性があります。「トンネルの宛先」への最適なパスがトンネル自体を経由する場合、再帰ルーティングが原因でトンネル インターフェイスがフラップします。再帰ルーティングの問題を回避するには、次の方法を使用して、コントロールプレーン ルーティングとトンネル ルーティングを分離します。
  - 異なる自律システム番号またはタグを使用する。
  - 異なるルーティング プロトコルを使用する。
  - スタティック ルートを使用して最初のホップを無効にする（ただし、ルーティング ループには注意する）。

  トンネルの宛先に対する再帰ルーティングが発生した場合は、次のエラーが表示されます。

```
%TUN-RECURDOWN Interface Tunnel 0
temporarily disabled due to recursive routing
```

トンネルの実装に関する情報

トンネルを設定するには、次の概念について理解しておく必要があります。

トンネリングとカプセル化

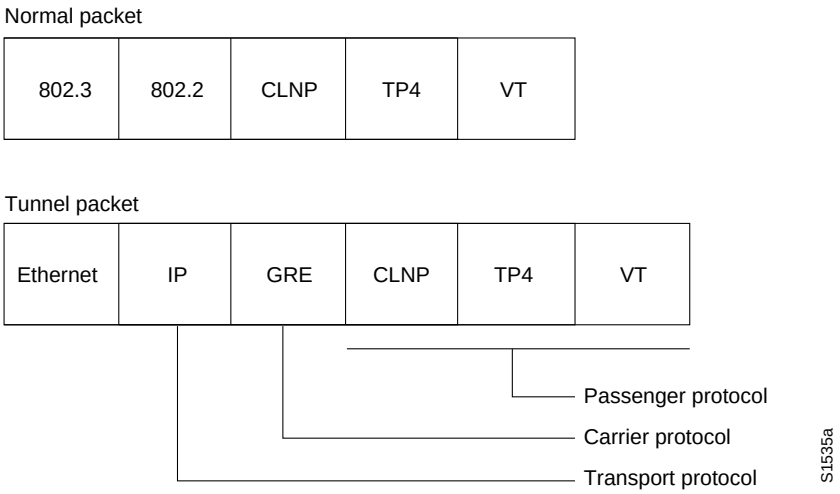
トンネルの動作のしくみを理解するためには、カプセル化とトンネリングの概念を区別することが重要です。カプセル化は、特定のプロトコル スタックの各レイヤでデータにヘッダーを追加するプロセスです。Open Systems Interconnection（OSI）リファレンス モデルは、それぞれのレイヤの上にレイヤが積み重なった 7 つのレイヤとしてネットワークの機能を説明しています。データがネットワーク上のホスト（たとえば PC）から別のホストに送信される場合、カプセル化のプロセスが使用されて、降順に並んでいる各プロトコル スタック レイヤのデータの先頭にヘッダーが追加されます。ヘッダーには、現在のレイヤのすぐ上のレイヤでカプセル化されているデータのタイプを示すデータ フィールドが含まれている必要があります。パケットがネットワークの受信側のプロトコル スタックを上っていくにつれて、各カプセル化ヘッダーは逆の順番で削除されます。

トンネリングは、異なるプロトコル内の 1 つのプロトコルからデータ パケットをカプセル化し、データ パケットを変更することなく外部ネットワークを通じて送信します。トンネリングはカプセル化とは異なり、トンネルを経由してより低いレイヤのプロトコルまたは同じレイヤのプロトコルを送信できます。トンネル インターフェイスは、仮想（または論理）インターフェイスです。さまざまなネットワークの問題を解決するために多くのタイプのトンネルが作成されていますが、トンネリングは、次の 3 つの主要コンポーネントで構成されています。

- パッセンジャ プロトコル：カプセル化の対象となるプロトコル。パッセンジャ プロトコルの例は、IPv4 および IPv6 です。
- キャリア プロトコル：カプセル化を実行するプロトコル。キャリア プロトコルの例は、Generic Routing Encapsulation（GRE; 総称ルーティング カプセル化）および Multiprotocol Label Switching（MPLS; マルチプロトコル ラベル スイッチング）です。
- トランスポート プロトコル：カプセル化されたプロトコルの送信に使用されるプロトコル。主なトランスポート プロトコルは、IP です。

図 2 に IP トンネリングの用語と概念を示します。

図 2　IP トンネリングの用語と概念



Tunnel ToS

Tunnel Type of Service（ToS; タイプ オブ サービス）では、ネットワーク トラフィックをトンネリングし、すべてのパケットを同一の特定の ToS バイト値にグループ化できます。ToS バイト値および Time-to-Live（TTL）ホップカウント値は、ルータ上の IP トンネル インターフェイス向けにトンネル パケットのカプセル化 IP ヘッダーで設定されます。Tunnel ToS 機能は、Cisco Express Forwarding（CEF; シスコ エクスプレス フォワーディング）、ファスト スイッチング、およびプロセス スイッチングでサポートされています。

ToS バイト値および TTL バイト値は、RFC 791 で定義されています。RFC 791 で定義されているように、RFC 2474 および RFC 2780 では ToS バイト値の使用を廃止しました。RFC 791 では、ToS バイトのビット 6 および 7（最初の 2 つの最下位ビット）は将来使用するために予約されており、0 に設定する必要があることが指定されています。Cisco IOS XE Release 2.1 では、Tunnel ToS 機能はこの標準に準拠していません。ビット 6 および 7 を含むすべての ToS バイト値を設定し、パケットの ToS バイトが準拠する RFC 標準を決定できるようになっています。

総称ルーティング カプセル化

Generic Routing Encapsulation（GRE; 総称ルーティング カプセル化）は、RFC 2784 で定義されています。GRE はキャリア プロトコルです。さまざまな基盤となるトランスポート プロトコルと共に使用でき、各種のパッセンジャ プロトコルを伝送できます。RFC 2784 は、トランスポート プロトコルおよびパッセンジャ プロトコルとして、GRE と IPv4 の併用についても記述しています。Cisco IOS XE ソフトウェアは、次のようにさまざまな組み合わせのパッセンジャ プロトコルおよびトランスポート プロトコルと共に使用するキャリア プロトコルとして GRE をサポートしています。

- GRE を IPv4 ネットワーク上で使用（GRE/IPv4）: GRE はキャリア プロトコル、IPv4 はトランスポート プロトコルです。これは、最も一般的なタイプの GRE トンネルです。設定の詳細については、「GRE トンネルの設定」（P.14）を参照してください。Cisco IOS XE ソフトウェアは、AppleTalk、IPX、IPv4、IPv6 など GRE/IPv4 向けの多くのパッセンジャ プロトコルをサポートしています。GRE/IPv4 で使用するパッセンジャ プロトコルとしての IPv6 の詳細については、「IPv6 トラフィックに対する GRE/IPv4 トンネルのサポート」（P.5）を参照してください。

- GRE を IPv6 ネットワーク上で使用（GRE/IPv6）：GRE はキャリア プロトコル、IPv6 はトランスポート プロトコルです。Cisco IOS XE ソフトウェアは、GRE/IPv6 で使用するパッセンジャ プロトコルとして IPv4 および IPv6 をサポートしています。GRE/IPv6 で使用するパッセンジャ プロトコルとしての IPv4 および IPv6 の設定の詳細については、「GRE/IPv6 トンネルの設定」（P.18）を参照してください。

GRE トンネルについては、次の各項を参照してください。

- 「GRE トンネルの IP 送信元および宛先の VRF メンバーシップ」（P.5）
- 「IPv6 トラフィックに対する GRE/IPv4 トンネルのサポート」（P.5）

GRE トンネルの IP 送信元および宛先の VRF メンバーシップ

GRE トンネルの IP 送信元および宛先の VRF メンバーシップ機能では、トンネルの送信元および宛先が任意の Virtual Private Network（VPN; バーチャル プライベート ネットワーク）Routing and Forwarding（VRF; VPN ルーティングおよび転送）テーブルに属するように設定できます。VRF テーブルは、各 VPN のルーティング データを保存します。VRF テーブルは、Network Access Server（NAS; ネットワーク アクセス サーバ）に接続されるカスタマー サイトの VPN メンバーシップを定義します。各 VRF テーブルは、IP ルーティング テーブル、生成された CEF テーブル、およびルーティング テーブルに含まれる情報を制御するガイドラインとプロトコル パラメータで構成されます。

Cisco IOS XE Release 2.2 よりも前の GRE IP トンネルでは､IP トンネルの宛先がグローバル ルーティング テーブルに含まれている必要がありました。この機能を実装することにより、トンネルの送信元および宛先が任意の VRF に属するように設定できます。トンネルの宛先へのルートが定義されていない場合、トンネルは、既存の GRE トンネルと同様にディセーブルになります。

IPv6 トラフィックに対する GRE/IPv4 トンネルのサポート

IPv6 トラフィックは、標準的なポイントツーポイントのカプセル化スキームの実装に必要なサービスを提供するように設計されている標準 GRE トンネリング技術を使用して、IPv4 GRE トンネルを介して伝送できます。IPv6 を手動設定したトンネルと同様、GRE トンネルは、各リンクに個別のトンネルを持つ 2 つのポイント間のリンクです。トンネルは、特定のパッセンジャ プロトコルまたはトランスポート プロトコルに関連付けられてはいませんが、この場合は IPv6 はパッセンジャ プロトコル、GRE はキャリア プロトコル、IPv4 はトランスポート プロトコルです。

GRE トンネルの主な用途は、2 つのエッジ ルータ間またはエッジ ルータとエンド システムの間に標準的でセキュアな通信を必要とする、安定した接続状態を確保することです。エッジ ルータとエンド システムは、デュアル スタック実装にする必要があります。

GRE には、パッセンジャ プロトコルを識別するプロトコル フィールドがあります。GRE トンネルでは、Intermediate System to Intermediate System（IS-IS）または IPv6 をパッセンジャ プロトコルとして指定でき、IS-IS トラフィックと IPv6 トラフィックをともに同じトンネルを介して送出できます。GRE にプロトコル フィールドがない場合、トンネルが IS-IS または IPv6 のどちらのパケットを伝送しているかを区別することはできません。つまり、GRE プロトコル フィールドを使用することで、GRE 内で IS-IS および IPv6 をトンネリングできるようになります。

EoMPLS over GRE

Ethernet over MPLS（EoMPLS）は、レイヤ 3 の MPLS ネットワークを経由したレイヤ 2 トラフィックのトンネリングを可能にするトンネリング メカニズムです。EoMPLS はレイヤ 2 トンネリングとしても知られています。

EoMPLS は、レイヤ 2 の長距離拡張を効果的に促進します。EoMPLS over GRE は、ハードウェア ベースのスイッチド トンネルとして GRE トンネルを作成し、GRE トンネル内で EoMPLS をカプセル化できるようにします。GRE 接続が 2 つのコア ルータ間で確立され、MPLS Label Switched Path（LSP; ラベル スイッチド パス）がトンネリングされます。

GRE カプセル化は、転送前に追加されるヘッダー情報を持つパケットを定義するために使用されます。カプセル化解除は、パケットが宛先トンネルのエンドポイントに到着したときに追加ヘッダー情報を削除するプロセスです。

パケットが GRE トンネルを経由して転送されると、パケットの先頭に 2 つの新しいヘッダーが追加されます。したがって、新しいペイロードの内容は変更されます。カプセル化が行われると、元のデータ ペイロードおよび独立した IP ヘッダーが GRE ペイロードとなります。GRE ヘッダーはパケットに追加されて、プロトコル タイプに関する情報を提供します。再計算されたチェックサムに関する情報も提供します。新しい IP ヘッダーは、GRE ヘッダーの先頭にも追加されます。この IP ヘッダーには、トンネルの宛先 IP アドレスが含まれます。

GRE ヘッダーは、ヘッダーがトンネルに入る前に IP、L2VPN、L3VPN などのパケットに追加されます。カプセル化されたパケットを受信する、パス沿いにあるすべてのルータは、新しい IP ヘッダーを使用してトンネル エンドポイントへのパケットの到達方法を決定します。

IP 転送では、新しい IP ヘッダーおよび GRE ヘッダーがトンネルの宛先エンドポイントに到着すると、これらのヘッダーはパケットから削除され、元の IP ヘッダーが使用されて最終の宛先にパケットが転送されるようになります。

EoMPLS over GRE 機能は、トンネルの宛先でパケットから新しい IP ヘッダーおよび GRE ヘッダーを削除し、MPLS ラベルを使用して適切なレイヤ 2 接続回線またはレイヤ 3 VRF へとパケットを転送します。

次の項のシナリオでは、Provider Edge（PE; プロバイダー エッジ）またはプロバイダー（P）ルータでの GRE 展開における L2VPN および L3VPN について説明します。

- PE to PE GRE トンネル
- P to P GRE トンネル
- PE to P GRE トンネル

PE to PE GRE トンネル

PE to PE GRE トンネルのシナリオでは、ユーザは、通常、コアのどの部分も MPLS に移行させてはいませんが、EoMPLS および基本的な MPLS VPN サービスの提供を希望しています。したがって、MPLS ラベル付きトラフィックの GRE トンネリングは、PE 間で実行されます。これは、ユーザのネットワークで使用される最も一般的なシナリオです。

P to P GRE トンネル

P to P GRE トンネルのシナリオは、MPLS が PE ルータと P ルータとの間でイネーブルになっているものの、ネットワーク コアに MPLS 対応ルータまたは IP 暗号化ボックスが存在しないというシナリオです。このシナリオでは、MPLS ラベル付きパケットの GRE トンネリングは IP ルータ間で実行されます。

PE to P GRE トンネル

PE to P GRE トンネルのシナリオでは、P to P ノードは MPLS 対応ですが、その一方で GRE トンネリングが PE to P 非 MPLS ネットワーク セグメントの間で実行されるネットワークです。

上記のシナリオを展開するには、次のリストに示される機能が必要です。

GRE に固有の機能

- トンネル エンドポイントは、ループバック インターフェイスまたは物理インターフェイス
- キープアライブ タイマーの期限が切れると、エンドポイント単位で設定可能なトンネルのキープアライブ タイマー パラメータおよび Syslog メッセージを生成
- トンネル障害およびトンネルを使用した Interior Gateway Protocol（IGP）に対する Bidirectional Forwarding Detection（BFD; 双方向フォワーディング検出）のサポート
- GRE トンネル全体における IGP ロード シェアリングのサポート
- GRE トンネル全体における IGP 冗長性のサポート
- GRE トンネル全体におけるフラグメンテーションのサポート
- ジャンボ フレーム通過機能のサポート
- すべての IGP コントロール プレーン トラフィックのサポート
- トンネル全体における IP ToS 保存のサポート
- トンネルは、ATM、Gig、Packet over SONET（POS）、TenGig などエンドポイントの物理インターフェイス タイプとは無関係
- 最大 100 の GRE トンネルのサポート

EoMPLS に固有の機能

- ポート モードの EoMPLS
- VLAN モードの EoMPLS
- 擬似配線の冗長性
- Any Transport over MPLS（AToM）シーケンス
- トンネル選択および特定の擬似配線を GRE トンネルにマップする機能
- IGP ロード シェアリングおよび冗長性
- 最大 200 の EoMPLS Virtual Circuit（VC; 仮想回線）のサポート

MPLS VPN に固有の機能

- IPv4 VRF での PE ロールのサポート
- すべての PE to Customer Edge（CE; カスタマー エッジ）プロトコルのサポート
- 複数のトンネルを経由したロード シェアリングおよび単一トンネルと等コストの IGP パス
- 単一トンネルと非等コストの IGP パスによる冗長性のサポート
- MPLS ラベルの expression（EXP）ビット フィールドにコピーされてから、GRE パケットの外部 IPv4 ToS フィールドの precedence ビットにコピーされる IP precedence 値のサポート

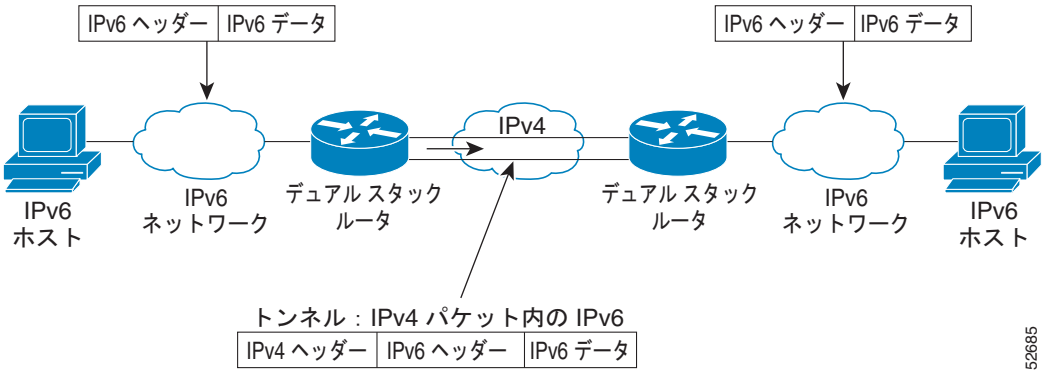
EoMPLS over GRE の設定シーケンスの例については、「EoMPLS over GRE の設定：例」（P.31）の項を参照してください。EoMPLS over GRE の詳細については、『Deploying and Configuring MPLS Virtual Private Networks In IP Tunnel Environments』を参照してください。

IPv6 向けオーバーレイ トンネル

オーバーレイ トンネリングでは、IPv4 パケット内で IPv6 パケットをカプセル化して、IPv4 インフラストラクチャ（コア ネットワークまたはインターネット）へ伝送します（図 3 を参照）。オーバーレイ トンネルを使用することで、独立した IPv6 ネットワークとの間で IPv4 インフラストラクチャをアップグレードしなくても、そのネットワークと通信できます。オーバーレイ トンネルは、境界ルータ間または境界ルータとホストの間で設定できますが、トンネルのエンドポイントがともに IPv4 プロトコル スタックおよび IPv6 プロトコル スタックを両方ともサポートしていることが必要となります。Cisco IOS XE IPv6 は現在、次のタイプのオーバーレイ トンネリング メカニズムをサポートしています。

- 6to4
- 総称ルーティング カプセル化
- Intra-Site Automatic Tunnel Addressing Protocol（ISATAP）
- IPv4 互換
- 手動

図 3　　オーバーレイ トンネル



（注） オーバーレイ トンネルにより、インターフェイスの Maximum Transmission Unit（MTU; 最大伝送ユニット）が 20 オクテット減少します（ただし、基本 IPv4 パケット ヘッダーにオプション フィールドが含まれていないことを前提とします）。オーバーレイ トンネルを使用するネットワークは、トラブルシューティングが困難です。したがって、独立した IPv6 ネットワークに接続するオーバーレイ トンネルは、最終的な IPv6 ネットワーク アーキテクチャと見なしてはいけません。オーバーレイ トンネルの使用は、IPv4 プロトコル スタックおよび IPv6 プロトコル スタックの両方、または IPv6 プロトコル スタックだけをサポートするネットワークに対する過渡的な技術と考える必要があります。

表 1 を使用して、IPv4 ネットワークを介して IPv6 パケットを伝送するために設定するトンネルのタイプを決定します。

表 1　IPv4 ネットワークを介して IPv6 パケットを伝送する際の推奨されるトンネル タイプの使用方法

トンネリングのタイプ	推奨される使用方法	使用方法に関する注釈
6to4	独立した IPv6 サイトへの接続に使用可能なポイントツーマルチポイント トンネル。	サイトは、プレフィクス 2002::/16 を使用します。
GRE/IPv4	サイト内またはサイト間で使用可能な単純なポイントツーポイント トンネル。	トンネルは、IPv6、CLNS、およびその他多数のパケット タイプを伝送可能です。
ISATAP	サイト内のシステム間の接続に使用可能なポイントツーマルチポイント トンネル。	サイトは任意の IPv6 ユニキャスト アドレスを使用できます。
手動	サイト内またはサイト間で使用可能な単純なポイントツーポイント トンネル。	トンネルは、IPv6 パケットのみ伝送可能です。

個々のトンネル タイプについては以降で詳しく説明します。シスコは、ユーザが実装を希望する特定のトンネル タイプについての情報を確認して理解することを推奨します。必要なトンネルのタイプについて十分理解できたら、表 2 のトンネル設定パラメータの概要を参照して役に立つ情報を確認してください。

表 2　トンネリング タイプ別オーバーレイ トンネルの設定パラメータ

オーバーレイ トンネリング タイプ	オーバーレイ トンネルの設定パラメータ			
	トンネル モード	トンネルの送信元	トンネルの宛先	インターフェイスのプレフィクス/アドレス
6to4	ipv6ip 6to4	IPv4 アドレスまたは IPv4 が設定されているインターフェイスのリファレンス	不要。これらは、すべてポイントツーマルチポイント トンネリング タイプです。IPv4 の宛先アドレスは、パケット単位で IPv6 の宛先から算出されます。	IPv6 アドレス。プレフィクスは、トンネル送信元の IPv4 アドレスを埋め込む必要があります。
GRE/IPv4	gre ip		IPv4 アドレス。	IPv6 アドレス。
ISATAP	ipv6ip isatap		不要。これらは、すべてポイントツーマルチポイント トンネリング タイプです。IPv4 の宛先アドレスは、パケット単位で IPv6 の宛先から算出されます。	eui-64 修正版 IPv6 形式のプレフィクス。IPv6 アドレスは、プレフィクスおよびトンネル送信元の IPv4 アドレスから生成されます。
手動	ipv6ip		IPv4 アドレス。	IPv6 アドレス。

手動設定された IPv6 トンネル

手動設定されたトンネルは、IPv4 バックボーン上の 2 つの IPv6 ドメイン間の固定リンクと同等です。主な用途は、2 つのエッジ ルータ間またはエンド システムとエッジ ルータの間に標準的でセキュアな通信を必要とする、安定した接続状態を確保したり、リモートの IPv6 ネットワークへ接続したりすることです。

IPv6 アドレスはトンネル インターフェイスで手動で設定します。また、手動設定された IPv4 アドレスはトンネルの送信元および宛先に割り当てられます。手動設定されたトンネルの両端のホストまたはルータは、IPv4 プロトコル スタックおよび IPv6 プロトコル スタックをサポートする必要があります。手動設定されたトンネルは、境界ルータ間または境界ルータとホストの間で設定できます。CEF スイッチングは、手動設定された IPv6 トンネルに使用できます。または、プロセス スイッチングが必要な場合は、CEF スイッチングをディセーブルにできます。

自動 6to4 トンネル

自動 6to4 トンネルにより、独立した IPv6 ドメインを IPv4 ネットワークを介してリモートの IPv6 ネットワークへ接続できます。自動 6to4 トンネルと手動設定されたトンネルとの主な違いは、トンネルがポイントツーポイントではなく、ポイントツーマルチポイントであることです。自動 6to4 トンネルでは、ルータは IPv4 インフラストラクチャを仮想 NonBroadcast MultiAccess（NBMA; 非ブロードキャスト マルチアクセス）リンクとして処理するため、ペアでは設定されません。IPv6 アドレスに埋め込まれている IPv4 アドレスは、自動トンネルのもう一端を検索するのに使用されます。

自動 6to4 トンネルは、独立した IPv6 ネットワーク内の境界ルータ上で設定できます。これにより、IPv4 インフラストラクチャ上の別の IPv6 ネットワーク内の境界ルータへのトンネルがパケット単位で作成されます。トンネルの宛先は、IPv6 アドレスから抽出された境界ルータの IPv4 アドレスによって決定され、プレフィクス 2002::/16 で始まる、2002:*border-router-IPv4-address*::/48 という形式になります。埋め込まれた IPv4 アドレスの後に 16 ビットが続き、サイト内のネットワークに番号付けをするのに使用できます。6to4 トンネルの各終端の境界ルータは、IPv4 プロトコル スタックおよび IPv6 プロトコル スタックの両方をサポートする必要があります。6to4 トンネルは、境界ルータ間または境界ルータとホストの間で設定されます。

6to4 トンネルの最も単純な展開シナリオは、複数の IPv6 サイトに相互接続し、各サイトに共有 IPv4 ネットワークへの接続を少なくとも 1 つ設定することです。この IPv4 ネットワークは、グローバル インターネットまたは企業バックボーンである可能性があります。主な要件は、各サイトがグローバルに一意の IPv4 アドレスを持つことです。Cisco IOS XE ソフトウェアはこのアドレスを使用して、グローバルに一意の 6to4/48 IPv6 プレフィクスを作成します。他のトンネル メカニズムと同様に、IPv4 および IPv6 の両方に対してホスト名と IP アドレスとの間のマッピングを行う Domain Name System（DNS; ドメインネーム システム）内の対応するエントリにより、アプリケーションは必要なアドレスを選択できます。

ISATAP トンネル

Intra-Site Automatic Tunnel Addressing Protocol（ISATAP）は、基盤となる IPv4 ネットワークを IPv6 に対する NBMA リンク層として使用する、自動オーバーレイ トンネリング メカニズムです。ISATAP は、ネイティブの IPv6 インフラストラクチャがまだ使用可能になっていない（希薄 IPv6 ホストがテスト用に展開されている場合などの）*サイト内*の IPv6 パケットを転送することを目的として設計されています。ISATAP トンネルにより、サイト内の個々の IPv4/IPv6 デュアルスタック ホストが同じ仮想リンク上の他のホストと通信でき、基本的に IPv4 インフラストラクチャを使用して IPv6 ネットワークを作成できます。

ISATAP ルータでは、ISATAP サイトに対して標準的なルータ アドバタイズメント ネットワーク設定がサポートされます。この機能により、クライアントは、イーサネットに接続した場合に行うと思われる設定をクライアント自身に自動的に実行できます。また、サイト外で接続を行えるよう設定することもできます。ISATAP は、任意のユニキャスト IPv6 プレフィクス（/64）で構成される明確な IPv6 アドレス形式を使用します。この形式は、リンクローカルまたはグローバル（6to4 プレフィクスを含む）にすることができ、IPv6 ルーティングをローカルまたはインターネットでイネーブルにできます。IPv4 アドレスは IPv6 アドレスの最後の 32 ビットで符号化され、自動 IPv6-in-IPv4 トンネリングをイネーブルにします。

ISATAP トンネリング メカニズムは他の自動トンネリング メカニズム（IPv6 6to4 トンネリングなど）と同様ですが、ISATAP は*サイト間*ではなく、*サイト内*の IPv6 パケットを転送するよう設計されています。

ISATAP は、64 ビットの IPv6 プレフィクスおよび 64 ビットのインターフェイス識別子を含むユニキャスト アドレスを使用します。インターフェイス識別子は、修正版 EUI-64 形式で作成され、この形式では、最初の 32 ビットにはこのアドレスが IPv6 ISATAP アドレスであることを示す 000:5EFE という値が含まれます。表 3 に、ISATAP アドレスのレイアウトを示します。

表 3　　ISATAP アドレスの例

64 ビット	32 ビット	32 ビット
リンク ローカルまたはグローバル IPv6 ユニキャスト プレフィクス	0000:5EFE	ISATAP リンクの IPv4 アドレス

表 3 に示すとおり、ISATAP アドレスは IPv6 プレフィクスおよび ISATAP インターフェイス識別子で構成されます。このインターフェイス識別子には、基盤となる IPv4 リンクの IPv4 アドレスが含まれます。プレフィクスが 2001:0DB8:1234:5678::/64 で、埋め込まれた IPv4 アドレスが 10.173.129.8 の場合、実際の ISATAP アドレスがどのようになるかを次の例に示します。ISATAP アドレスでは、IPv4 アドレスは 0AAD:8108 のような 16 進数として表現されます。

2001:0DB8:1234:5678:0000:5EFE:0AAD:8108

Path MTU Discovery（PMTUD）

Path MTU Discovery（PMTUD）は、GRE または IP-in-IP トンネル インターフェイスでイネーブルにできます。トンネル インターフェイスで PMTUD（RFC 1191）がイネーブルの場合、ルータは GRE（または IP-in-IP）トンネル IP パケットに対して PMTUD 処理を実行します。ルータは、トンネルに入ってくる元のデータの IP パケットに対して常に PMTUD 処理を実行します。PMTUD がイネーブルの場合、Don't Fragment（DF）ビットがすべてのパケットに設定されるため、トンネルを通過するパケットに対してはパケットのフラグメンテーションは許可されません。トンネルに入ったパケットがそのパケットの MTU 値よりも小さい MTU 値を持つリンクを検出すると、パケットは廃棄され、パケットの送信元に Internet Control Message Protocol（ICMP; インターネット制御メッセージ プロトコル）メッセージが返されます。このメッセージは、フラグメンテーションが要求されたこと（しかし許可されなかったこと）およびパケットが廃棄される原因となったリンクの MTU を示します。

（注） トンネル インターフェイスの PMTUD は、トンネル エンドポイントがトンネルのパスでルータによって生成される ICMP メッセージを受信できることを要求します。ファイアウォール接続を通じて PMTUD を使用する前に、ICMP メッセージが受信できることを確認してください。

トンネルのパケットに対して PMTUD をイネーブルにするには、**tunnel path-mtu-discovery** コマンドを使用し、トンネルの PMTUD パラメータを確認するには、**show interfaces tunnel** コマンドを使用します。PMTUD が動作するトンネル インターフェイスは現在、GRE および IP-in-IP だけです。

トンネル用 QoS オプション

トンネル インターフェイスは、物理インターフェイスと同じ多数の Quality of Service（QoS）機能をサポートします。QoS により、ミッション クリティカルなトラフィックのパフォーマンスを確実に受け入れ可能なレベルにする方法が提供されます。トンネル用 QoS オプションでサポートされる項目には、Generic Traffic Shaping（GTS; ジェネリック トラフィック シェーピング）のトンネル インターフェイスへの直接適用や、Modular QoS Command-Line Interface（MQC; モジュラ QoS コマンドライン インターフェイス）を使用したクラスベースのシェーピングなどが含まれます。またトンネル インターフェイスは、クラスベースのポリシングもサポートしますが、Committed Access Rate（CAR; 専用アクセス レート）はサポートしません。

GRE トンネルでは、ルータは、ToS バイトの IP precedence ビット値をトンネルまたは内部パケットをカプセル化している GRE IP ヘッダーにコピーできます。トンネルのエンドポイント間の中間ルータは、IP precedence 値を使用して、QoS 機能（ポリシー ルーティング、Weighted Fair Queuing（WFQ; 重み付け均等化キューイング）、Weighted Random Early Detection（WRED; 重み付けランダム早期検出）など）向けにパケットを分類できます。

トンネルまたは暗号化ヘッダーによってパケットがカプセル化されている場合、QoS 機能は元のパケットのヘッダーを調べてパケットを正しく分類することができません。同じトンネルを通過するパケットは、同じトンネル ヘッダーを持つため、物理インターフェイスが輻輳している場合、パケットは同等に扱われます。ただしトンネルのパケットは、トンネリング前に分類でき、QoS の事前分類機能を使用することで、トンネル インターフェイス上またはクリプト マップ上で暗号化を行うことができます。

（注） クラスベースのシェーピング内の Class-based WFQ（CBWFQ; クラスベース WFQ）は、マルチポイント インターフェイスではサポートされません。

トンネル インターフェイスでの一部の QoS 機能の実装方法の例については、「トンネル インターフェイスにおける QoS オプションの設定：例」（P.34）を参照してください。

トンネルの実装方法

ここでは、次の作業について説明します。

- 「トンネル タイプの決定」（P.13）（必須）
- 「GRE トンネルの設定」（P.14）（任意）
- 「GRE/IPv6 トンネルの設定」（P.18）（任意）
- 「GRE トンネルの IP 送信元および宛先の VRF メンバーシップの設定」（P.20）（任意）
- 「IPv6 トンネルの手動設定」（P.21）（任意）
- 「6to4 トンネルの設定」（P.23）（任意）
- 「ISATAP トンネルの設定」（P.25）（任意）
- 「トンネルの設定と動作の確認」（P.27）（任意）

トンネル タイプの決定

トンネルを設定する前に、作成するトンネルのタイプを決定する必要があります。

手順の概要

- パッセンジャ プロトコルを決定します。
- 必要に応じて、**tunnel mode** コマンド キーワードを決定します。

手順の詳細

1. パッセンジャ プロトコルを決定します。パッセンジャ プロトコルとは、カプセル化の対象となるプロトコルです。
2. 必要に応じて、**tunnel mode** コマンド キーワードを決定します。

表 4 に、**tunnel mode** コマンドで使用する適切なキーワードを決定する方法を示します。次の作業では、**tunnel mode** コマンドに関連するキーワードだけが示されます。

- GRE/IPv6 トンネルの設定
- 手動 IPv6 トンネルの設定
- 6to4 トンネルの設定
- ISATAP トンネルの設定

表 4　トンネル モードのコマンド キーワードの決定

キーワード	目的
dvmrp	ディスタンス ベクトル マルチキャスト ルーティング プロトコルのカプセル化の使用を指定するには、**dvmrp** キーワードを使用します。
gre ip	IP での GRE カプセル化の使用を指定するには、**gre ip** キーワードを使用します。
gre ipv6	IPv6 での GRE カプセル化の使用を指定するには、**gre ipv6** キーワードを使用します。
ipip [**decapsulate-any**]	IP-in-IP カプセル化の使用を指定するには、**ipip** キーワードを指定します。オプションの **decapsulate-any** キーワードは、あるトンネル インターフェイスの任意の数の IP-in-IP トンネルを終了させます。このトンネルは発信トラフィックを伝送しませんが、任意の数のリモート トンネル エンドポイントは、このように設定されたトンネルを宛先として使用できることに注意してください。
ipv6	IPv6 での汎用パケット トンネリングの使用を指定するには、**ipv6** キーワードを使用します。

表 4　トンネル モードのコマンド キーワードの決定（続き）

キーワード	目的
ipv6ip	IPv6 をパッセンジャ プロトコルとして使用し、IPv4 をキャリア（カプセル化）プロトコルおよびトランスポート プロトコルの両方として使用することを指定するには、**ipv6ip** キーワードを使用します。追加のキーワードを使用しない場合は、手動 IPv6 トンネルが設定されます。追加のキーワードを使用して、IPv4 互換、6to4、または ISATAP の各トンネルを指定できます。
mpls	Traffic Engineering（TE; トラフィック エンジニアリング）トンネルの設定に MPLS の使用を指定するには、**mpls** キーワードを使用します。

次の作業

- IP データ パケットを伝送するようにトンネルを設定するには、「GRE トンネルの設定」（P.14）に進みます。
- IPv6 データ パケットを伝送するようにトンネルを設定するには、「IPv6 向けオーバーレイ トンネル」（P.8）を確認し、次の作業のいずれかに進みます。
  - 「GRE/IPv6 トンネルの設定」（P.18）
  - 「IPv6 トンネルの手動設定」（P.21）
  - 「6to4 トンネルの設定」（P.23）
  - 「ISATAP トンネルの設定」（P.25）

GRE トンネルの設定

GRE トンネルを設定するには、次の作業を実行します。トンネル インターフェイスを使用して、通常プロトコルをサポートしないネットワークへプロトコル トラフィックを通過させます。トンネルを構築するには、トンネル インターフェイスを 2 つのルータそれぞれで定義し、そのトンネル インターフェイスが互いを参照することが必要です。各ルータでは、トンネル インターフェイスはレイヤ 3 アドレスを使用して設定する必要があります。トンネルのエンドポイント、トンネル送信元、およびトンネル宛先を定義して、トンネルのタイプを選択する必要があります。オプションの手順を実行して、トンネルをカスタマイズできます。

必ずトンネルの両側にルータを設定するようにしてください。トンネルの片側だけが設定されている場合、（キープアライブが設定されていない限り）トンネル インターフェイスはアップした状態になっていますが、トンネルに入ったパケットは廃棄されます。

GRE トンネル キープアライブ

キープアライブ パケットは、IP カプセル化された GRE トンネルを介して送信されるよう設定できます。キープアライブが送信されるレートと、インターフェイスが非アクティブになるまでデバイスが応答なしでキープアライブ パケットの送信を続行する回数を指定できます。GRE キープアライブ パケットは、トンネルの両側または片側のみのどちらからでも送信できます。

前提条件

この作業でトンネルの送信元として使用する物理インターフェイスがアップしており、適切な IP アドレスを使用して設定されていることを確認します。ハードウェアの技術的説明およびインターフェイスの取り付けに関する情報については、ご使用の製品のハードウェアの取り付けおよび設定に関するマニュアルを参照してください。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **bandwidth** *kb/s*
5. **keepalive** [*period* [*retries*]]
6. **tunnel source** {*ip-address* | *interface-type interface-number*}
7. **tunnel destination** {*hostname* | *ip-address*}
8. **tunnel key** *key-number*
9. **tunnel mode** {**gre ip** | **gre multipoint**}
10. **ip mtu** *bytes*
11. **ip tcp mss** *mss-value*
12. **tunnel path-mtu-discovery** [**age-timer** {*aging-mins* | **infinite**}]
13. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `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 3	**interface** *type number* **例：** `Router(config)# interface tunnel 0`	インターフェイスのタイプと番号を指定し、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。 • トンネルを設定するには、*type* 引数に **tunnel** を使用します。

	コマンドまたはアクション	目的
ステップ 4	**bandwidth** *kb/s* **例：** `Router(config-if)# bandwidth 1000`	インターフェイスに対する現在の帯域幅を設定し、上位レベル プロトコルと通信します。パケットの送信に使用されるトンネル帯域幅を指定します。 • 帯域幅をキロビット/秒単位（kb/s）で設定するには、*kb/s* 引数を使用します。 **（注）** これはルーティング パラメータのみのため、物理インターフェイスには影響を及ぼしません。トンネル インターフェイスのデフォルトの帯域幅設定は 9.6 kb/s です。トンネルの帯域幅を適切な値に設定する必要があります。
ステップ 5	**keepalive** [*period* [*retries*]] **例：** `Router(config-if)# keepalive 3 7`	（任意）トンネル インターフェイス プロトコルがダウン状態になるまで、デバイスが応答なしでキープアライブ パケットの送信を続行する回数を指定します。 • GRE キープアライブ パケットは、トンネルの片側または両側のどちらででも設定できます。 • GRE キープアライブをトンネルの両側で設定した場合、リンクの各側の *period* 引数および *retries* 引数は異なる値に設定できます。 **（注）** このコマンドがサポートされるのは、GRE ポイントツーポイント トンネルだけです。 **（注）** GRE トンネルのキープアライブ機能は、VRF トンネルでは設定しないでください。この組み合わせの機能はサポートされていません。
ステップ 6	**tunnel source** {*ip-address* | *interface-type interface-number*} **例：** `Router(config-if)# tunnel source GigabitEthernet 0/0/0`	トンネルの送信元を設定します。 • 送信元 IP アドレスを指定するには、*ip-address* 引数を使用します。 • 使用するインターフェイスを指定するには、*interface-type* 引数および *interface-number* 引数を使用します。 **（注）** トンネルの送信元と宛先の IP アドレスは、2 つの個別のデバイス上で定義する必要があります。
ステップ 7	**tunnel destination** {*hostname* | *ip-address*} **例：** `Router(config-if)# tunnel destination 172.17.2.1`	トンネルの宛先を設定します。 • ホストの宛先の名前を指定するには、*hostname* 引数を指定します。 • ホストの宛先の IP アドレスを指定するには、*ip-address* 引数を指定します。 **（注）** トンネルの送信元と宛先の IP アドレスは、2 つの個別のデバイス上で定義する必要があります。

	コマンドまたはアクション	目的
ステップ 8	**tunnel key** *key-number* **例：** `Router(config-if)# tunnel key 1000`	（オプション）トンネル インターフェイスの ID キーをイネーブルにします。 • 各パケットで運ばれるトンネル キーを識別するには、*key-number* 引数を使用します。 • トンネルの ID キーは、強度の劣るセキュリティ形式として使用して、外部ソースからのパケットの不適切な設定や挿入を防止できます。 **（注）** このコマンドがサポートされるのは、GRE トンネル インターフェイスだけです。セキュリティ目的でこのキーに依存することは推奨しません。
ステップ 9	**tunnel mode** {**gre ip** | **gre multipoint**} **例：** `Router(config-if)# tunnel mode gre ip`	トンネルで使用されるカプセル化プロトコルを指定します。 • IP カプセル化での GRE の使用を指定するには、**gre ip** キーワードを使用します。 • Mltipoint GRE（mGRE; マルチポイント GRE）の使用を指定するには、**gre multipoint** キーワードを使用します。
ステップ 10	**ip mtu** *bytes* **例：** `Router(config-if)# ip mtu 1400`	（任意）各インターフェイスで送信される IP パケットの MTU サイズを設定します。 • インターフェイスに設定されている MTU を IP パケットが超過した場合、DF ビットが設定されていなければ、Cisco IOS XE ソフトウェアは IP パケットをフラグメント化します。 • 物理メディアのすべてのデバイスが動作するには、同じプロトコル MTU が設定されている必要があります。 • IPv6 パケットに対しては、**ipv6 mtu** コマンドを使用します。 **（注）** ステップ 12 で **tunnel path-mtu-discovery** コマンドがイネーブルになっている場合は、このコマンドを設定しないでください。
ステップ 11	**ip tcp mss** *mss-value* **例：** `Router(config-if)# ip tcp mss 250`	（任意）ルータ上で開始または終了する TCP 接続に対して、Maximum Segment Size（MSS; 最大セグメント サイズ）を指定します。 • TCP 接続に対する最大セグメント サイズをバイト単位で指定するには、*mss-value* 引数を使用します。
ステップ 12	**tunnel path-mtu-discovery** [**age-timer** {*aging-mins* | **infinite**}] **例：** `Router(config-if)# tunnel path-mtu-discovery`	（任意）GRE または IP-in-IP トンネル インターフェイスで Path MTU Discovery（PMTUD）をイネーブルにします。 • トンネル インターフェイスで PMTUD をイネーブルにした場合、PMTUD は GRE IP トンネル パケット用に動作し、トンネル エンドポイント間のパス内のフラグメンテーションを最低限に抑えます。
ステップ 13	**end** **例：** `Router(config-if)# end`	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、特権 EXEC モードに戻ります。

次の作業

「トンネルの設定と動作の確認」（P.27）に進みます。

GRE/IPv6 トンネルの設定

この作業では、IPv6 ネットワーク上での GRE トンネルの設定方法について説明します。GRE トンネルを設定して、IPv6 ネットワーク層を介して実行し、IPv6 トンネルに IPv6 パケットを転送して、IPv6 トンネルに IPv4 パケットを転送できます。

前提条件

GRE/IPv6 トンネルが設定されている場合、IPv6 アドレスはトンネルの送信元および宛先に割り当てられます。トンネル インターフェイスには、IPv4 アドレスまたは IPv6 アドレスのいずれかを割り当てることができます（このことは、以降の作業では示されていません）。手動設定されたトンネルの両端のホストまたはルータは、IPv4 プロトコル スタックおよび IPv6 プロトコル スタックをサポートする必要があります。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **tunnel source** {*ipv6-address* | *interface-type interface-number*}
5. **tunnel destination** *ipv6-address*
6. **tunnel mode gre ipv6**
7. **ipv6 mtu** *bytes*
8. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `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 3	**interface tunnel** *tunnel-number* **例：** `Router(config)# interface tunnel 0`	トンネルのインターフェイスおよび番号を指定し、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 4	**tunnel source** {*ipv6-address* | *interface-type interface-number*} **例：** `Router(config-if)# tunnel source GigabitEthernet 0/0/0`	送信元 IPv6 アドレスまたは送信元のインターフェイス タイプおよびトンネル インターフェイスの番号を指定します。 • インターフェイスのタイプおよび番号が指定されている場合、そのインターフェイスは IPv6 アドレスを使用して設定する必要があります。 **（注）** このコンテキストで使用される構文だけが表示されます。詳細については、『*Cisco IOS IPv6 Command Reference*』を参照してください。
ステップ 5	**tunnel destination** *ipv6-address* **例：** `Router(config-if)# tunnel destination 2001:0DB8:0C18:2::300`	トンネル インターフェイスの宛先の IPv6 アドレスを指定します。 **（注）** このコンテキストで使用される構文だけが表示されます。詳細については、『*Cisco IOS IPv6 Command Reference*』を参照してください。
ステップ 6	**tunnel mode gre ipv6** **例：** `Router(config-if)# tunnel mode gre ipv6`	GRE IPv6 トンネルを指定します。 **（注）** **tunnel mode gre ipv6** コマンドは、トンネルのカプセル化プロトコルとして GRE を指定します。
ステップ 7	**ipv6 mtu** *bytes* **例：** `Router(config-if)# ipv6 mtu 1400`	（任意）各インターフェイスで送信される IPv6 パケットの MTU サイズを設定します。
ステップ 8	**end** **例：** `Router(config-if)# end`	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、特権 EXEC モードに戻ります。

次の作業

「トンネルの設定と動作の確認」（P.27）に進みます。

GRE トンネルの IP 送信元および宛先の VRF メンバーシップの設定

この作業では、トンネルの送信元および宛先を任意の VRF テーブルに属するように設定する方法について説明します。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface tunnel** *slot*
4. **ip vrf forwarding** *vrf-name*
5. **ip address** *ip-address subnet-mask*
6. **tunnel source** {*ip-address* | *type number*}
7. **tunnel destination** *ip-address* {*hostname* | *ip-address*}
8. **tunnel vrf** *vrf-name*

手順の詳細

ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `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 3	**interface tunnel** *slot* **例：** `Router(config)# interface tunnel 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。
ステップ 4	**ip vrf forwarding** *vrf-name* **例：** `Router(config-if)# ip vrf forwarding vrf1`	トンネル インターフェイスに関連付けられる VRF を定義します。
ステップ 5	**ip address** *ip-address subnet-mask* **例：** `Router(config-if)# ip address 10.7.7.7 255.255.255.255`	IP アドレスおよびサブネット マスクを指定します。
ステップ 6	**tunnel source** {*ip-address* | *type number*} **例：** `Router(config-if)# tunnel source loopback 0`	トンネル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 7	**tunnel destination** {*hostname* | *ip-address*} **例：** `Router(config-if)# tunnel destination 10.5.5.5`	トンネルの宛先を定義します。
ステップ 8	**tunnel vrf** *vrf-name* **例：** `Router(config-if)# tunnel vrf finance1`	トンネル パケットの送信元である物理インターフェイスに関連付けられる VRF を定義します。

次の作業

「トンネルの設定と動作の確認」（P.27）に進みます。

IPv6 トンネルの手動設定

この作業では、IPv6 オーバーレイ トンネルの手動設定方法について説明します。

前提条件

IPv6 トンネルを手動で設定した場合、IPv6 アドレスはトンネル インターフェイスで設定されます。また、手動設定された IPv4 アドレスはトンネルの送信元および宛先に割り当てられます。手動設定されたトンネルの両端のホストまたはルータは、IPv4 プロトコル スタックおよび IPv6 プロトコル スタックをサポートする必要があります。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **ipv6 address** *ipv6-prefix/prefix-length* [**eui-64**]
5. **tunnel source** {*ip-address* | *interface-type interface-number*}
6. **tunnel destination** *ip-address*
7. **tunnel mode ipv6ip**
8. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `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 3	**interface tunnel** *tunnel-number* **例：** `Router(config)# interface tunnel 0`	トンネルのインターフェイスおよび番号を指定し、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 4	**ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**] **例：** `Router(config-if)# ipv6 address 2001:0DB8:1234:5678::3/126`	インターフェイスに割り当てられる IPv6 ネットワークを指定し、インターフェイスでの IPv6 処理をイネーブルにします。 **（注）** IPv6 アドレスの設定の詳細は、「Configuring Basic Connectivity for IPv6」モジュールを参照してください。
ステップ 5	**tunnel source** {*ip-address* | *interface-type interface-number*} **例：** `Router(config-if)# tunnel source GigabitEthernet 0/0/0`	送信元 IPv4 アドレスまたは送信元インターフェイス タイプおよびトンネル インターフェイスの番号を指定します。 • インターフェイスが指定されている場合、そのインターフェイスは IPv4 アドレスを使用して設定する必要があります。
ステップ 6	**tunnel destination** *ip-address* **例：** `Router(config-if)# tunnel destination 192.168.30.1`	トンネル インターフェイスの宛先の IPv4 アドレスを指定します。
ステップ 7	**tunnel mode ipv6ip** **例：** `Router(config-if)# tunnel mode ipv6ip`	手動設定した IPv6 トンネルを指定します。 **（注）** **tunnel mode ipv6ip** コマンドは、手動 IPv6 トンネル向けに IPv6 をパッセンジャ プロトコルとして指定し、IPv4 をキャリア（カプセル化）プロトコルおよびトランスポート プロトコルの両方として指定します。
ステップ 8	**end** **例：** `Router(config-if)# end`	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、特権 EXEC モードに戻ります。

次の作業

「トンネルの設定と動作の確認」（P.27）に進みます。

6to4 トンネルの設定

この作業では、6to4 オーバーレイ トンネルの設定方法について説明します。

前提条件

6to4 トンネルでは、トンネルの宛先は、境界ルータの IPv4 アドレスによって決定されます。このアドレスは、プレフィクス 2002::/16 と連結されて 2002:*border-router-IPv4-address*::/48 という形式になります。6to4 トンネルの各終端の境界ルータは、IPv4 プロトコル スタックおよび IPv6 プロトコル スタックの両方をサポートする必要があります。

制約事項

ルータでサポートされる設定は、IPv4 互換トンネルおよび 6to4 IPv6 トンネルそれぞれ 1 つだけです。同じルータで両方のトンネル タイプの設定を選択する場合は、これらが同じ送信元を共有しないようにすることを強く推奨します。

6to4 トンネルと IPv4 互換トンネルが同じインターフェイスを共有できない理由は、これらがともに NBMA「ポイントツーマルチポイント」アクセス リンクであり、多重化されたパケット ストリームを着信インターフェイスの単一パケット ストリームに再度配列するには、トンネルの送信元しか使用できないためです。したがって、IPv4 プロトコル タイプが 41 のパケットがインターフェイスに到着すると、このパケットは IPv4 アドレスに基づいて、IPv6 トンネル インターフェイスにマッピングされます。ただし、6to4 トンネルと IPv4 互換トンネルが同じ送信元インターフェイスを共有している場合、ルータは、着信パケットを割り当てるべき IPv6 トンネル インターフェイスを区別できません。

手動トンネルは「ポイントツーポイント」リンクであり、トンネルの IPv4 送信元と IPv4 宛先がともに定義されているので、IPv6 の手動設定トンネルは同じ送信元インターフェイスを共有できます。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**]
5. **tunnel source** {*ip-address* | *interface-type interface-number*}
6. **tunnel mode ipv6ip 6to4**
7. **exit**
8. **ipv6 route** *ipv6-prefix*/*prefix-length* **tunnel** *tunnel-number*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `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 3	**interface tunnel** *tunnel-number* **例：** `Router(config)# interface tunnel 0`	トンネルのインターフェイスおよび番号を指定し、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 4	**ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**] **例：** `Router(config-if)# ipv6 address 2002:c0a8:6301:1::1/64`	インターフェイスに割り当てられる IPv6 アドレスを指定し、インターフェイスでの IPv6 処理をイネーブルにします。 • 最初のプレフィクス 2002::/16 に続く 32 ビットは、トンネル送信元に割り当てられている IPv4 アドレスに相当します。 **（注）** IPv6 アドレスの設定の詳細は、「Configuring Basic Connectivity for IPv6」モジュールを参照してください。
ステップ 5	**tunnel source** {*ip-address* | *interface-type interface-number*} **例：** `Router(config-if)# tunnel source GigabitEthernet 0/0/0`	送信元 IPv4 アドレスまたは送信元インターフェイス タイプおよびトンネル インターフェイスの番号を指定します。 **（注）** **tunnel source** コマンドで指定されているインターフェイスのタイプおよび番号は、IPv4 アドレスを使用して設定することが必要です。
ステップ 6	**tunnel mode ipv6ip 6to4** **例：** `Router(config-if)# tunnel mode ipv6ip 6to4`	6to4 アドレスを使用して、IPv6 オーバーレイ トンネルを指定します。

	コマンドまたはアクション	目的
ステップ 7	**exit** **例：** `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、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 8	**ipv6 route** *ipv6-prefix*/*prefix-length* **tunnel** *tunnel-number* **例：** `Router(config)# ipv6 route 2002::/16 tunnel 0`	指定されているトンネル インターフェイスへの IPv6 6to4 プレフィクス 2002::/16 のスタティック ルートを設定します。 **（注）** 6to4 オーバーレイ トンネルを設定する際、6to4 トンネル インターフェイスへの IPv6 6to4 プレフィクス 2002::/16 のスタティック ルートを設定する必要があります。 • **ipv6 route** コマンドで指定されるトンネル番号は、**interface tunnel** コマンドで指定されているトンネル番号と同じであることが必要です。

次の作業

「トンネルの設定と動作の確認」（P.27）に進みます。

ISATAP トンネルの設定

この作業では、ISATAP オーバーレイ トンネルの設定方法について説明します。

前提条件

ISATAP トンネルの設定で使用される **tunnel source** コマンドは、IPv4 アドレスを使用して設定されたインターフェイスをポイントすることが必要です。ISATAP IPv6 アドレスおよびアドバタイズされたプレフィクス（1 つまたは複数）は、ネイティブ IPv6 インターフェイス向けに設定されます。IPv6 トンネル インターフェイスは、修正された EUI-64 アドレスを使用して設定する必要があります。これは、インターフェイス識別子の最後の 32 ビットが IPv4 トンネル送信元アドレスを使用して作成されているためです。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**]
5. **no ipv6 nd suppress-ra**
6. **tunnel source** {*ip-address* | *interface-type interface-number*}
7. **tunnel mode ipv6ip isatap**
8. **end**

手順の詳細

ステップ 1	**enable** **例：** `Router> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** **例：** `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 3	**interface tunnel** *tunnel-number* **例：** `Router(config)# interface tunnel 1`	トンネルのインターフェイスおよび番号を指定し、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。

コマンドまたはアクション	目的
ステップ 4 **ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**] **例：** `Router(config-if)# ipv6 address 2001:0DB8:6301::/64 eui-64`	インターフェイスに割り当てられる IPv6 アドレスを指定し、インターフェイスでの IPv6 処理をイネーブルにします。 **（注）** IPv6 アドレスの設定の詳細は、「Configuring Basic Connectivity for IPv6」モジュールを参照してください。
ステップ 5 **no ipv6 nd suppress-ra** **例：** `Router(config-if)# no ipv6 nd suppress-ra`	IPv6 ルータ アドバタイズメントの送信をイネーブルにして、クライアントによる自動設定を可能にします。 • デフォルトでは、トンネル インターフェイスでの IPv6 ルータ アドバタイズメントはディセーブルになります。
ステップ 6 **tunnel source** {*ip-address* | *interface-type interface-number*} **例：** `Router(config-if)# tunnel source GigabitEthernet 0/0/1`	送信元 IPv4 アドレスまたは送信元インターフェイス タイプおよびトンネル インターフェイスの番号を指定します。 **（注）** **tunnel source** コマンドで指定されているインターフェイスのタイプおよび番号は、IPv4 アドレスを使用して設定することが必要です。
ステップ 7 **tunnel mode ipv6ip isatap** **例：** `Router(config-if)# tunnel mode ipv6ip isatap`	ISATAP アドレスを使用して、IPv6 オーバーレイ トンネルを指定します。
ステップ 8 **end** **例：** `Router(config-if)# end`	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、特権 EXEC モードに戻ります。

次の作業

「トンネルの設定と動作の確認」（P.27）に進みます。

トンネルの設定と動作の確認

この任意の作業では、トンネルの設定と動作の確認方法について説明します。この手順に含まれる **show** コマンドおよび **ping** コマンドは、任意の順序で実行でき、繰り返し実行する必要がある場合があります。次のコマンドは、GRE トンネル、IPv6 手動設定トンネル、および IPv4 GRE トンネルを介する IPv6 に使用できます。

手順の概要

1. **enable**
2. **show interfaces tunnel** *number* [**accounting**]
3. **ping** [*protocol*] *destination*
4. **show ip route** [*address* [*mask*]]
5. **ping** [*protocol*] *destination*

手順の詳細

ステップ 1　enable

特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。

```
Router> enable
```

ステップ 2　show interfaces tunnel *number* [accounting]

IPv6 手動設定トンネルおよび IPv4 GRE トンネルを介する IPv6 の両方に適した汎用例を想定して、2 つのルータがトンネルのエンドポイントとして設定されているとします。ルータ A では、IPv4 アドレスが 10.0.0.1、IPv6 プレフィクスが 2001:0DB8:1111:2222::1/64 のトンネル インターフェイス 0 に対する送信元として、GigabitEthernet インターフェイス 0/0/0 が設定されています。ルータ B では、IPv4 アドレスが 10.0.0.2、IPv6 プレフィクスが 2001:0DB8:1111:2222::2/64 のトンネル インターフェイス 1 に対する送信元として、ギガビット イーサネット インターフェイス 0/0/0 が設定されています。

トンネルの送信元および宛先アドレスが設定されていることを確認するには、ルータ A で **show interfaces tunnel** コマンドを使用します。

```
RouterA# show interfaces tunnel 0

Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.0.0.1 (GigabitEthernet0/0/0), destination 10.0.0.2, fastswitch TTL 255
  Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
  Tunnel TTL 255
  Checksumming of packets disabled,  fast tunneling enabled
  Last input 00:00:14, output 00:00:04, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     4 packets input, 352 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
```

```
     8 packets output, 704 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

ステップ 3 **ping** [*protocol*] *destination*

ローカル エンドポイントが設定され、動作していることを確認するには、ルータ A で **ping** コマンドを使用します。

```
RouterA# ping 2001:0DB8:1111:2222::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:0DB8:1111:2222::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/20 ms
```

ステップ 4 **show ip route** [*address* [*mask*]]

リモート エンドポイント アドレスに対するルートが存在するかどうかを確認するには、**show ip route** コマンドを使用します。

```
RouterA# show ip route 10.0.0.2

Routing entry for 10.0.0.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Routing Descriptor Blocks:
  * directly connected, via GigabitEthernet0/0/0
      Route metric is 0, traffic share count is 1
```

ステップ 5 **ping** [*protocol*] *destination*

リモート エンドポイント アドレスが到達可能かどうかを確認するには、ルータ A で **ping** コマンドを使用します。

（注） フィルタリングが原因で、**ping** コマンドを使用してもリモート エンドポイント アドレスが到達可能でない場合がありますが、トンネル トラフィックは宛先に到達可能です。

```
RouterA# ping 10.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/21/28 ms
```

リモート IPv6 トンネル エンドポイントが到達可能かどうかを確認するには、ルータ A でもう一度 **ping** コマンドを使用します。この例にも、フィルタリングに関する同じ注釈が適用されます。

```
RouterA# ping 1::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/20 ms
```

これらの手順は、トンネルのもう一方のエンドポイントでも繰り返します。

トンネル実装の設定例

ここでは、次の例について説明します。

GRE/IPv4 トンネルの設定：例

次に、GRE トンネリングの単純な設定例を示します。ギガビット イーサネット インターフェイス 0/0/1 はルータ A のトンネル送信元であり、ルータ B のトンネル宛先です。ファスト イーサネット インターフェイス 0/0/1 はルータ B のトンネル送信元であり、ルータ A のトンネル宛先です。

ルータ A

```
interface Tunnel 0
 ip address 10.1.1.2 255.255.255.0
 tunnel source GigabitEthernet 0/0/1
 tunnel destination 192.168.3.2
 tunnel mode gre ip
!
interface GigabitEthernet 0/0/1
 ip address 192.168.4.2 255.255.255.0
```

ルータ B

```
interface Tunnel 0
 ip address 10.1.1.1 255.255.255.0
 tunnel source FastEthernet 0/0/1
 tunnel destination 192.168.4.2
 tunnel mode gre ip
!
interface FastEthernet 0/0/1
 ip address 192.168.3.2 255.255.255.0
```

次に、ルータ A とルータ B との間で IS-IS および IPv6 トラフィックをともに送出する GRE トンネルを設定する例を示します。

ルータ A

```
ipv6 unicast-routing
clns routing
!
interface Tunnel 0
 no ip address
 ipv6 address 2001:0DB8:1111:2222::1/64
 ipv6 router isis
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 10.0.0.2
```

```
 tunnel mode gre ip
!
interface GigabitEthernet 0/0/0
 ip address 10.0.0.1 255.255.255.0
!
router isis
 network 49.0000.0000.000a.00
```

ルータ B

```
ipv6 unicast-routing
clns routing
!
interface Tunnel 0
 no ip address
 ipv6 address 2001:0DB8:1111:2222::2/64
 ipv6 router isis
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 10.0.0.1
 tunnel mode gre ip
!
interface GigabitEthernet 0/0/0
 ip address 10.0.0.2 255.255.255.0
!
router isis
 network 49.0000.0000.000b.00
 address-family ipv6
 redistribute static
 exit-address-family
```

GRE/IPv6 トンネルの設定：例

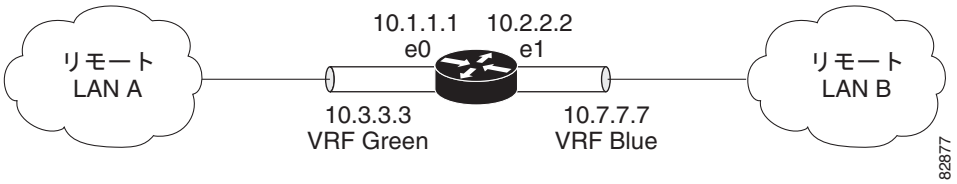
次に、IPv6 トランスポートで GRE トンネルを設定する方法の例を示します。ギガビット イーサネット インターフェイス 0/0/0 には IPv6 アドレスが設定されており、これがトンネル インターフェイスが使用する送信元アドレスとなります。トンネルの宛先 IPv6 アドレスは、直接指定されます。この例では、トンネルは IPv4 トラフィックおよび IS-IS トラックの両方を伝送します。

```
interface Tunnel 0
 ip address 10.1.1.1 255.255.255.0
 ip router isis
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 2001:DB8:1111:2222::1
 tunnel mode gre ipv6
!
interface FastEthernet 0/0
 no ip address
 ipv6 address 2001:DB8:1111:1111::1/64
!
router isis
 net 49.0001.0000.0000.000a.00
```

GRE トンネルの IP 送信元および宛先の VRF メンバーシップの設定：例

この例では、VRF green を使用してファスト イーサネット インターフェイス 0 で受信されたパケットが、VRF blue を使用してファスト イーサネット インターフェイス 1 を使用してトンネルから外部へ転送されます。図 4 に、単純なトンネルのシナリオを示します。

図 4　GRE トンネルの図



次に、図 4 に示したトンネルの設定例を示します。

```
ip vrf blue
 rd 1:1

ip vrf green
 rd 1:2

interface loopback 0
 ip vrf forwarding vrf blue
 ip address 10.7.7.7 255.255.255.255

interface tunnel 0
 ip vrf forwarding vrf green
 ip address 10.3.3.3 255.255.255.0
 tunnel source loopback 0
 tunnel destination 10.5.5.5
 tunnel vrf blue

interface GigabitEthernet 0/0/0
 ip vrf forwarding vrf green
 ip address 10.1.1.1 255.255.255.0

interface GigabitEthernet 0/0/1
 ip vrf forwarding vrf blue
 ip address 10.2.2.2 255.255.255.0

 ip route vrf blue 10.5.5.5 255.255.255.0 GigabitEthernet 0/0/1
```

EoMPLS over GRE の設定：例

次に、EoMPLS over GRE の設定シーケンスの例を示します。

ルータ A の設定

```
vrf definition VPN1
 rd 100:1
 address-family ipv4
 route-target both 100:1
 exit-address-family
 !
mpls label protocol ldp
mpls ldp neighbor 209.165.200.224 targeted
mpls ldp router-id Loopback0 force
!
interface Tunnel 0
 ip address 209.165.200.225 255.255.255.224
 mpls label protocol ldp
 mpls ip
 keepalive 10 3
 tunnel source TenGigabitEthernet 2/1/0
```

```
 tunnel destination 209.165.200.226
!
interface Loopback 0
 ip address 209.165.200.230 255.255.255.224
 !
interface TenGigabitEthernet 2/1/0
 mtu 9216
 ip address 209.165.200.235 255.255.255.224
!
interface TenGigabitEthernet 9/1
 no ip address
!
interface TenGigabitEthernet 9/1.11
 vrf forwarding VPN1
 encapsulation dot1Q 300
 ip address 209.165.200.237 255.255.255.224
!
interface TenGigabitEthernet 9/2
 mtu 9216
 no ip address
xconnect 209.165.200.239 200 encapsulation mpls
!
router bgp 65000
 bgp log-neighbor-changes
 neighbor 209.165.200.240 remote-as 65000
 neighbor 209.165.200.240 update-source Loopback0
neighbor 209.165.200.245 remote-as 100
 !
address-family vpnv4
  neighbor 209.165.200.240 activate
  neighbor 209.165.200.240 send-community extended
 !
 address-family ipv4 vrf VPN1
  no synchronization
  neighbor 209.165.200.247 remote-as 100
  neighbor 209.165.200.248 activate
  neighbor 209.165.200.249 send-community extended
!
ip route 209.165.200.251 255.255.255.224 Tunnel 0
ip route 209.165.200.254 255.255.255.224 209.165.200.256
```

ルータ B の設定

```
vrf definition VPN1
 rd 100:1
 address-family ipv4
 route-target both 100:1
exit-address-family
 !
mpls ldp neighbor 209.165.200.229 targeted
mpls label protocol ldp
mpls ldp router-id Loopback0 force
!
interface Tunnel 0
 ip address 209.165.200.230 255.255.255.224
 mpls label protocol ldp
 mpls ip
 keepalive 10 3
 tunnel source TenGigabitEthernet 3/3/0
 tunnel destination 209.165.200.232
!
interface Loopback 0
 ip address 209.165.200.234 255.255.255.224
!
```

```
interface TenGigabitEthernet 2/1
 mtu 9216
 no ip address
 xconnect 209.165.200.237 200 encapsulation mpls
!
interface TenGigabitEthernet 2/3
 mtu 9216
 no ip address
!
interface TenGigabitEthernet 2/3.11
 vrf forwarding VPN1
 encapsulation dot1Q 300
 ip address 209.165.200.239 255.255.255.224
!
interface TenGigabitEthernet 3/3/0
 mtu 9216
 ip address 209.165.200.240 255.255.255.224
!
router bgp 65000
 bgp log-neighbor-changes
 neighbor 209.165.200.241 remote-as 65000
 neighbor 209.165.200.241 update-source Loopback0
 neighbor 209.165.200.244 remote-as 200
 !
 address-family vpnv4
  neighbor 209.165.200.241 activate
  neighbor 209.165.200.241 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  no synchronization
  neighbor 209.165.200.246 remote-as 200
  neighbor 209.165.200.246 activate
  neighbor 209.165.200.246 send-community extended
 exit-address-family
¡
ip route 209.165.200.226 255.255.255.224 Tunnel 0
ip route 209.165.200.229 255.255.255.224 209.165.200.235
```

IPv6 トンネルの手動設定：例

次に、ルータ A とルータ B の間に IPv6 トンネルを手動で設定する例を示します。例では、ルータ A とルータ B の両方のトンネル インターフェイス 0 は、グローバル IPv6 アドレスを使用して手動で設定されています。トンネルの送信元および宛先のアドレスも手動設定されています。

ルータ A

```
interface GigabitEthernet 0/0/0
 ip address 192.168.99.1 255.255.255.0

interface tunnel 0
 ipv6 address 2001:0db8:c18:1::3/126
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 192.168.30.1
 tunnel mode ipv6ip
```

ルータ B

```
interface GigabitEthernet 0/0/0
 ip address 192.168.30.1 255.255.255.0
```

```
interface tunnel 0
 ipv6 address 2001:0db8:c18:1::2/126
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 192.168.99.1
 tunnel mode ipv6ip
```

6to4 トンネルの設定：例

次に、独立した IPv6 ネットワーク内の境界ルータで 6to4 トンネルを設定する例を示します。IPv4 アドレスは 192.168.99.1 で、IPv6 プレフィクス 2002:c0a8:6301::/48 に変換されます。IPv6 プレフィクスは、トンネル インターフェイス用に 2002:c0a8:6301::/64、1 番目の IPv6 ネットワーク用に 2002:c0a8:6301:1::/64、2 番目の IPv6 ネットワーク用に 2002:c0a8:6301:2::/64 というようにサブネット化されます。スタティック ルートにより、IPv6 プレフィクス 2002::/16 はトンネル インターフェイス 0 に送信され自動的にトンネリングが行われます。

```
interface GigabitEthernet 0/0/0
 description IPv4 uplink
 ip address 192.168.99.1 255.255.255.0
!
interface GigabitEthernet 0/0/1
 description IPv6 local network 1
 ipv6 address 2002:c0a8:6301:1::1/64
!
interface GigabitEthernet 0/0/2
 description IPv6 local network 2
 ipv6 address 2002:c0a8:6301:2::1/64
!
interface Tunnel 0
 description IPv6 uplink
 no ip address
 ipv6 address 2002:c0a8:6301::1/64
 tunnel source GigabitEthernet 0/0/0
 tunnel mode ipv6ip 6to4
!
ipv6 route 2002::/16 Tunnel0
```

ISATAP トンネルの設定：例

次に、ギガビット イーサネット インターフェイス 0/0/0 で定義されるトンネル送信元および ISATAP トンネルの設定に使用される **tunnel mode** コマンドの例を示します。ルータ アドバタイズメントをイネーブルにして、クライアントによる自動設定を可能にします。

```
interface Tunnel 1
 tunnel source GigabitEthernet 0/0/0
 tunnel mode ipv6ip isatap
 ipv6 address 2001:0DB8::/64 eui-64
 no ipv6 nd suppress-ra
```

トンネル インターフェイスにおける QoS オプションの設定：例

次の設定例は、トンネル インターフェイスの GTS に直接適用されます。この例では、設定によりトンネル インターフェイスが総出力レート 500 kb/s にシェーピングされます。

```
interface Tunnel 0
 ip address 10.1.2.1 255.255.255.0
```

```
 traffic-shape rate 500000 125000 125000 1000
 tunnel source 10.1.1.1
 tunnel destination 10.2.2.2
```

次の設定例では、Modular QoS CLI（MQC）コマンドを備えたトンネル インターフェイスに同じシェーピング ポリシーを適用する方法を示します。

```
policy-map tunnel
 class class-default
 shape average 500000 125000 125000
!
interface Tunnel 0
 ip address 10.1.2.1 255.255.255.0
 service-policy output tunnel
 tunnel source 10.1.35.1
 tunnel destination 10.1.35.2
```

ポリシングの例

インターフェイスが混雑しており、パケットのキューイングを開始した場合、送信待ちのパケットにキューイング方式を適用できます。この例に挙げているトンネル インターフェイスである Cisco IOS XE 論理インターフェイスは、混雑した状況を本質的にサポートしておらず、キューイング方式を適用するサービス ポリシーの直接適用もサポートしていません。その代わりに、階層型ポリシーを適用する必要があります。**priority** コマンドを使用した低遅延キューイングや、**bandwidth** コマンドを使用したキューイング メカニズムを設定する「子」ポリシー、つまり下位ポリシーを作成します。

```
policy-map child
 class voice
 priority 512
```

クラスベースのシェーピングに適用する「親」ポリシー、つまり上位ポリシーを作成します。子クラスのアドミッション制御は親クラスのシェーピング レートに従って実行されるので、親ポリシー下で子ポリシーをコマンドとして適用します。

```
policy-map tunnel
 class class-default
 shape average 2000000
 service-policy child
```

親ポリシーをトンネル インターフェイスに適用します。

```
interface tunnel 0
 service-policy tunnel
```

次の例では、トンネル インターフェイスは、シェーピングを行わないキューイングを適用するサービス ポリシーを使用して設定されます。この設定がサポートされないことを通知するログ メッセージが表示されます。

```
Router(config)# interface tunnel1
Router(config-if)# service-policy output child

Class Based Weighted Fair Queueing not supported on this interface
```

その他の参考資料

ここでは、トンネルの実装に関する関連資料について説明します。

関連資料

関連項目	参照先
トンネル コマンド：コマンド構文の詳細、コマンド モード、デフォルト、コマンド履歴、使用上の注意事項、および例	『*Cisco IOS Interface and Hardware Component Command Reference*』
IPv6 コマンド：コマンド構文の詳細、コマンド モード、デフォルト、コマンド履歴、使用上の注意事項、および例	『*Cisco IOS IPv6 Command Reference*』
すべての Cisco IOS XE コマンド	『*Cisco IOS Master Command List, All Releases*』
Cisco IOS XE Interface and Hardware Component コンフィギュレーション モジュール	『*Cisco IOS XE Interface and Hardware Component Configuration Guide, Release 2*』
Cisco IOS XE IPv6 コンフィギュレーション モジュール	『*Cisco IOS XE IPv6 Configuration Guide, Release 2*』
Cisco IOS XE Quality of Service Solutions コンフィギュレーション モジュール	『*Cisco IOS XE Quality of Service Solutions Configuration Guide, Release 2*』
Cisco IOS XE Multiprotocol Label Switching コンフィギュレーション モジュール	『*Cisco IOS XE Multiprotocol Label Switching Configuration Guide, Release 2*』
VRF 対応 Dynamic Multipoint VPN（DMVPN; ダイナミック マルチポイント VPN）の設定例	『*Cisco IOS XE Security Configuration Guide: Secure Connectivity, Release 2*』の「Dynamic Multipoint VPN (DMVPN)」コンフィギュレーション モジュール

規格

規格	タイトル
新しい規格または変更された規格はサポートされていません。また、既存の規格に対するサポートに変更はありません。	—

MIB

MIB	MIB リンク
新しい MIB または変更された MIB はサポートされていません。また、既存の MIB に対するサポートに変更はありません。	選択したプラットフォーム、Cisco IOS XE ソフトウェア リリース、およびフィーチャ セットの MIB の場所を検索しダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

RFC

RFC	タイトル
RFC 791	「*Internet Protocol*」
RFC 1191	「*Path MTU Discovery*（*PMTUD*）」
RFC 1323	「*TCP Extensions for High Performance*」

RFC	タイトル
RFC 1483	「*Multiprotocol Encapsulation over ATM Adaptation Layer 5*」
RFC 2003	「*IP Encapsulation Within IP*」
RFC 2018	「*TCP Selective Acknowledgment Options*」
RFC 2460	「*Internet Protocol, Version 6 (IPv6)*」
RFC 2473	「*Generic Packet Tunneling in IPv6 Specification*」
RFC 2474	「*Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*」
RFC 2516	「*A Method for Transmitting PPP over Ethernet (PPPoE)*」
RFC 2547	「*BGP/MPLS VPNs*」
RFC 2780	「*IANA Allocation Guidelines for Values in the Internet Protocol and Related Headers*」
RFC 2784	「*Generic Routing Encapsulation (GRE)*」
RFC 2890	「*Key and Sequence Number Extensions to GRE*」
RFC 2893	「*Transition Mechanisms for IPv6 Hosts and Routers*」
RFC 3056	「*Connection of IPv6 Domains via IPv4 Clouds*」
RFC 3147	「*Generic Routing Encapsulation over CLNS Networks*」

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカル サポートを最大限に活用してください。 以下を含むさまざまな作業にこの Web サイトが役立ちます。 • テクニカル サポートを受ける • ソフトウェアをダウンロードする • セキュリティの脆弱性を報告する、またはシスコ製品のセキュリティ問題に対する支援を受ける • ツールおよびリソースへアクセスする – Product Alert の受信登録 – Field Notice の受信登録 – Bug Toolkit を使用した既知の問題の検索 • Networking Professionals（NetPro）コミュニティで、技術関連のディスカッションに参加する • トレーニング リソースへアクセスする • TAC Case Collection ツールを使用して、ハードウェアや設定、パフォーマンスに関する一般的な問題をインタラクティブに特定および解決する この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	http://www.cisco.com/en/US/support/index.html

トンネルの実装に関する機能情報

表 5 に、このモジュールで説明した機能をリストし、特定の設定情報へのリンクを示します。

プラットフォーム サポートとソフトウェア イメージ サポートに関する情報を入手するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator を使用すると、Cisco IOS XE ソフトウェア イメージがサポートする特定のソフトウェア リリース、機能セット、またはプラットフォームを確認できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスします。Cisco.com のアカウントは必要ありません。

（注） 表 5 には、一連の Cisco IOS XE ソフトウェア リリースのうち、特定の機能が初めて導入された Cisco IOS XE ソフトウェア リリースだけを示します。その機能は、特に断りがない限り、それ以降の一連の Cisco IOS XE ソフトウェア リリースでもサポートされます。

表 5　トンネルの実装に関する機能情報

機能名	リリース	機能の設定情報
GRE トンネルの IP 送信元および宛先の VRF メンバーシップ	Cisco IOS XE Release 2.2	この機能では、トンネルの送信元および宛先が任意の VPN VRF テーブルに属するように設定できます。 この機能に関する詳細については、次の各項を参照してください。 • 「GRE トンネルの IP 送信元および宛先の VRF メンバーシップ」（P.5） • 「GRE トンネルの IP 送信元および宛先の VRF メンバーシップの設定」（P.20） • 「GRE トンネルの IP 送信元および宛先の VRF メンバーシップの設定：例」（P.30） この機能をサポートするために **tunnel vrf** コマンドが導入されました。
GRE トンネル キープアライブ	Cisco IOS XE Release 2.1	GRE トンネル キープアライブ機能により、IP カプセル化された GRE トンネルを介してキープアライブ パケットが送信されるように設定できるようになります。キープアライブが送信されるレートと、インターフェイスが非アクティブになるまでデバイスが応答なしでキープアライブ パケットの送信を続行する回数を指定できます。GRE キープアライブ パケットは、トンネルの両側または片側のみのどちらからでも送信できます。 この機能については、次の項に説明があります。 • 「GRE トンネルの設定」（P.14） この機能により、**keepalive** コマンド（トンネル インターフェイス）が導入されました。

表 5　トンネルの実装に関する機能情報（続き）

機能名	リリース	機能の設定情報
IPv6 IP トンネルを介する IP	Cisco IOS XE Release 2.4	この機能に関する詳細については、次の各項を参照してください。 • 「GRE/IPv6 トンネルの設定」（P.18） • 「GRE/IPv6 トンネルの設定：例」（P.30） この機能により、**tunnel destination**、**tunnel mode**、および **tunnel source** の各コマンドが導入されました。
GRE トンネル向け IP Precedence	Cisco IOS XE Release 2.1	この機能は、Cisco ASR 1000 アグリゲーション サービス ルータに導入されました。
Tunnel ToS	Cisco IOS XE Release 2.1	Tunnel ToS 機能を使用して、ルータの IP トンネル インターフェイス向けのトンネル パケットの IP カプセル化ヘッダーに、ToS および Time-to-Live（TTL）バイト値を設定できます。Tunnel ToS 機能は、シスコ エクスプレス フォワーディング、ファスト スイッチング、およびプロセス スイッチング フォワーディングの各モードでサポートされます。 この機能については、次の項に説明があります。 • 「Tunnel ToS」（P.4） この機能によって、**show interfaces tunnel**、**tunnel tos**、および **tunnel ttl** の各コマンドが導入されました。
EoMPLS over GRE	Cisco IOS XE Release 2.5	EoMPLS over GRE 機能により、レイヤ 3 MPLS ネットワークを経由してレイヤ 2 トラフィックをトンネリングできます。またこの機能では、GRE トンネル内で EoMPLS フレームをカプセル化する高性能のハードウェアベースのスイッチド トンネルとして GRE トンネルを作成できます。 この機能については、次の項に説明があります。 • 「EoMPLS over GRE」（P.5） この機能によって導入または変更された新しいコマンドはありません。